LuaCV

Generated by Doxygen latex template and Lua interpreter from LuaCV source and OpenCV docs

Made by Jiří Prymus

Mon Apr 30 16:44:39 2012

Contents

Chapter 1

About LuaCV

LuaCV is OpenCV library wrapper for Lua language. Main goal is to access fast image processing library to light-weight, non-type language. Thanks to CMake project is a much easier to compile it under many OS platforms which are supported by OpenCV. LuaCV is being developed at BRNO UNIVERSITY OF TECHNOLOGY in Czech Republic and it aims to partialy replace Matlab in our Image Processing Courses. In current stage of development of LuaCV are implemented C API function and objects from imgproc/core/highgui/calibration modules.

1.1 LuaCV License

Copyright (C) 2010-2011, Jiří Prymus <jiri DOT prymus AT gmail DOT com>

LuaCV is licensed under the terms of the LGPL v3 license reproduced below. This means that LuaCV is free software and can be used for both academic and commercial purposes.

1.2 GPL Licecnse

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

1. Additional Definitions.

   As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

2. Exception to Section 3 of the GNU GPL.

   You may convey a covered work under sections 3 and 4 of this License without begin bound by section 3 of the GNU GPL.

3. Conveying Modified Versions.

   If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

   - under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
   - under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

4. Object Code Incorporating Material from Library Header Files.

   The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

   - Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
   - Accompany the object code with a copy of the GNU GPL and this license document.

5. Combined Works.

   You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- Accompany the Combined Work with a copy of the GNU GPL and this license document.
- For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- Do one of the following:

  0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

  1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

6. Combined Libraries.

   You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

   - Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
   - Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

7. Revised Versions of the GNU Lesser General Public License.

   The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General

Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Chapter 2

Changelog

There is list of changes between LuaCV versions in this chapter.

2.1 Changes in 0.2.1

- New modular system
- Some samples completion
- Beginings of OpenCV C++ API
- No further backward compatibility for Lua 5.1

2.2 Changes in 0.2.0

- Almost all OpenCV C API implemented.
- Many new samples to test new functions.
- Allocation bugs fixed.
- OpenCV 2.3 compatibility.
- Lua 5.2 compatibility.
- NSIS MS Windows installer.
- Partial documentation.

2.3 Changes in 0.1.4

- New allocation system for all objects (using boxed pointer). Big objects like matrix and images aren't copied now. You can releas every object by obj=nil collectgarbage('collect') Releasing of objects is done by __gc callback with (cvFree/cvRelease* func).
- Minor changes in LuaCV API to maximize optimalization by compiler (template functions, using aligned OpenCV alloc/free).
- Some new types for image processing module (CvSubdiv2D*).

- Implemented functions for CvGraph handling in core module.
- Implemented various containters for imgproc module.
- Whole highgui C API is implemented (or almost all functions from C api).
- Most of functions from imgproc C API module are implemented, but need proper testing.
- New samples in sample directory to test new implemented functions from imgproc module (almost all c api samples).
- OpenCV callback functions for tracbars and mouse events are now fully implemented. See functions CreateTrackbar and SetMouseCallback from highgui module or samples for basic usage.

2.4 Changes in 0.1.3

- Whole wrapper was rewritten to new OpenCV 2.2. It has to be done because of new modular system in OpenCV. And that's why new LuaCV won't be back compatibile with OpenCV < 2.2.

Chapter 3

Function reference manual

This section contains actual references for LuaCV functions sorted by name and modules . There is actualy implemented only C API OpenCV functions in LuaCV, but C API is still major part of OpenCV functions.

3.1 Module core

3.1.1 AXPY

- **AXPY**(CvArr src1, num real_scalar, CvArr src2, CvArr dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
real_scalar	A floating point number
src2	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.2 Abs

- **Abs**(CvArr src1, CvArr dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.3 AbsDiff

- **AbsDiff**(CvArr src1, CvArr src2, CvArr dst)

  Calculates absolute difference between two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.4 AbsDiffS

- **AbsDiffS**(CvArr src1, CvArr dst, CvScalar value)

  Calculates absolute difference between an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array
value	The scalar

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.5 Add

- **Add**(CvArr src1, CvArr src2, CvArr dst, CvArr mask=nil)

  Computes the per-element sum of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.6 AddS

- **AddS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

  Computes the sum of an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Added scalar
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.7 AddWeighted

- **AddWeighted**(CvArr src1, num alpha, CvArr src2, num beta, num gamma, CvArr dst)

  Computes the weighted sum of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
alpha	Weight for the first array elements
src2	The second source array
beta	Weight for the second array elements
gamma	Scalar, added to each sum
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.8 And

- **And**(CvArr src1, CvArr src2, CvArr dst, CvArr mask=nil)

  Calculates per-element bit-wise conjunction of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.9 AndS

- **AndS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

  Calculates per-element bit-wise conjunction of an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Scalar to use in the operation
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.10 AvgSdv

- **AvgSdv**(CvArr arr, CvScalar mean, CvScalar std_dev, CvArr mask=nil)

  Calculates average (mean) of array elements.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	The array
mean	Mean value, a CvScalar
std_dev	A wrappered OpenCV object of type CvScalar
mask	The optional operation mask

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.11 BackProjectPCA

- **BackProjectPCA**(CvArr proj, CvArr mean, CvArr eigenvals, CvArr result)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

proj	A wrappered OpenCV object of type CvArr
mean	A wrappered OpenCV object of type CvArr
eigenvals	A wrappered OpenCV object of type CvArr
result	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.12 CalcCovarMatrix

- **CalcCovarMatrix**(CvArr[] vects, int count, CvArr cov_mat, CvArr avq, int flags)

  Calculates covariance matrix of a set of vectors.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

vects	The input vectors, all of which must have the same type and the same size. The vectors do not have to be 1D, they can be 2D (e.g., images) and so forth
count	The number of input vectors
cov_mat	A wrappered OpenCV object of type CvArr
avq	A wrappered OpenCV object of type CvArr
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.13 CalcPCA

- **CalcPCA**(CvArr data, CvArr mean, CvArr eigenvals, CvArr eigenvects, int flags)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

data	A wrappered OpenCV object of type CvArr
mean	A wrappered OpenCV object of type CvArr
eigenvals	A wrappered OpenCV object of type CvArr
eigenvects	A wrappered OpenCV object of type CvArr
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.14 CartToPolar

- **CartToPolar**(CvArr x, CvArr y, CvArr magnitude, CvArr angle=nil, int angle_in_degrees=0)

  Calculates the magnitude and/or angle of 2d vectors.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

x	The array of x-coordinates
y	The array of y-coordinates
magnitude	The destination array of magnitudes, may be set to NULL if it is not needed
angle	The destination array of angles, may be set to NULL if it is not needed. The angles are measured in radians $(0$ to $2\pi)$ or in degrees (0 to 360 degrees).
angle_in_degrees	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.15 Cbrt

- num **Cbrt**(num value)

  Calculates the cubic root

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

value	The input floating-point value

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.16 Ceil

- int **Ceil**(num value)

  Converts a floating-point number to the nearest integer value that is not smaller than the argument.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

value	The input floating-point value

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.17 ChangeSeqBlock

- **ChangeSeqBlock**(userdata reader, int direction)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

reader	An unspecified C pointer to memory
direction	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.18 CheckArr

- int **CheckArr**(CvArr arr, int flags=0, num min_val=0, num max_val=0)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
flags	An integer type number with default value *0*
min_val	A floating point number with default value *0*
max_val	A floating point number with default value *0*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.19 Circle

- **Circle**(CvArr img, CvPoint center, int radius, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws a circle.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image where the circle is drawn
center	Center of the circle
radius	Radius of the circle
color	Circle color
thickness	Thickness of the circle outline if positive, otherwise this indicates that a filled circle is to be drawn
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the center coordinates and radius value

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.20 ClearGraph

- **ClearGraph**(CvGraph graph)

  Clears a graph.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.21 ClearMemStorage

- **ClearMemStorage**(CvMemStorage storage)

  Clears memory storage.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

storage	Memory storage

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.22 ClearND

- **ClearND**(CvArr arr, int[] idx)

  Clears a specific array element.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx	Array of the element indices

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.23 ClearSeq

- **ClearSeq**(CvSeq seq)

  Clears a sequence.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

seq	Sequence

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.24 ClearSet

- **ClearSet**(CvSet set_header)

  Clears a set.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

set_header	A wrappered OpenCV object of type CvSet

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.25 ClipLine

- int **ClipLine**(CvSize img_size, CvPoint pt1, CvPoint pt2)

  Clips the line against the image rectangle.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img_size	A wrappered OpenCV object of type CvSize
pt1	First ending point of the line segment. **It is modified by the function.**
pt2	Second ending point of the line segment. **It is modified by the function.**

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.26 Cmp

- **Cmp**(CvArr src1, CvArr src2, CvArr dst, int cmp_op)

  Performs per-element comparison of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array. Both source arrays must have a single channel.
dst	The destination array, must have 8u or 8s type
cmp_op	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.27 CmpS

- **CmpS**(CvArr src1, num value, CvArr dst, int cmp_op)

  Performs per-element comparison of an array and a scalar.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	The scalar value to compare each array element with
dst	The destination array, must have 8u or 8s type
cmp_op	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.28 CompleteSymm

- **CompleteSymm**(CvMat matrix, int LtoR=0)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

matrix	A wrappered OpenCV object of type CvMat
LtoR	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.29 Convert

- **Convert**(CvArr src, CvArr dst)

  Converts one array to another.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array
dst	Destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.30 ConvertScale

- **ConvertScale**(CvArr src, CvArr dst, num scale=1, num shift=0)

  Converts one array to another with optional linear transformation.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array
dst	Destination array
scale	Scale factor
shift	Value added to the scaled source array elements

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.31 ConvertScaleAbs

- **ConvertScaleAbs**(CvArr src, CvArr dst, num scale=1, num shift=0)

  Converts input array elements to another 8-bit unsigned integer with optional linear transformation.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array
dst	Destination array (should have 8u depth)
scale	ScaleAbs factor
shift	Value added to the scaled source array elements

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.32 Copy

- **Copy**(CvArr src, CvArr dst, CvArr mask=nil)

  Copies one array to another.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.33 CountNonZero

- int **CountNonZero**(CvArr arr)

  Counts non-zero array elements.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	The array must be a single-channel array or a multi-channel image with COI set

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.34 CreateData

- **CreateData**(CvArr arr)

  Allocates array data

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Array header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.35 CreateGraph

- **CreateGraph**(int graph_flags, int header_size, int vtx_size, int edge_size, CvMemStorage storage)

  Creates an empty graph.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph_flags	An integer type number
header_size	Graph header size; may not be less than `sizeof(CvGraph)`
vtx_size	Graph vertex size; the custom vertex structure must start with **CvGraphVtx** (use `CV_GRAPH_VERTEX_FIELDS()`)
edge_size	Graph edge size; the custom edge structure must start with **CvGraphEdge** (use `CV_GRAPH_EDGE_FIELDS()`)
storage	The graph container

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.36 CreateSeqBlock

- **CreateSeqBlock**(CvSeqWriter writer)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

writer	A wrappered OpenCV object of type CvSeqWriter

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.37 CrossProduct

- **CrossProduct**(CvArr src1, CvArr src2, CvArr dst)

  Calculates the cross product of two 3D vectors.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source vector
src2	The second source vector
dst	The destination vector

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.38 CvtSeqToArray

- userdata **CvtSeqToArray**(CvSeq seq, userdata[] elements, CvSlice slice=CV_WHOLE_SEQ, string convert_to=")

  Copies a sequence to one continuous block of memory.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
elements	Pointer to the destination array that must be large enough. It should be a pointer to data, not a matrix header.
slice	The sequence portion to copy to the array
convert_to	An array of characters with default value *with default value CV_WHOLE_SEQ*

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.39 DCT

- **DCT**(CvArr src, CvArr dst, int flags)

  Performs a forward or inverse Discrete Cosine transform of a 1D or 2D floating-point array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array, real 1D or 2D array
dst	Destination array of the same size and same type as the source
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.40 DFT

- **DFT**(CvArr src, CvArr dst, int flags, int nonzero_rows=0)

  Performs a forward or inverse Discrete Fourier transform of a 1D or 2D floating-point array.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	Source array, real or complex
dst	Destination array of the same size and same type as the source
flags	An integer type number
nonzero_rows	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.41 DecRefData

- **DecRefData**(CvArr arr)

  Decrements an array data reference counter.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Pointer to an array header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.42 Det

- num **Det**(CvArr mat)

  Returns the determinant of a matrix.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

mat	The source matrix

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.43 Div

- **Div**(CvArr src1, CvArr src2, CvArr dst, num scale=1)

  Performs per-element division of two arrays.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	The first source array. If the pointer is NULL, the array is assumed to be all 1's.
src2	The second source array
dst	The destination array
scale	Optional scale factor

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.44 DotProduct

- num **DotProduct**(CvArr src1, CvArr src2)

  Calculates the dot product of two arrays in Euclidian metrics.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.45 DrawContours

- **DrawContours**(CvArr img, CvSeq contour, CvScalar external_color, CvScalar hole_color, int max_level, int thickness=1, int line_type=8, CvPoint offset=Point(0,0))

  Draws contour outlines or interiors in an image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image where the contours are to be drawn. As with any other drawing function, the contours are clipped with the ROI.
contour	Pointer to the first contour
external_color	Color of the external contours
hole_color	Color of internal contours (holes)
max_level	An integer type number
thickness	An integer type number with default value *1*
line_type	An integer type number with default value *8*
offset	A wrappered OpenCV object of type CvPoint with default value *Point(0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.46 EigenVV

- **EigenVV**(CvArr mat, CvArr evects, CvArr evals, num eps=0, int lowindex=-1, int highindex=-1)

  Computes eigenvalues and eigenvectors of a symmetric matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	The input symmetric square matrix, modified during the processing
evects	The output matrix of eigenvectors, stored as subsequent rows
evals	The output vector of eigenvalues, stored in the descending order (order of eigenvalues and eigenvectors is syncronized, of course)
eps	A floating point number with default value *0*
lowindex	An integer type number with default value *with default value 0*
highindex	An integer type number with default value *with default value with default value 0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.47 Ellipse

- **Ellipse**(CvArr img, CvPoint center, CvSize axes, num angle, num start_angle, num end_angle, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws a simple or thick elliptic arc or an fills ellipse sector.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	The image
center	Center of the ellipse
axes	Length of the ellipse axes

angle	Rotation angle
start_angle	Starting angle of the elliptic arc
end_angle	Ending angle of the elliptic arc.
color	Ellipse color
thickness	Thickness of the ellipse arc outline if positive, otherwise this indicates that a filled ellipse sector is to be drawn
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the center coordinates and axes' values

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.48 Ellipse2Poly

- int **Ellipse2Poly**(CvPoint center, CvSize axes, int angle, int argc_end, CvPoint pts, int delta)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

center	A wrappered OpenCV object of type CvPoint
axes	A wrappered OpenCV object of type CvSize
angle	An integer type number
argc_end	An integer type number
pts	A wrappered OpenCV object of type CvPoint
delta	An integer type number

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.49 EllipseBox

- **EllipseBox**(CvArr img, CvBox2D box, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws a simple or thick elliptic arc or fills an ellipse sector.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image
box	The enclosing box of the ellipse drawn
color	A wrappered OpenCV object of type CvScalar
thickness	Thickness of the ellipse boundary
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the box vertex coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.50 Exp

- **Exp**(CvArr src1, CvArr dst)

  Calculates the exponent of every array element.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array, it should have `double` type or the same type as the source

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.51 FastArctan

- num **FastArctan**(num y, num x)

  Calculates the angle of a 2D vector.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

y	y-coordinate of 2D vector
x	x-coordinate of 2D vector

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.52 FillConvexPoly

- **FillConvexPoly**(CvArr img, CvPoint pts, int npts, CvScalar color, int line_type=8, int shift=0)

  Fills a convex polygon.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

img	Image
pts	Array of pointers to a single polygon
npts	Polygon vertex counter
color	Polygon color
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the vertex coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.53 FillPoly

- **FillPoly**(CvArr img, CvPoint[] pts, int[] npts, int contours, CvScalar color, int line_type=8, int shift=0)

  Fills a polygon's interior.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image
pts	Array of pointers to polygons
npts	Array of polygon vertex counters
contours	Number of contours that bind the filled region
color	Polygon color
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the vertex coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.54 Flip

- **Flip**(CvArr src, CvArr dst=nil, int flip_mode=0)

  Flip a 2D array around vertical, horizontal or both axes.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array
dst	A wrappered OpenCV object of type CvArr with default value *nil*
flip_mode	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.55 Floor

- int **Floor**(num value)

  Converts a floating-point number to the nearest integer value that is not larger than the argument.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

value	The input floating-point value

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.56 FlushSeqWriter

- **FlushSeqWriter**(CvSeqWriter writer)

  Updates sequence headers from the writer.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

writer	Writer state

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.57 GEMM

- **GEMM**(CvArr src1, CvArr src2, num alpha, CvArr src3, num beta, CvArr dst, int tABC=0)

  Performs generalized matrix multiplication.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
alpha	A floating point number
src3	The third source array (shift). Can be NULL, if there is no shift.
beta	A floating point number
dst	The destination array
tABC	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.58 GetDims

- int **GetDims**(CvArr arr, int[] sizes=nil)

  Returns list of array dimensions

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Input array
sizes	An integer type number which is an array with default value *nil*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.59 GetDimsSize

- int **GetDimsSize**(CvArr arr,int index)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
index	An integer type number

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.60 GetElemType

- int **GetElemType**(CvArr arr)

  Returns type of array elements.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.61 GetImageCOI

- int **GetImageCOI**(IplImage image)

  Returns the index of the channel of interest.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A pointer to the image header

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.62 GetOptimalDFTSize

- int **GetOptimalDFTSize**(int size0)

  Returns optimal DFT size for a given vector size.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

size0	Vector size

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.63 GetReal1D

- num **GetReal1D**(CvArr arr, int idx0)

  Return a specific element of single-channel 1D array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array. Must have a single channel.
idx0	The first zero-based component of the element index

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.64 GetReal2D

- num **GetReal2D**(CvArr arr, int idx0, int idx1)

  Return a specific element of single-channel 2D array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array. Must have a single channel.
idx0	The first zero-based component of the element index
idx1	The second zero-based component of the element index

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.65 GetReal3D

- num **GetReal3D**(CvArr arr, int idx0, int idx1, int idx2)

  Return a specific element of single-channel array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array. Must have a single channel.
idx0	The first zero-based component of the element index
idx1	The second zero-based component of the element index
idx2	The third zero-based component of the element index

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.66 GetRealND

- num **GetRealND**(CvArr arr, int[] idx)

  Return a specific element of single-channel array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx	Array of the element indices

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.67 GetSeqElem

- userdata **GetSeqElem**(CvSeq seq, int index, string convert_to=")

  Returns a pointer to a sequence element according to its index.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
index	Index of element
convert_to	An array of characters with default value

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.68 GetSeqReaderPos

- int **GetSeqReaderPos**(CvSeqReader reader)

  Returns the current reader position.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

reader	Reader state

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.69 GetTextSize

- int **GetTextSize**(string text, CvFont font, CvSize text_size)

  Retrieves the width and height of a text string.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

text	An array of characters
font	Pointer to the font structure
text_size	A wrappered OpenCV object of type CvSize

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.70 GetTickCount

- num **GetTickCount**()

  Returns the number of ticks.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

none	function doesn't have input arguments.

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.71 GetTickFrequency

- num **GetTickFrequency**()

  Returns the number of ticks per microsecond.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

none	function doesn't have input arguments.

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.72 GraphAddEdge

- int **GraphAddEdge**(CvGraph graph, int start_idx, int end_idx, CvGraphEdge edge=nil, CvGraphEdge inserted_edge=nil)

  Adds an edge to a graph.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
start_idx	Index of the starting vertex of the edge
end_idx	Index of the ending vertex of the edge. For an unoriented graph, the order of the vertex parameters does not matter.
edge	Optional input parameter, initialization data for the edge
inserted_edge	Optional output parameter to contain the address of the inserted edge

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.73 GraphAddEdgeByPtr

- int **GraphAddEdgeByPtr**(CvGraph graph, CvGraphVtx start_vtx, CvGraphVtx end_vtx, CvGraphEdge edge=nil, CvGraphEdge inserted_edge=nil)

  Adds an edge to a graph by using its pointer.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

graph	Graph
start_vtx	Pointer to the starting vertex of the edge
end_vtx	Pointer to the ending vertex of the edge. For an unoriented graph, the order of the vertex parameters does not matter.
edge	Optional input parameter, initialization data for the edge
inserted_edge	Optional output parameter to contain the address of the inserted edge within the edge set

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.74 GraphAddVtx

- int **GraphAddVtx**(CvGraph graph, CvGraphVtx vtx=nil, CvGraphVtx inserted_vtx=nil)

  Adds a vertex to a graph.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

graph	Graph
vtx	Optional input argument used to initialize the added vertex (only user-defined fields beyond `sizeof(CvGraphVtx)` are copied)
inserted_vtx	A wrappered OpenCV object of type CvGraphVtx with default value *nil*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.75 GraphEdgeIdx

- int **GraphEdgeIdx**(CvGraph graph, CvGraphEdge edge)

  Returns the index of a graph edge.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

graph	Graph
edge	Pointer to the graph edge

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.76 GraphGetEdgeCount

- int **GraphGetEdgeCount**(CvGraph graph)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	A wrappered OpenCV object of type CvGraph

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.77 GraphGetVtxCount

- int **GraphGetVtxCount**(CvGraph graph)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	A wrappered OpenCV object of type CvGraph

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.78 GraphRemoveEdge

- **GraphRemoveEdge**(CvGraph graph, int start_idx, int end_idx)

  Removes an edge from a graph.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
start_idx	Index of the starting vertex of the edge
end_idx	Index of the ending vertex of the edge. For an unoriented graph, the order of the vertex parameters does not matter.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.79 GraphRemoveEdgeByPtr

- **GraphRemoveEdgeByPtr**(CvGraph graph, CvGraphVtx start_vtx, CvGraphVtx end_vtx)

  Removes an edge from a graph by using its pointer.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
start_vtx	Pointer to the starting vertex of the edge
end_vtx	Pointer to the ending vertex of the edge. For an unoriented graph, the order of the vertex parameters does not matter.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.80 GraphRemoveVtx

- **GraphRemoveVtx**(CvGraph graph, int index)

  Removes a vertex from a graph.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
index	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.81 GraphRemoveVtxByPtr

- **GraphRemoveVtxByPtr**(CvGraph graph, CvGraphVtx vtx)

  Removes a vertex from a graph by using its pointer.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
vtx	Pointer to the removed vertex

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.82 GraphVtxDegree

- int **GraphVtxDegree**(CvGraph graph, int vtx_idx)

  Counts the number of edges indicent to the vertex.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
vtx_idx	An integer type number

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.83 GraphVtxDegreeByPtr

- int **GraphVtxDegreeByPtr**(CvGraph graph, CvGraphVtx vtx)

  Finds an edge in a graph.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
vtx	Pointer to the graph vertex

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.84 GraphVtxIdx

- int **GraphVtxIdx**(CvGraph graph, CvGraphVtx vtx)

  Returns the index of a graph vertex.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

graph	Graph
vtx	Pointer to the graph vertex

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.85 InRange

- **InRange**(CvArr src1, CvArr lower, CvArr upper, CvArr dst)

  Checks that array elements lie between the elements of two other arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
lower	The inclusive lower boundary array
upper	The exclusive upper boundary array
dst	The destination array, must have 8u or 8s type

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.86 InRangeS

- **InRangeS**(CvArr src1, CvScalar lower, CvScalar upper, CvArr dst)

  Checks that array elements lie between two scalars.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
lower	The inclusive lower boundary
upper	The exclusive upper boundary
dst	The destination array, must have 8u or 8s type

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.87 IncRefData

- **IncRefData**(CvArr arr)

  Increments array data reference counter.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Array header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.88 InitFont

- **InitFont**(CvFont font, int font_face, num hscale, num vscale, num shear=0, int thickness=1, int line_type=8)

  Initializes font structure.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

font	Pointer to the font structure initialized by the function
font_face	An integer type number
hscale	Horizontal scale. If equal to `1.0f`, the characters have the original width depending on the font type. If equal to `0.5f`, the characters are of half the original width.
vscale	Vertical scale. If equal to `1.0f`, the characters have the original height depending on the font type. If equal to `0.5f`, the characters are of half the original height.
shear	A floating point number with default value *0*
thickness	Thickness of the text strokes
line_type	An integer type number with default value *8*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.89 InitLineIterator

- int **InitLineIterator**(CvArr image, CvPoint pt1, CvPoint pt2, CvLineIterator line_iterator, int connectivity=8, int left_to_righ=0)

  Initializes the line iterator.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

image	Image to sample the line from
pt1	First ending point of the line segment
pt2	Second ending point of the line segment

line_iterator	Iterator over the pixels of the line
connectivity	The scanned line connectivity, 4 or 8.
left_to_righ	An integer type number with default value *0*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.90 InitNArrayIterator

- int **InitNArrayIterator**(int count, CvArr arrs[], CvArr mask, CvMatND stubs, CvNArrayIterator array_iterator, int flags=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

count	An integer type number
arrs	A wrappered OpenCV object of type CvArr
mask	A wrappered OpenCV object of type CvArr
stubs	A wrappered OpenCV object of type CvMatND
array_iterator	A wrappered OpenCV object of type CvNArrayIterator
flags	An integer type number with default value *0*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.91 InitTreeNodeIterator

- **InitTreeNodeIterator**(CvTreeNodeIterator tree_iterator, userdata first, int max_level)

  Initializes the tree node iterator.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tree_iterator	Tree iterator initialized by the function
first	The initial node to start traversing from
max_level	The maximal level of the tree (`first` node assumed to be at the first level) to traverse up to. For example, 1 means that only nodes at the same level as `first` should be visited, 2 means that the nodes on the same level as `first` and their direct children should be visited, and so forth.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.92 InsertNodeIntoTree

- **InsertNodeIntoTree**(userdata node, userdata parent, userdata frame)

  Adds a new node to a tree.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

node	The inserted node
parent	The parent node that is already in the tree
frame	The top level node. If `parent` and `frame` are the same, the `v_prev` field of `node` is set to NULL rather than `parent`.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.93 InvSqrt

- num **InvSqrt**(num value)

  Calculates the inverse square root.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

value	The input floating-point value

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.94 Invert

- num **Invert**(CvArr src, CvArr dst, int method=CV_LU)

  Finds the inverse or pseudo-inverse of a matrix.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	The source matrix
dst	The destination matrix
method	An integer type number with default value *CV_LU*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.95 IplDepth

- int **IplDepth**(int type)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

type	An integer type number

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.96 IsInf

- int **IsInf**(num value)

  Determines if the argument is Infinity.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

value	The input floating-point value

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.97 IsNaN

- int **IsNaN**(num value)

  Determines if the argument is Not A Number.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

value	The input floating-point value

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.98 KMeans2

- int,num **KMeans2**(CvArr samples, int cluster_count, CvArr labels, CvTermCriteria termcrit, int attmepts=1, CvRNG rng=0, int flags=0, CvArr _centers=0, num compactness=0)

  Splits set of vectors by a given number of clusters.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

samples	Floating-point matrix of input samples, one row per sample
cluster_count	An integer type number

labels	Output integer vector storing cluster indices for every sample
termcrit	Specifies maximum number of iterations and/or accuracy (distance the centers can move by between subsequent iterations)
attmepts	An integer type number with default value *1*
rng	Optional external random number generator; can be used to fully control the function behaviour
flags	An integer type number with default value *0*
_centers	A wrappered OpenCV object of type CvArr with default value *0*
compactness	A floating point number with default value *0*

Returns

int - An integer type number defined by architecture
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.99 LUT

- **LUT**(CvArr src, CvArr dst, CvArr lut)

  Performs a look-up table transform of an array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array of 8-bit elements
dst	Destination array; will have the same size and the same number of channels as `src`, and the same depth as `lut`
lut	Look-up table of 256 elements. In the case of multi-channel source array, the table should either have a single channel (in this case the same table is used for all channels) or the same number of channels as in the source array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.100 Line

- **Line**(CvArr img, CvPoint pt1, CvPoint pt2, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws a line segment connecting two points.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	The image
pt1	First point of the line segment
pt2	Second point of the line segment
color	Line color
thickness	Line thickness
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the point coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.101 Load

- userdata,string **Load**(string filename, string convert_to="", CvMemStorage storage=nil, string name="")

  Loads an object from a file.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

filename	File name
convert_to	An array of characters with default value
storage	Memory storage for dynamic structures, such as **CvSeq** or **CvGraph** . It is not used for matrices or images.
name	Optional object name. If it is NULL, the first top-level object in the storage will be loaded.

Returns

userdata - An unspecified C pointer to memory
string - An array of characters

The documentation was generated from :

- lua_core.cpp

3.1.102 Log

- **Log**(CvArr src1, CvArr dst)

  Calculates the natural logarithm of every array element's absolute value.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array, it should have `double` type or the same type as the source

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.103 Mahalanobis

- num **Mahalanobis**(CvArr vec1, CvArr vec2, CvArr mat)

  Calculates the Mahalanobis distance between two vectors.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

vec1	The first 1D source vector
vec2	The second 1D source vector
mat	A wrappered OpenCV object of type CvArr

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.104 MatMul

- **MatMul**(CvArr src1, CvArr src2, CvArr dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
src2	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.105 MatMulAdd

- **MatMulAdd**(CvArr src1, CvArr src2, CvArr src3, CvArr dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
src2	A wrappered OpenCV object of type CvArr
src3	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.106 Max

- **Max**(CvArr src1, CvArr src2, CvArr dst)

  Finds per-element maximum of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.107 MaxS

- **MaxS**(CvArr src1, num value, CvArr dst)

  Finds per-element maximum of array and scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	The scalar value
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.108 Merge

- **Merge**(CvArr src0, CvArr src1=nil, CvArr src2=nil, CvArr src3=nil, CvArr dst)

  Composes a multi-channel array from several single-channel arrays or inserts a single channel into the array.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src0	Input channel 0
src1	Input channel 1
src2	Input channel 2
src3	Input channel 3
dst	Destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.109 Min

- **Min**(CvArr src1, CvArr src2, CvArr dst)

  Finds per-element minimum of two arrays.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.110 MinMaxLoc

- num,num **MinMaxLoc**(CvArr arr, CvPoint min_loc=nil, CvPoint max_loc=nil, CvArr mask=nil)

  Finds global minimum and maximum in array or subarray.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	The source array, single-channel or multi-channel with COI set
min_loc	A wrappered OpenCV object of type CvPoint with default value *nil*
max_loc	A wrappered OpenCV object of type CvPoint with default value *nil*
mask	The optional mask used to select a subarray

Returns

num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.111 MinS

- **MinS**(CvArr src1, num value, CvArr dst)

  Finds per-element minimum of an array and a scalar.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	The scalar value
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.112 MixChannels

- **MixChannels**(CvArr src, int src_count, CvArr dst, int dst_count, int[] from_to, int pair_count)

  Copies several channels from input arrays to certain channels of output arrays

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Input arrays
src_count	An integer type number
dst	Destination arrays
dst_count	An integer type number
from_to	An integer type number which is an array
pair_count	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.113 Mul

- **Mul**(CvArr src1, CvArr src2, CvArr dst, num scale=1)

  Calculates the per-element product of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
scale	Optional scale factor

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.114 MulSpectrums

- **MulSpectrums**(CvArr src1, CvArr src2, CvArr dst, int flags)

  Performs per-element multiplication of two Fourier spectrums.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array of the same type and the same size as the source arrays
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.115 MulTransposed

- **MulTransposed**(CvArr src1, CvArr dst, int order, CvArr delta=nil, num scale=1)

  Calculates the product of an array and a transposed array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination matrix. Must be `CV_32F` or `CV_64F`.
order	Order of multipliers
delta	An optional array, subtracted from `src` before multiplication
scale	An optional scaling

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.116 NextGraphItem

- int **NextGraphItem**(CvGraphScanner scanner)

  Executes one or more steps of the graph traversal procedure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

scanner	Graph traversal state. It is updated by this function.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.117 NextNArraySlice

- int **NextNArraySlice**(CvNArrayIterator array_iterator)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

array_iterator	A wrappered OpenCV object of type CvNArrayIterator

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.118 NextTreeNode

- userdata **NextTreeNode**(CvTreeNodeIterator tree_iterator)

  Returns the currently observed node and moves the iterator toward the next node.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tree_iterator	Tree iterator initialized by the function

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.119 Norm

- num **Norm**(CvArr arr1, CvArr arr2=nil, int norm_type=CV_L2, CvArr mask=nil)

  Calculates absolute array norm, absolute difference norm, or relative difference norm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr1	The first source image
arr2	The second source image. If it is NULL, the absolute norm of `arr1` is calculated, otherwise the absolute or relative norm of `arr1-arr2` is calculated.
norm_type	An integer type number with default value *CV_L2*
mask	The optional operation mask

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.120 Normalize

- **Normalize**(CvArr src, CvArr dst, num a=1, num b=0, int norm_type=CV_L2, CvArr mask=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
a	A floating point number with default value *1*
b	A floating point number with default value *0*
norm_type	An integer type number with default value *CV_L2*
mask	A wrappered OpenCV object of type CvArr with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.121 Not

- **Not**(CvArr src1, CvArr dst)

  Performs per-element bit-wise inversion of array elements.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.122 Or

- **Or**(CvArr src1, CvArr src2, CvArr dst, CvArr mask=nil)

  Calculates per-element bit-wise disjunction of two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.123 OrS

- **OrS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

  Calculates a per-element bit-wise disjunction of an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Scalar to use in the operation
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.124 PerspectiveTransform

- **PerspectiveTransform**(CvArr src1, CvArr dst, CvMat mat)

  Performs perspective matrix transformation of a vector array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination three-channel floating-point array
mat	3×3 or 4×4 transformation matrix

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.125 PolarToCart

- **PolarToCart**(CvArr magnitude, CvArr angle, CvArr x, CvArr y, int angle_in_degrees=0)

  Calculates Cartesian coordinates of 2d vectors represented in polar form.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

magnitude	The array of magnitudes. If it is NULL, the magnitudes are assumed to be all 1's.
angle	The array of angles, whether in radians or degrees
x	The destination array of x-coordinates, may be set to NULL if it is not needed
y	The destination array of y-coordinates, mau be set to NULL if it is not needed
angle_in_degrees	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.126 PolyLine

- **PolyLine**(CvArr img, CvPoint[] pts, int[] npts, int contours, int is_closed, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws simple or thick polygons.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image
pts	Array of pointers to polygons
npts	Array of polygon vertex counters
contours	Number of contours that bind the filled region
is_closed	An integer type number
color	Polyline color
thickness	Thickness of the polyline edges
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the vertex coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.127 Pow

- **Pow**(CvArr src1, CvArr dst, num power)

  Raises every array element to a power.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array, should be the same type as the source
power	The exponent of power

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.128 PrevTreeNode

- userdata **PrevTreeNode**(CvTreeNodeIterator tree_iterator)

Returns the currently observed node and moves the iterator toward the previous node.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tree_iterator	Tree iterator initialized by the function

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.129 ProjectPCA

- **ProjectPCA**(CvArr data, CvArr mean, CvArr eigenvals, CvArr result)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

data	A wrappered OpenCV object of type CvArr
mean	A wrappered OpenCV object of type CvArr
eigenvals	A wrappered OpenCV object of type CvArr
result	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.130 Ptr1D

- userdata, int **Ptr1D**(CvArr arr,int idx0, int gettype=0)

Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
idx0	An integer type number
gettype	An integer type number with default value *0*

Returns

userdata - An unspecified C pointer to memory
int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.131 Ptr2D

- userdata, int **Ptr2D**(CvArr arr,int idx0, int idx1, int gettype=0)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
idx0	An integer type number
idx1	An integer type number
gettype	An integer type number with default value *0*

Returns

userdata - An unspecified C pointer to memory
int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.132 Ptr3D

- userdata, int **Ptr3D**(CvArr arr,int idx0, int idx1, int idx2, int gettype=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
idx0	An integer type number
idx1	An integer type number
idx2	An integer type number
gettype	An integer type number with default value *0*

Returns

userdata - An unspecified C pointer to memory
int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.133 PtrND

- userdata, int, int **PtrND**(CvArr arr,int[] idx, int gettype=0, int create_node=1, int gethashval=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
idx	An integer type number which is an array
gettype	An integer type number with default value *0*
create_node	An integer type number with default value *1*
gethashval	An integer type number with default value *0*

Returns

userdata - An unspecified C pointer to memory
int - An integer type number defined by architecture
int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.134 PutText

- **PutText**(CvArr img, string text, CvPoint org, CvFont font, CvScalar color)

  Draws a text string.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Input image
text	String to print
org	Coordinates of the bottom-left corner of the first letter
font	Pointer to the font structure
color	Text color

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.135 RandArr

- **RandArr**(CvRNG rng, CvArr arr, int dist_type, CvScalar param1, CvScalar param2)

  Fills an array with random numbers and updates the RNG state.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

rng	RNG state initialized by **RNG**
arr	The destination array
dist_type	An integer type number
param1	The first parameter of the distribution. In the case of a uniform distribution it is the inclusive lower boundary of the random numbers range. In the case of a normal distribution it is the mean value of the random numbers.
param2	The second parameter of the distribution. In the case of a uniform distribution it is the exclusive upper boundary of the random numbers range. In the case of a normal distribution it is the standard deviation of the random numbers.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.136 RandInt

- int **RandInt**(CvRNG rng)

  Returns a 32-bit unsigned integer and updates RNG.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

rng	RNG state initialized by `RandInit` and, optionally, customized by `RandSetRange` (though, the latter function does not affect the discussed function outcome)

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.137 RandReal

- num **RandReal**(CvRNG rng)

  Returns a floating-point random number and updates RNG.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

rng	RNG state initialized by **RNG**

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.138 RandShuffle

- **RandShuffle**(CvArr mat, CvRNG rng, num iter_factor=1)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	A wrappered OpenCV object of type CvArr
rng	A wrappered OpenCV object of type CvRNG
iter_factor	A floating point number with default value *1*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.139 Rectangle

- **Rectangle**(CvArr img, CvPoint pt1, CvPoint pt2, CvScalar color, int thickness=1, int line_type=8, int shift=0)

  Draws a simple, thick, or filled rectangle.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	Image
pt1	One of the rectangle's vertices
pt2	Opposite rectangle vertex
color	Line color (RGB) or brightness (grayscale image)
thickness	Thickness of lines that make up the rectangle. Negative values, e.g., CV_FILLED, cause the function to draw a filled rectangle.
line_type	An integer type number with default value *8*
shift	Number of fractional bits in the point coordinates

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.140 Reduce

- **Reduce**(CvArr src, CvArr dst, int dim=-1, int op=CV_REDUCE_SUM)

  Reduces a matrix to a vector.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The input matrix.
dst	The output single-row/single-column vector that accumulates somehow all the matrix rows/columns.
dim	The dimension index along which the matrix is reduced. 0 means that the matrix is reduced to a single row, 1 means that the matrix is reduced to a single column and -1 means that the dimension is chosen automatically by analysing the dst size.
op	An integer type number with default value *CV_REDUCE_SUM*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.141 ReleaseGraphScanner

- **ReleaseGraphScanner**(CvGraphScanner scanner)

  Completes the graph traversal procedure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

scanner	Double pointer to graph traverser

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.142 ReleaseImage

- **ReleaseImage**(IplImage image)

  Deallocates the image header and the image data.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Double pointer to the image header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.143 ReleaseImageHeader

- **ReleaseImageHeader**(IplImage image)

  Deallocates an image header.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Double pointer to the image header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.144 ReleaseMat

- **ReleaseMat**(CvMat mat)

  Deallocates a matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	Double pointer to the matrix

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.145 ReleaseMatND

- **ReleaseMatND**(CvMatND mat)

  Deallocates a multi-dimensional array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	Double pointer to the array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.146 ReleaseMemStorage

- **ReleaseMemStorage**(CvMemStorage storage)

  Releases memory storage.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

storage	Pointer to the released storage

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.147 ReleaseSparseMat

- **ReleaseSparseMat**(CvSparseMat mat)

  Deallocates sparse array.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

mat	Double pointer to the array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.148 RemoveNodeFromTree

- **RemoveNodeFromTree**(userdata node, userdata frame)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

node	An unspecified C pointer to memory
frame	An unspecified C pointer to memory

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.149 Repeat

- **Repeat**(CvArr src, CvArr dst)

  Fill the destination array with repeated copies of the source array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array, image or matrix
dst	Destination array, image or matrix

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.150 ResetImageROI

- **ResetImageROI**(IplImage image)

  Resets the image ROI to include the entire image and releases the ROI structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A pointer to the image header

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.151 RestoreMemStoragePos

- **RestoreMemStoragePos**(CvMemStorage storage, CvMemStorage pos)

  Restores memory storage position.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

storage	Memory storage
pos	New storage top position

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.152 Round

- int **Round**(num value)

  Converts a floating-point number to the nearest integer value.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

value	The input floating-point value

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.153 SVBkSb

- **SVBkSb**(CvArr W, CvArr U, CvArr V, CvArr B, CvArr X, int flags)

  Performs singular value back substitution.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

W	Matrix or vector of singular values
U	Left orthogonal matrix (tranposed, perhaps)
V	Right orthogonal matrix (tranposed, perhaps)
B	The matrix to multiply the pseudo-inverse of the original matrix `A` by. This is an optional parameter. If it is omitted then it is assumed to be an identity matrix of an appropriate size (so that `X` will be the reconstructed pseudo-inverse of `A`).
X	The destination matrix: result of back substitution
flags	Operation flags, should match exactly to the `flags` passed to **SVD**

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.154 SVD

- **SVD**(CvArr A, CvArr W, CvArr U=nil, CvArr V=nil, int flags=0)

  Performs singular value decomposition of a real floating-point matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

A	Source $\mathtt{M} \times \mathtt{N}$ matrix
W	Resulting singular value diagonal matrix ($\mathtt{M} \times \mathtt{N}$ or $\min(\mathtt{M},\mathtt{N}) \times \min(\mathtt{M},\mathtt{N})$) or $\min(\mathtt{M},\mathtt{N}) \times 1$ vector of the singular values
U	Optional left orthogonal matrix, $\mathtt{M} \times \min(\mathtt{M},\mathtt{N})$ (when `CV_SVD_U_T` is not set), or $\min(\mathtt{M},\mathtt{N}) \times \mathtt{M}$ (when `CV_SVD_U_T` is set), or $\mathtt{M} \times \mathtt{M}$ (regardless of `CV_SVD_U_T` flag).
V	Optional right orthogonal matrix, $\mathtt{N} \times \min(\mathtt{M},\mathtt{N})$ (when `CV_SVD_V_T` is not set), or $\min(\mathtt{M},\mathtt{N}) \times \mathtt{N}$ (when `CV_SVD_V_T` is set), or $\mathtt{N} \times \mathtt{N}$ (regardless of `CV_SVD_V_T` flag).
flags	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.155 SaveMemStorage

- **SaveMemStorage**(CvMemStorage storage, CvMemStorage pos)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

storage	A wrappered OpenCV object of type CvMemStorage
pos	A wrappered OpenCV object of type CvMemStorage

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.156 ScaleAdd

- **ScaleAdd**(CvArr src1, CvScalar scale, CvArr src2, CvArr dst)

Calculates the sum of a scaled array and another array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
scale	Scale factor for the first array
src2	The second source array
dst	The destination array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.157 SeqElemIdx

- int **SeqElemIdx**(CvSeq seq, userdata element, CvSeqBlock block=nil)

  Returns the index of a specific sequence element.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
element	Pointer to the element within the sequence
block	Optional argument. If the pointer is not `NULL`, the address of the sequence block that contains the element is stored in this location.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.158 SeqInsert

- userdata **SeqInsert**(CvSeq seq, int before_index, userdata element=nil)

  Inserts an element in the middle of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
before_index	An integer type number
element	An unspecified C pointer to memory with default value *nil*

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.159 SeqInsertSlice

- **SeqInsertSlice**(CvSeq seq, int before_index, CvArr from_arr)

  Inserts an array in the middle of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
before_index	An integer type number
from_arr	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.160 SeqInvert

- **SeqInvert**(CvSeq seq)

  Reverses the order of sequence elements.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.161 SeqPop

- **SeqPop**(CvSeq seq, userdata element=nil)

  Removes an element from the end of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
element	Optional parameter . If the pointer is not zero, the function copies the removed element to this location.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.162 SeqPopFront

- **SeqPopFront**(CvSeq seq, userdata element=nil)

  Removes an element from the beginning of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
element	Optional parameter. If the pointer is not zero, the function copies the removed element to this location.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.163 SeqPopMulti

- **SeqPopMulti**(CvSeq seq, userdata[] elements, int count, int in_front=0, string convert_to='')

  Removes several elements from either end of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
elements	Removed elements
count	Number of elements to pop
in_front	An integer type number with default value *0*
convert_to	An array of characters with default value *with default value 0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.164 SeqPush

- userdata **SeqPush**(CvSeq seq, userdata element=nil)

  Adds an element to the end of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
element	Added element

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.165 SeqPushFront

- userdata **SeqPushFront**(CvSeq seq, userdata element=nil)

  Adds an element to the beginning of a sequence.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
element	Added element

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.166 SeqPushMulti

- **SeqPushMulti**(CvSeq seq, userdata[] elements, int count, int in_front=0)

  Pushes several elements to either end of a sequence.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
elements	Added elements
count	Number of elements to push
in_front	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.167 SeqRemove

- **SeqRemove**(CvSeq seq, int index)

  Removes an element from the middle of a sequence.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
index	Index of removed element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.168 SeqRemoveSlice

- **SeqRemoveSlice**(CvSeq seq, CvSlice slice)

  Removes a sequence slice.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
slice	The part of the sequence to remove

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.169 Set

- **Set**(CvArr src, CvScalar value, CvArr mask=nil)

  Sets every element of an array to a given value.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
value	The assigned value
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.170 Set1D

- **Set1D**(CvArr arr, int idx0, CvScalar value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Input array
idx0	An integer type number
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.171 Set2D

- **Set2D**(CvArr arr,int idx0, int idx1, CvScalar value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Input array
idx0	Zero-based element row index
idx1	Zero-based element column index
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.172 Set3D

- **Set3D**(CvArr arr,int idx0, int idx1, int idx2, CvScalar value)

  Set a specific array element.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx0	Zero-based element index
idx1	Zero-based element index
idx2	Zero-based element index
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.173 SetAdd

- int,userdata **SetAdd**(CvSet set_header, CvSetElem elem=nil, int getnewelem_ptr=0)

  Occupies a node in the set.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

set_header	A wrappered OpenCV object of type CvSet
elem	Optional input argument, an inserted element. If not NULL, the function copies the data to the allocated node (the MSB of the first integer field is cleared after copying).
getnewelem_ptr	An integer type number with default value *0*

Returns

int - An integer type number defined by architecture
userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_core.cpp

3.1.174 SetData

- **SetData**(CvArr arr, num[] data, int step, table type)

  Assigns user data to the array header.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Array header
data	User data
step	Full row length in bytes
type	A lua table

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.175 SetIdentity

- **SetIdentity**(CvArr mat, CvScalar value=RealScalar(1))

  Initializes a scaled identity matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	The matrix to initialize (not necesserily square)
value	The value to assign to the diagonal elements

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.176 SetImageCOI

- **SetImageCOI**(IplImage image, int coi)

  Sets the channel of interest in an IplImage.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A pointer to the image header
coi	The channel of interest. 0 - all channels are selected, 1 - first channel is selected, etc. Note that the channel indices become 1-based.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.177 SetImageROI

- **SetImageROI**(IplImage image, CvRect rect)

  Sets an image Region Of Interest (ROI) for a given rectangle.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A pointer to the image header
rect	The ROI rectangle

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.178 SetND

- **SetND**(CvArr arr, int[] idx, CvScalar value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx	An integer type number which is an array
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.179 SetReal1D

- **SetReal1D**(CvArr arr, int idx0, num value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx0	An integer type number
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.180 SetReal2D

- **SetReal2D**(CvArr arr, int idx0, int idx1, num value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Input array
idx0	Zero-based element row index
idx1	Zero-based element column index
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.181 SetReal3D

- **SetReal3D**(CvArr arr, int idx0, int idx1, int idx2, num value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

arr	Input array
idx0	Zero-based element index
idx1	Zero-based element index
idx2	Zero-based element index
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.182 SetRealND

- **SetRealND**(CvArr arr, int[] idx, num value)

  Set a specific array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Input array
idx	An integer type number which is an array
value	The value to assign to the element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.183 SetRemove

- **SetRemove**(CvSet set_header, int index)

  Removes an element from a set.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

set_header	A wrappered OpenCV object of type CvSet
index	Index of the removed element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.184 SetRemoveByPtr

- **SetRemoveByPtr**(CvSet set_header, userdata elem)

  Removes a set element based on its pointer.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

set_header	A wrappered OpenCV object of type CvSet
elem	Removed element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.185 SetSeqBlockSize

- **SetSeqBlockSize**(CvSeq seq, int delta_elems)

  Sets up sequence block size.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
delta_elems	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.186 SetSeqReaderPos

- **SetSeqReaderPos**(CvSeqReader reader, int index, int is_relative=0)

  Moves the reader to the specified position.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

reader	Reader state
index	The destination position. If the positioning mode is used (see the next parameter), the actual position will be `index` mod `reader->seq->total`.
is_relative	If it is not zero, then `index` is a relative to the current position

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.187 SetZero

- **SetZero**(CvArr arr)

  Clears the array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Array to be cleared

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.188 SliceLength

- int **SliceLength**(CvSlice slice, CvSeq seq)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

slice	A wrappered OpenCV object of type CvSlice
seq	A wrappered OpenCV object of type CvSeq

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.189 Solve

- int **Solve**(CvArr src1, CvArr src2, CvArr dst, int method=CV_LU)

  Solves a linear system or least-squares problem.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
src2	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
method	An integer type number with default value *CV_LU*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.190 SolveCubic

- int **SolveCubic**(CvMat coeffs, CvMat roots)

  Finds the real roots of a cubic equation.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

coeffs	The equation coefficients, an array of 3 or 4 elements
roots	The output array of real roots which should have 3 elements

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.191 SolvePoly

- **SolvePoly**(CvMat coeffs , CvMat roots2, int maxiter=20, int fig=100)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

coeffs	A wrappered OpenCV object of type CvMat
roots2	A wrappered OpenCV object of type CvMat
maxiter	An integer type number with default value *20*
fig	An integer type number with default value *100*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.192 Sort

- **Sort**(CvArr src, CvArr dst=nil, CvArr idxmat=nil, int flags=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr with default value *nil*
idxmat	A wrappered OpenCV object of type CvArr with default value *nil*
flags	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.193 Split

- **Split**(CvArr src, CvArr dst0, CvArr dst1, CvArr dst2, CvArr dst3)

  Divides multi-channel array into several single-channel arrays or extracts a single channel from the array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array
dst0	Destination channel 0
dst1	Destination channel 1
dst2	Destination channel 2
dst3	Destination channel 3

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.194 Sqrt

- num **Sqrt**(num value)

  Calculates the square root.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

value	The input floating-point value

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.195 StartAppendToSeq

- **StartAppendToSeq**(CvSeq seq, CvSeqWriter writer)

  Initializes the process of writing data to a sequence.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Pointer to the sequence
writer	Writer state; initialized by the function

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.196 StartReadSeq

- **StartReadSeq**(CvSeq seq, CvSeqReader reader, int reverse=0)

  Initializes the process of sequential reading from a sequence.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq	Sequence
reader	Reader state; initialized by the function
reverse	Determines the direction of the sequence traversal. If `reverse` is 0, the reader is positioned at the first sequence element; otherwise it is positioned at the last element.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.197 StartWriteSeq

- **StartWriteSeq**(int seq_flags, int header_size, int elem_size, CvMemStorage storage, CvSeqWriter writer)

  Creates a new sequence and initializes a writer for it.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

seq_flags	Flags of the created sequence. If the sequence is not passed to any function working with a specific type of sequences, the sequence value may be equal to 0; otherwise the appropriate type must be selected from the list of predefined sequence types.
header_size	Size of the sequence header. The parameter value may not be less than `sizeof(CvSeq)`. If a certain type or extension is specified, it must fit within the base type header.
elem_size	Size of the sequence elements in bytes; must be consistent with the sequence type. For example, if a sequence of points is created (element type `CV_SEQ_ELTYPE_POINT`), then the parameter `elem_size` must be equal to `sizeof(CvPoint)`.
storage	Sequence location
writer	Writer state; initialized by the function

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.198 Sub

- **Sub**(CvArr src1, CvArr src2, CvArr dst, CvArr mask=nil)

  Computes the per-element difference between two arrays.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.199 SubRS

- **SubRS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

  Computes the difference between a scalar and an array.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Scalar to subtract from
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.200 SubS

- **SubS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

  Computes the difference between an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Subtracted scalar
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.201 Transform

- **Transform**(CvArr src1, CvArr dst, CvMat transmat, CvMat shiftvec=nil)

  Performs matrix transformation of every array element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination array
transmat	Transformation matrix
shiftvec	Optional shift vector

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.202 Transpose

- **Transpose**(CvArr src1, CvArr dst)

  Transposes a matrix.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
dst	The destination matrix

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.203 Xor

- **Xor**(CvArr src1, CvArr src2, CvArr dst, CvArr mask=nil)

  Performs per-element bit-wise "exclusive or" operation on two arrays.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src1	The first source array
src2	The second source array
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.204 XorS

- **XorS**(CvArr src1, CvScalar value, CvArr dst, CvArr mask=nil)

Performs per-element bit-wise "exclusive or" operation on an array and a scalar.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src1	A wrappered OpenCV object of type CvArr
value	Scalar to use in the operation
dst	The destination array
mask	Operation mask, 8-bit single channel array; specifies elements of the destination array to be changed

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.205 getrawdata

- userdata,int **getrawdata**(CvArr arr, table type)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
type	A lua table

Returns

userdata - An unspecified C pointer to memory
int - An integer type number defined by architecture

The documentation was generated from :

- lua_core.cpp

3.1.206 iGet

- num[] **iGet**(IplImage image, table type, int row, int col)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type IplImage
type	A lua table
row	An integer type number
col	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.207 iSet

- **iSet**(IplImage image, table type, int row, int col, num[] value)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type IplImage
type	A lua table
row	An integer type number
col	An integer type number
value	A floating point number which is an array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.1.208 mGet

- num[] **mGet**(CvMat mat, table type, int row, int col)

  Returns the particular element of single-channel floating-point matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	Input matrix
type	A lua table
row	The zero-based index of row
col	The zero-based index of column

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_core.cpp

3.1.209 mSet

- **mSet**(CvMat mat, table type, int row, int col, num[] value)

  Returns a specific element of a single-channel floating-point matrix.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mat	The matrix
type	A lua table
row	The zero-based index of row
col	The zero-based index of column
value	The new value of the matrix element

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_core.cpp

3.2 Module calib3d

3.2.1 CalcMatMulDeriv

- **CalcMatMulDeriv**(CvMat A, CvMat B, CvMat dAbda, CvMat daBdb)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

A	A wrappered OpenCV object of type CvMat
B	A wrappered OpenCV object of type CvMat
dAbda	A wrappered OpenCV object of type CvMat
daBdb	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.2 CalibrateCamera2

- num **CalibrateCamera2**(CvMat object_points, CvMat image_points, CvMat point_counts, CvSize image_size, CvMat camera_matrix, CvMat distortion_matrix, CvMat rotation_vectors=nil, CvMat translation_vectors=nil, int flags=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

object_points	A wrappered OpenCV object of type CvMat
image_points	A wrappered OpenCV object of type CvMat
point_counts	A wrappered OpenCV object of type CvMat
image_size	A wrappered OpenCV object of type CvSize
camera_matrix	A wrappered OpenCV object of type CvMat
distortion_matrix	A wrappered OpenCV object of type CvMat
rotation_vectors	A wrappered OpenCV object of type CvMat with default value *nil*
translation_vectors	A wrappered OpenCV object of type CvMat with default value *nil*
flags	An integer type number with default value *0*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.3 CalibrationMatrixValues

- num,num,num,num **CalibrationMatrixValues**(CvMat camera_matrix, CvSize image_size, num aperature_width=0, num aperature_height=0, CvPoint2D64f principal_point=nil)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

camera_matrix	A wrappered OpenCV object of type CvMat
image_size	A wrappered OpenCV object of type CvSize
aperature_width	A floating point number with default value *0*
aperature_height	A floating point number with default value *0*
principal_point	A wrappered OpenCV object of type CvPoint2D64f with default value *nil*

Returns

num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.4 CheckChessboard

- **CheckChessboard**(IplImage src, CvSize size)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type IplImage
size	A wrappered OpenCV object of type CvSize

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.5 ComposeRT

- **ComposeRT**(CvMat _rvec1, CvMat _tvec1, CvMat _rvec2, CvMat _tvec2, CvMat _rvec3, CvMat _tvec3, CvMat dr3dr1=nil, CvMat dr3dt1=nil, CvMat dr3dr2=nil, CvMat dr3dt2=nil, CvMat dt3dr1=nil, CvMat dt3dt1=nil, CvMat dt3dr2=nil, CvMat dt3dt2=ni)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

_rvec1	A wrappered OpenCV object of type CvMat
_tvec1	A wrappered OpenCV object of type CvMat
_rvec2	A wrappered OpenCV object of type CvMat
_tvec2	A wrappered OpenCV object of type CvMat
_rvec3	A wrappered OpenCV object of type CvMat
_tvec3	A wrappered OpenCV object of type CvMat
dr3dr1	A wrappered OpenCV object of type CvMat with default value *nil*
dr3dt1	A wrappered OpenCV object of type CvMat with default value *nil*
dr3dr2	A wrappered OpenCV object of type CvMat with default value *nil*
dr3dt2	A wrappered OpenCV object of type CvMat with default value *nil*
dt3dr1	A wrappered OpenCV object of type CvMat with default value *nil*
dt3dt1	A wrappered OpenCV object of type CvMat with default value *nil*
dt3dr2	A wrappered OpenCV object of type CvMat with default value *nil*
dt3dt2	A wrappered OpenCV object of type CvMat with default value *ni*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.6 ComputeCorrespondEpilines

- **ComputeCorrespondEpilines**(CvMat points, int which_image, CvMat fundamental_matrix, CvMat correspondent_lines)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

points	A wrappered OpenCV object of type CvMat
which_image	An integer type number
fundamental_matrix	A wrappered OpenCV object of type CvMat
correspondent_lines	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.7 ConvertPointsHomogeneous

- **ConvertPointsHomogeneous**(CvMat src, CvMat dst)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	A wrappered OpenCV object of type CvMat
dst	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.8 CorrectMatches

- **CorrectMatches**(CvMat F, CvMat points1, CvMat points2, CvMat new_points1, CvMat new_points2)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

F	A wrappered OpenCV object of type CvMat
points1	A wrappered OpenCV object of type CvMat
points2	A wrappered OpenCV object of type CvMat

new_points1	A wrappered OpenCV object of type CvMat
new_points2	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.9 DecomposeProjectionMatrix

- **DecomposeProjectionMatrix**(CvMat projMatr, CvMat calibMatr, CvMat rotMatr, CvMat posVect, CvMat rotMatrX=nil, CvMat rotMatrY=nil, CvMat rotMatrZ=nil, CvPoint3D64f eurlerAngles=nil)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

projMatr	A wrappered OpenCV object of type CvMat
calibMatr	A wrappered OpenCV object of type CvMat
rotMatr	A wrappered OpenCV object of type CvMat
posVect	A wrappered OpenCV object of type CvMat
rotMatrX	A wrappered OpenCV object of type CvMat with default value *nil*
rotMatrY	A wrappered OpenCV object of type CvMat with default value *nil*
rotMatrZ	A wrappered OpenCV object of type CvMat with default value *nil*
eurlerAngles	A wrappered OpenCV object of type CvPoint3D64f with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.10 DrawChessboardCorners

- **DrawChessboardCorners**(CvArr image, CvSize pattern_size, CvPoint2D32f[] corners, int count, int pattern_was_found)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type CvArr
pattern_size	A wrappered OpenCV object of type CvSize
corners	A wrappered OpenCV object of type CvPoint2D32f which is an array
count	An integer type number
pattern_was_found	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.11 FindChessboardCorners

- int **FindChessboardCorners**(CvArr image, CvSize pattern_size, CvPoint2D32f[] corners, int flags=(CV_CALIB_CB_ADAPTIVE_THRESH+CV_CALIB_CB_NORMALIZE_IMAGE))

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type CvArr
pattern_size	A wrappered OpenCV object of type CvSize
corners	A wrappered OpenCV object of type CvPoint2D32f which is an array
flags	An integer type number with default value *(CV_CALIB_CB_ADAPTIVE_THRESH*

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_calib3d.cpp

3.2.12 FindExtrinsicCameraParams2

- **FindExtrinsicCameraParams2**(CvMat object_points, CvMat image_points, CvMat camera_matrix, CvMat distortoin_coeffs, CvMat rotation_vector, CvMat translation_vector, int use_extrinsic_guess=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

object_points	A wrappered OpenCV object of type CvMat
image_points	A wrappered OpenCV object of type CvMat
camera_matrix	A wrappered OpenCV object of type CvMat
distortoin_coeffs	A wrappered OpenCV object of type CvMat
rotation_vector	A wrappered OpenCV object of type CvMat
translation_vector	A wrappered OpenCV object of type CvMat
use_extrinsic_guess	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.13 FindFundamentalMat

- num **FindFundamentalMat**(CvMat points1, CvMat points2, CvMat fundamental_matrix, int method=CV_FM_RANSAC, num param1=3, num param2=0.99, CvMat status=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

points1	A wrappered OpenCV object of type CvMat
points2	A wrappered OpenCV object of type CvMat
fundamental_matrix	A wrappered OpenCV object of type CvMat
method	An integer type number with default value *CV_FM_RANSAC*
param1	A floating point number with default value *3*
param2	A floating point number with default value *0*
status	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.14 FindHomography

- num **FindHomography**(CvMat src_points, CvMat dst_points, CvMat homography, int method=0, num ransacReprojThreshold=3, CvMat mask=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src_points	A wrappered OpenCV object of type CvMat
dst_points	A wrappered OpenCV object of type CvMat
homography	A wrappered OpenCV object of type CvMat
method	An integer type number with default value *0*

ransacReprojThreshold	A floating point number with default value *3*
mask	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.15 FindStereoCorrespondenceBM

- **FindStereoCorrespondenceBM**(CvArr left, CvArr right, CvArr disparity, CvStereoBMStatestate)

  Computes the disparity map using block matching algorithm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

left	The left single-channel, 8-bit image.
right	The right image of the same size and the same type.
disparity	The output single-channel 16-bit signed, or 32-bit floating-point disparity map of the same size as input images. In the first case the computed disparities are represented as fixed-point numbers with 4 fractional bits (i.e. the computed disparity values are multiplied by 16 and rounded to integers).

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.16 FindStereoCorrespondenceGC

- **FindStereoCorrespondenceGC**(CvMat left, CvMat right, CvMat disparityLeft, CvMat disparityRight, CvStereoBMStatestate, int useDiparityGuess=0)

Computes the disparity map using graph cut-based algorithm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

left	The left single-channel, 8-bit image.
right	The right image of the same size and the same type.
disparityLeft	A wrappered OpenCV object of type CvMat
disparityRight	A wrappered OpenCV object of type CvMat
useDisparityGuess	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.17 GetOptimalNewCameraMatrix

- **GetOptimalNewCameraMatrix**(CvMat camera_matrix, CvMat dist_coeffs, CvSize image_size, num alpha, CvMat new_camera_matrix, CvSize new_image_size=cv.Size(0,0), CvRect valid_pixel_roi=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

camera_matrix	A wrappered OpenCV object of type CvMat
dist_coeffs	A wrappered OpenCV object of type CvMat
image_size	A wrappered OpenCV object of type CvSize
alpha	A floating point number
new_camera_matrix	A wrappered OpenCV object of type CvMat
new_image_size	A wrappered OpenCV object of type CvSize with default value *cv*
valid_pixel_roi	A wrappered OpenCV object of type CvRect with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.18 InitIntrinsicParams2D

- **InitIntrinsicParams2D**(CvMat object_points, CvMat image_points, CvMat npoints, CvSize image_size, CvMat camera_matrix, num aspect_ratio=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

object_points	A wrappered OpenCV object of type CvMat
image_points	A wrappered OpenCV object of type CvMat
npoints	A wrappered OpenCV object of type CvMat
image_size	A wrappered OpenCV object of type CvSize
camera_matrix	A wrappered OpenCV object of type CvMat
aspect_ratio	A floating point number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.19 POSIT

- **POSIT**(CvPOSITObject posit_obj, CvPoint2D32f[] image_points, num focal_lenght, CvTermCriteria criteria,)

  Implements the POSIT algorithm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

posit_obj	A wrappered OpenCV object of type CvPOSITObject
image_points	A wrappered OpenCV object of type CvPoint2D32f which is an array
focal_lenght	A floating point number
criteria	Termination criteria of the iterative POSIT algorithm

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.20 ProjectPoints2

- **ProjectPoints2**(CvMat object_points, CvMat rotation_vector, CvMat translation_vector, CvMat camera_matrix, CvMat distortoin_coeffs, CvMat image_points, CvMat dpdrot=nil, CvMat dpdt=nil, CvMat dpdf=nil, CvMat dpdc=nil, CvMat dpddist=nil, num aspect_ratio=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

object_points	A wrappered OpenCV object of type CvMat
rotation_vector	A wrappered OpenCV object of type CvMat
translation_vector	A wrappered OpenCV object of type CvMat
camera_matrix	A wrappered OpenCV object of type CvMat
distortoin_coeffs	A wrappered OpenCV object of type CvMat
image_points	A wrappered OpenCV object of type CvMat

dpdrot	A wrappered OpenCV object of type CvMat with default value *nil*
dpdt	A wrappered OpenCV object of type CvMat with default value *nil*
dpdf	A wrappered OpenCV object of type CvMat with default value *nil*
dpdc	A wrappered OpenCV object of type CvMat with default value *nil*
dpddist	A wrappered OpenCV object of type CvMat with default value *nil*
aspect_ratio	A floating point number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.21 RANSACUpdateNumIters

- num **RANSACUpdateNumIters**(num p, num err_prob, int model_points, int max_iter)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

p	A floating point number
err_prob	A floating point number
model_points	An integer type number
max_iter	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.22 RQDecomp3x3

- **RQDecomp3x3**(CvMat matrixM, CvMat matrixR, CvMat matrixQ, CvMat matrixQx=nil, CvMat matrixQy=nil, CvMat matrixQz=nil, CvPoint3D64f eurlerAngles=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

matrixM	A wrappered OpenCV object of type CvMat
matrixR	A wrappered OpenCV object of type CvMat
matrixQ	A wrappered OpenCV object of type CvMat
matrixQx	A wrappered OpenCV object of type CvMat with default value *nil*
matrixQy	A wrappered OpenCV object of type CvMat with default value *nil*
matrixQz	A wrappered OpenCV object of type CvMat with default value *nil*
eurlerAngles	A wrappered OpenCV object of type CvPoint3D64f with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.23 ReleasePOSITObject

- **ReleasePOSITObject**(CvPOSITObject posit_obj)

  Deallocates a 3D object structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

posit_obj	A wrappered OpenCV object of type CvPOSITObject

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.24 ReleaseStereoBMState

- **ReleaseStereoBMState**(CvStereoBMState state)

  Releases block matching stereo correspondence structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

state	Double pointer to the released structure.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.25 ReleaseStereoGCState

- **ReleaseStereoGCState**(CvStereoGCState state)

  Releases the state structure of the graph cut-based stereo correspondence algorithm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

state	Double pointer to the released structure.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.26 ReprojectImageTo3D

- **ReprojectImageTo3D**(CvArr disparityImage, CvArr _3dImage, CvMat Q, int handleMissingValues=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

disparityImage	A wrappered OpenCV object of type CvArr
_3dImage	A wrappered OpenCV object of type CvArr
Q	A wrappered OpenCV object of type CvMat
handleMissingValues	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.27 Rodrigues2

- num **Rodrigues2**(CvMat src, CvMat dst, CvMat jacobian=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvMat
dst	A wrappered OpenCV object of type CvMat
jacobian	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.28 StereoCalibrate

- num **StereoCalibrate**(CvMat object_points, CvMat image_points, CvMat camera_matrix1, CvMat dist_coeffs1, CvMat camera_matrix2, CvMat dist_coeffs2, CvSize size, CvMat R, CvMat T, CvMat E=nil, CvMat P2, CvMat F=nil, CvTermCriteria term_crit=cv.TermCriteria(CV_TERMCRIT_ITER+CV_TERMCRIT_EPS,30,0,000006), int flags=CV_CALIB_FIX_INTRINSIC)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

object_points	A wrappered OpenCV object of type CvMat
image_points	A wrappered OpenCV object of type CvMat
camera_matrix1	A wrappered OpenCV object of type CvMat
dist_coeffs1	A wrappered OpenCV object of type CvMat
camera_matrix2	A wrappered OpenCV object of type CvMat
dist_coeffs2	A wrappered OpenCV object of type CvMat
size	A wrappered OpenCV object of type CvSize
R	A wrappered OpenCV object of type CvMat
T	A wrappered OpenCV object of type CvMat
E	A wrappered OpenCV object of type CvMat with default value *nil*
P2	A wrappered OpenCV object of type CvMat
F	A wrappered OpenCV object of type CvMat with default value *nil*
term_crit	A wrappered OpenCV object of type CvTermCriteria with default value *cv*
flags	An integer type number with default value *CV_CALIB_FIX_INTRINSIC*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.29 StereoRectify

- **StereoRectify**(CvMat camera_matrix1, CvMat camera_matrix2, CvMat dist_coeffs1, CvMat dist_coeffs2, CvSize image_size, CvMat R, CvMat T, CvMat R1, CvMat R2, CvMat P1, CvMat P2, CvMat Q=nil, int flags=CV_CALIB_ZERO_DISPARITY, num alpha=-1, CvSize new_image_size=cv.Size(0,0), CvRect valid_pix_roi1=nil, CvRect valid_pix_roi2=nil)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

camera_matrix1	A wrappered OpenCV object of type CvMat
camera_matrix2	A wrappered OpenCV object of type CvMat
dist_coeffs1	A wrappered OpenCV object of type CvMat
dist_coeffs2	A wrappered OpenCV object of type CvMat
image_size	A wrappered OpenCV object of type CvSize
R	A wrappered OpenCV object of type CvMat
T	A wrappered OpenCV object of type CvMat
R1	A wrappered OpenCV object of type CvMat
R2	A wrappered OpenCV object of type CvMat
P1	A wrappered OpenCV object of type CvMat
P2	A wrappered OpenCV object of type CvMat
Q	A wrappered OpenCV object of type CvMat with default value *nil*
flags	An integer type number with default value *CV_CALIB_ZERO_DISPARITY*
alpha	A floating point number with default value *with default value CV_CALIB_ZERO_DISPARITY*
new_image_size	A wrappered OpenCV object of type CvSize with default value *cv*
valid_pix_roi1	A wrappered OpenCV object of type CvRect with default value *nil*
valid_pix_roi2	A wrappered OpenCV object of type CvRect with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.30 StereoRectifyUncalibrated

- num **StereoRectifyUncalibrated**(CvMat points1, CvMat points2, CvMat F, CvSize img_size, CvMat H1, CvMat T, CvMat H2, num threshold=5)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

points1	A wrappered OpenCV object of type CvMat
points2	A wrappered OpenCV object of type CvMat
F	A wrappered OpenCV object of type CvMat
img_size	A wrappered OpenCV object of type CvSize
H1	A wrappered OpenCV object of type CvMat
T	A wrappered OpenCV object of type CvMat
H2	A wrappered OpenCV object of type CvMat
threshold	A floating point number with default value *5*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_calib3d.cpp

3.2.31 TriangulatePoints

- **TriangulatePoints**(CvMat projMatr1, CvMat projMatr2, CvMat projPoints1, CvMat projPoints2, CvMat points4D)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

projMatr1	A wrappered OpenCV object of type CvMat
projMatr2	A wrappered OpenCV object of type CvMat
projPoints1	A wrappered OpenCV object of type CvMat
projPoints2	A wrappered OpenCV object of type CvMat
points4D	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.2.32 ValidateDisparity

- **ValidateDisparity**(CvArr disparity, CvArr cost, int minDisparity, int numberOfDisparities, int disp12MaxDiff=1)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

disparity	A wrappered OpenCV object of type CvArr
cost	A wrappered OpenCV object of type CvArr
minDisparity	An integer type number
numberOfDisparities	An integer type number
disp12MaxDiff	An integer type number with default value *1*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_calib3d.cpp

3.3 Module highgui

3.3.1 AddText

- **AddText**(CvArrimg, string text, CvPoint org, CvFont arg2)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

text	An array of characters
org	A wrappered OpenCV object of type CvPoint
arg2	A wrappered OpenCV object of type CvFont

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.2 ConvertImage

- **ConvertImage**(CvArr src, CvArr dst, int flags=0

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	Source image.
dst	Destination image. Must be single-channel or 3-channel 8-bit image.
flags	An integer type number with default value

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.3 CreateTrackbar

- int **CreateTrackbar**(string tbarname, string wname, int count, func on_change(int pos))

Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tbarname	An array of characters
wname	An array of characters
count	Maximal position of the slider. Minimal position is always 0.
on_change(int	A lua function

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.4 DestroyAllWindows

- **DestroyAllWindows**()

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

none	function doesn't have input arguments.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.5 DestroyWindow

- **DestroyWindow**(string name)

Destroys a window.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	Name of the window to be destroyed.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.6 DisplayOverlay

- **DisplayOverlay**(string name, string text, int delayms)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters
text	An array of characters
delayms	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.7 DisplayStatusBar

- **DisplayStatusBar**(string name, string text, int delayms)

Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters
text	An array of characters
delayms	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.8 FOURCC

- num CV_**FOURCC**(string[4] codec)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_highgui.cpp

3.3.9 GetCaptureDomain

- int **GetCaptureDomain**(CvCapture capture)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

capture	A wrappered OpenCV object of type CvCapture

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.10 GetCaptureProperty

- num **GetCaptureProperty**(CvCapture capture, int property_id)

  Gets video capturing properties.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

capture	video capturing structure.
property_id	Property identifier. Can be one of the following:

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_highgui.cpp

3.3.11 GetTrackbarPos

- int **GetTrackbarPos**(string trackbar_name, string window_name)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

trackbar_name	An array of characters
window_name	An array of characters

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.12 GetWindowHandle

- userdata **GetWindowHandle**(string name)

  Gets the window's handle by its name.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_highgui.cpp

3.3.13 GetWindowName

- string **GetWindowName**(userdata handle)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

handle	An unspecified C pointer to memory

Returns

string - An array of characters

The documentation was generated from :

- lua_highgui.cpp

3.3.14 GetWindowProperty

- num **GetWindowProperty**(string name, int prop_id)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

name	An array of characters
prop_id	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_highgui.cpp

3.3.15 GrabFram

- int **GrabFram**(CvCapture capture)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

capture	A wrappered OpenCV object of type CvCapture

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.16 InitSystem

- int **InitSystem**(int argc, string[] argv)

  Initializes HighGUI.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

argc	Number of command line arguments
argv	Array of command line arguments

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.17 LoadWindowParameters

- **LoadWindowParameters**(string name)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.18 MoveWindow

- **MoveWindow**(string name,int x, int y)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	Name of the window to be moved.
x	New x coordinate of the top-left corner
y	New y coordinate of the top-left corner

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.19 NamedWindow

- int **NamedWindow**(string name, int flags=CV_WINDOW_AUTOSIZE)

  Creates a window.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	Name of the window in the window caption that may be used as a window identifier.
flags	Flags of the window. Currently the only supported flag is `CV_WINDOW_AUTOSIZE`. If this is set, window size is automatically adjusted to fit the displayed image (see **ShowImage**), and the user can not change the window size manually.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.20 ReleaseCapture

- **ReleaseCapture**(CvCapture capture)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

capture	Pointer to video the capturing structure.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.21 ReleaseVideoWriter

- **ReleaseVideoWriter**(CvVideoWriter writer)

  Releases the AVI writer.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

writer	Pointer to the video file writer structure.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.22 ResizeWindow

- **ResizeWindow**(string name, int width, int height)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	Name of the window to be resized.
width	New width
height	New height

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.23 SaveImage

- int **SaveImage**(string filename, CvArr image, int[] params)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

filename	Name of the file.
image	Image to be saved.
params	An integer type number which is an array

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.24 SaveWindowParameters

- **SaveWindowParameters**(string name)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.25 SetCaptureProperty

- int **SetCaptureProperty**(CvCapture capture, int property_id, num value)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

capture	video capturing structure.
property_id	property identifier. Can be one of the following:
value	value of the property.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.26 SetMouseCallback

- **SetMouseCallback**(string wname, func on_mouse(int event,int x,int y, int flags)

  Assigns callback for mouse events.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

wname	An array of characters
on_mouse(int	A lua function
x	An integer type number
y	An integer type number
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.27 SetTrackbarPos

- **SetTrackbarPos**(string trackbar_name, string window_name, int pos)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

trackbar_name	An array of characters
window_name	An array of characters
pos	New position.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.28 SetWindowProperty

- **SetWindowProperty**(string name, int prop_id, num prop_value)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	An array of characters
prop_id	An integer type number
prop_value	A floating point number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.29 ShowImage

- **ShowImage**(string name, CvArr image)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

name	Name of the window.
image	Image to be shown.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_highgui.cpp

3.3.30 StartWindowThread

- int **StartWindowThread**()

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

none	function doesn't have input arguments.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.31 WaitKey

- int **WaitKey**(int delay=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

delay	Delay in milliseconds.

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.3.32 WriteFrame

- int **WriteFrame**(CvVideoWriter writer, IplImage image)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

writer	Video writer structure
image	The written frame

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_highgui.cpp

3.4 Module imgproc

3.4.1 Acc

- **Acc**(CvArr image, CvArr sum, CvArr mask=nil)

  Adds a frame to an accumulator.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input image, 1- or 3-channel, 8-bit or 32-bit floating point. (each channel of multi-channel image is processed independently)
sum	Accumulator with the same number of channels as input image, 32-bit or 64-bit floating-point
mask	Optional operation mask

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.2 AdaptiveThreshold

- **AdaptiveThreshold**(CvArr src, CvArr dst, num max_values, int adaptive_method=CV_ADAPTIVE_THRESH_MEAN_C, int threshold_type=CV_THRESH_BINARY, int block_size=3, num param1=5)

  Applies an adaptive threshold to an array.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
max_values	A floating point number
adaptive_method	Adaptive thresholding algorithm to use: `CV_ADAPTIVE_THRESH_MEAN_C` or `CV_ADAPTIVE_THRESH_GAUSSIAN_C` (see the discussion)
threshold_type	An integer type number with default value *CV_THRESH_BINARY*
block_size	An integer type number with default value *3*
param1	The method-dependent parameter. For the methods `CV_ADAPTIVE_THRESH_MEAN_C` and `CV_ADAPTIVE_THRESH_GAUSSIAN_C` it is a constant subtracted from the mean or weighted mean (see the discussion), though it may be negative

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.3 ArcLength

- num **ArcLength**(CvArr contour, CvSlice slice=CV_WHOLE_SEQ, int is_closed=-1)

  Calculates the contour perimeter or the curve length.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

contour	A wrappered OpenCV object of type CvArr
slice	Starting and ending points of the curve, by default, the whole curve length is calculated
is_closed	An integer type number with default value *with default value CV_WHOLE_SEQ*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.4 BoxPoints

- **BoxPoints**(CvBox2D box, CvPoint2D32f[] pt)

  Finds the box vertices.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

box	Box
pt	A wrappered OpenCV object of type CvPoint2D32f which is an array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.5 CalcArrBackProject

- **CalcArrBackProject**(IplImage image, CvArr dst, CvHistogram hist)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type IplImage
dst	A wrappered OpenCV object of type CvArr
hist	A wrappered OpenCV object of type CvHistogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.6 CalcArrBackProjectPatch

- **CalcArrBackProjectPatch**(IplImage image, CvArr dst, CvSize range, CvHistogram hist, int method, num factor)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type IplImage
dst	A wrappered OpenCV object of type CvArr

range	A wrappered OpenCV object of type CvSize
hist	A wrappered OpenCV object of type CvHistogram
method	An integer type number
factor	A floating point number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.7 CalcArrHist

- **CalcArrHist**(CvArr arr, CvHistogram hist, int accumulate=0, CvArr mask=NULL)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	A wrappered OpenCV object of type CvArr
hist	A wrappered OpenCV object of type CvHistogram
accumulate	An integer type number with default value *0*
mask	A wrappered OpenCV object of type CvArr with default value *NULL*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.8 CalcBayesianProb

- **CalcBayesianProb**(CvHistogram src, int number, CvHistogram dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvHistogram
number	An integer type number
dst	A wrappered OpenCV object of type CvHistogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.9 CalcHist

- **CalcHist**(CvArr image, CvHistogram hist, int accumulate=0, CvArr mask=NULL)

Calculates the histogram of image(s).

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Source images (though you may pass CvMat** as well)
hist	Pointer to the histogram
accumulate	Accumulation flag. If it is set, the histogram is not cleared in the beginning. This feature allows user to compute a single histogram from several images, or to update the histogram online
mask	The operation mask, determines what pixels of the source images are counted

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.10 CalcProbDensity

- **CalcProbDensity**(CvHistogram hist1,CvHistogram hist2,CvHistogram dst_hist, num scale=255)

  Divides one histogram by another.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

hist1	first histogram (the divisor)
hist2	second histogram
dst_hist	destination histogram
scale	scale factor for the destination histogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.11 CalcSubdivVoronoi2D

- **CalcSubdivVoronoi2D**(CvSubdiv2D subdiv)

  Calculates the coordinates of Voronoi diagram cells.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

subdiv	Delaunay subdivision, in which all the points are already added

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.12 Canny

- **Canny**(CvArr image, CvArr edges, num threshold1, num threshold2, int aperture_size=3)

  Finds edges in an image using Canny algorithm.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Single-channel 8-bit input image
edges	The output edge map. It will have the same size and the same type as `image`
threshold1	The first threshold for the hysteresis procedure
threshold2	The second threshold for the hysteresis procedure
aperture_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.13 CheckContourConvexity

- int **CheckContourConvexity**(CvArr contour)

  Tests contour convexity.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

contour	Tested contour (sequence or array of points)

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_imgproc.cpp

3.4.14 ClearHist

- **ClearHist**(CvHistogram hist)

  Clears the histogram.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

hist	Histogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.15 ClearSubdivVoronoi2D

- **ClearSubdivVoronoi2D**(CvSubdiv2D subdiv)

  Removes all virtual points.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

subdiv	Delaunay subdivision

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.16 CompareHist

- num **CompareHist**(CvHistogram hist1, CvHistogram hist2, int method)

  Compares two dense histograms.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

hist1	The first dense histogram
hist2	The second dense histogram
method	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.17 ContourArea

- num **ContourArea**(CvArr contour, CvSlice slice=CV_WHOLE_SEQ, int oriented=0)

  Calculates the area of a whole contour or a contour section.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

contour	Contour (sequence or array of vertices)
slice	Starting and ending points of the contour section of interest, by default, the area of the whole contour is calculated
oriented	An integer type number with default value *0*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.18 ContourPerimetere

- num **ContourPerimetere**(CvArr contour)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

contour	A wrappered OpenCV object of type CvArr

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.19 ConvertMaps

- **ConvertMaps**(CvArr mapx, CvArr mapy, CvArr mapxy, CvArr mapalpha)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mapx	A wrappered OpenCV object of type CvArr
mapy	A wrappered OpenCV object of type CvArr
mapxy	A wrappered OpenCV object of type CvArr
mapalpha	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.20 CopyHist

- **CopyHist**(CvHistogram src, CvHistogram dst)

  Copies a histogram.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source histogram
dst	Pointer to destination histogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.21 CopyMakeBorder

- **CopyMakeBorder**(CvArr src, CvArr dst, CvPoint offset, int bordertyp, CvScalar value=ScallarAll(0))

  Copies an image and makes a border around it.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source image
dst	The destination image
offset	Coordinates of the top-left corner (or bottom-left in the case of images with bottom-left origin) of the destination image rectangle where the source image (or its ROI) is copied. Size of the rectanlge matches the source image size/ROI size
bordertyp	An integer type number
value	Value of the border pixels if `bordertype` is `IPL_BORDER_CONSTANT`

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.22 CorenerEigenValsAndVecs

- **CorenerEigenValsAndVecs**(CvArr image, CvArr eigenvv, int block_size, int aperature_size=3)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type CvArr
eigenvv	A wrappered OpenCV object of type CvArr
block_size	An integer type number
aperature_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.23 CorenerHarris

- **CorenerHarris**(CvArr image, CvArr harris_responce, int block_size, int aperature_size=3, num k=0.04)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type CvArr
harris_responce	A wrappered OpenCV object of type CvArr
block_size	An integer type number
aperature_size	An integer type number with default value *3*
k	A floating point number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.24 CreatePyramid

- userdata **CreatePyramid**(CvArr img, int extra_layers, num rate, CvSize[] layer_sizes=nil, CvArr bufarr=nil, int calc=0, int filter=CV_GAUSSIAN_5x5)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	A wrappered OpenCV object of type CvArr
extra_layers	An integer type number
rate	A floating point number
layer_sizes	A wrappered OpenCV object of type CvSize which is an array with default value *nil*
bufarr	A wrappered OpenCV object of type CvArr with default value *nil*
calc	An integer type number with default value *0*
filter	An integer type number with default value *CV_GAUSSIAN_5x5*

Returns

userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_imgproc.cpp

3.4.25 CreateSubdivDelaunay2D

- **CreateSubdivDelaunay2D**(CvRect rect, CvMemStorage storage)

  Creates an empty Delaunay triangulation.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

rect	Rectangle that includes all of the 2d points that are to be added to the subdivision
storage	Container for subdivision

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.26 CvtColor

- **CvtColor**(CvArr src, CvArr dst, int code)

  Converts an image from one color space to another.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source 8-bit (8u), 16-bit (16u) or single-precision floating-point (32f) image
dst	The destination image of the same data type as the source. The number of channels may be different
code	Color conversion operation that can be specifed using `CV_` *`src_color_space 2 dst_color_space`* constants (see below)

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.27 Dilate

- **Dilate**(CvArr src, CvArr dst, IplConvKernel element=nil, int iterations=1)

  Dilates an image by using a specific structuring element.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	Source image
dst	Destination image
element	A wrappered OpenCV object of type IplConvKernel with default value *nil*
iterations	Number of times dilation is applied

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.28 DistTransfomr

- **DistTransfomr**(CvArr src, CvArr dst, int distance_type=CV_DIST_L2, int mask_size=3, num[] mask=nil, CvArr labels=0)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
distance_type	An integer type number with default value *CV_DIST_L2*
mask_size	An integer type number with default value *3*
mask	A floating point number which is an array with default value *nil*
labels	A wrappered OpenCV object of type CvArr with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.29 EqualizeHist

- **EqualizeHist**(CvArr src, CvArr dst)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.30 Erode

- **Erode**(CvArr src, CvArr dst, IplConvKernel element=nil, int iterations=1)

  Erodes an image by using a specific structuring element.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
element	A wrappered OpenCV object of type IplConvKernel with default value *nil*
iterations	Number of times erosion is applied

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.31 Filter2D

- **Filter2D**(CvArr src, CvArr dst, CvMat kernel, CvPoint anchor=Point(-1,-1))

  Convolves an image with the kernel.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source image
dst	The destination image
kernel	Convolution kernel, a single-channel floating point matrix. If you want to apply different kernels to different channels, split the image into separate color planes using **Split** and process them individually
anchor	The anchor of the kernel that indicates the relative position of a filtered point within the kernel. The anchor shoud lie within the kernel. The special default value (-1,-1) means that it is at the kernel center

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.32 FindContours

- int,CvSeq **FindContours**(CvArr image, CvMemStorage storage, int header_size=CvContour[s̈ize]̈, int mode=CV_RETR_LISTS, int method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset=Point(0,0))

  Finds the contours in a binary image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	The source, an 8-bit single channel image. Non-zero pixels are treated as 1's, zero pixels remain 0's - the image is treated as `binary`. To get such a binary image from grayscale, one may use **Threshold**, **AdaptiveThreshold** or **Canny**. The function modifies the source image's content
storage	Container of the retrieved contours
header_size	An integer type number with default value *CvContour*
mode	An integer type number with default value *CV_RETR_LISTS*
method	An integer type number with default value *CV_CHAIN_APPROX_SIMPLE*
offset	Offset, by which every contour point is shifted. This is useful if the contours are extracted from the image ROI and then they should be analyzed in the whole image context

Returns

int - An integer type number defined by architecture
CvSeq - A wrappered OpenCV object of type CvSeq

The documentation was generated from :

- lua_imgproc.cpp

3.4.33 FindCornerSubPix

- **FindCornerSubPix**(CvArr image, CvPoint2D32f[] corners, CvSize win, CvSize zero_none, CvTermCriteria criteria)

  Refines the corner locations.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input image
corners	Initial coordinates of the input corners as a list of (x, y) pairs
win	Half of the side length of the search window. For example, if win=(5,5), then a $5*2+1 \times 5*2+1 = 11 \times 11$ search window would be used
zero_none	A wrappered OpenCV object of type CvSize
criteria	Criteria for termination of the iterative process of corner refinement. That is, the process of corner position refinement stops either after a certain number of iterations or when a required accuracy is achieved. The criteria may specify either of or both the maximum number of iteration and the required accuracy

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.34 FindFeatures

- **FindFeatures**(CvFeatureTree tr, CvMat query_points, CvMat indices, CvMat dist, int k, int emax=20)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tr	A wrappered OpenCV object of type CvFeatureTree
query_points	A wrappered OpenCV object of type CvMat
indices	A wrappered OpenCV object of type CvMat
dist	A wrappered OpenCV object of type CvMat
k	An integer type number
emax	An integer type number with default value *20*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.35 FindFeaturesBoxed

- int **FindFeaturesBoxed**(CvFeatureTree tr, CvMat bounds_im, CvMat bounds_max, CvMat out_indices)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tr	A wrappered OpenCV object of type CvFeatureTree
bounds_im	A wrappered OpenCV object of type CvMat
bounds_max	A wrappered OpenCV object of type CvMat
out_indices	A wrappered OpenCV object of type CvMat

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_imgproc.cpp

3.4.36 FitLine

- **FitLine**(CvArr points, int dist_type, num param, num reps, num aeps, num[] line)

  Fits a line to a 2D or 3D point set.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

points	Sequence or array of 2D or 3D points with 32-bit integer or floating-point coordinates
dist_type	The distance used for fitting (see the discussion)
param	Numerical parameter (`C`) for some types of distances, if 0 then some optimal value is chosen

reps	Sufficient accuracy for the radius (distance between the coordinate origin and the line). 0.01 is a good default value.
aeps	Sufficient accuracy for the angle. 0.01 is a good default value.
line	A floating point number which is an array

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.37 FloodFill

- **FloodFill**(CvArr image, CvPoint seed_point, CvScalar new_val, CvScalar lo_diff=ScalarAll(0), CvScalar up_diff=ScalarAll(0), CvConnectedComp comp=nil, int flags=4, CvArr mask=nil)

  Fills a connected component with the given color.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input 1- or 3-channel, 8-bit or floating-point image. It is modified by the function unless the `CV_FLOODFILL_MASK_ONLY` flag is set (see below)
seed_point	The starting point
new_val	New value of the repainted domain pixels
lo_diff	Maximal lower brightness/color difference between the currently observed pixel and one of its neighbors belonging to the component, or a seed pixel being added to the component. In the case of 8-bit color images it is a packed value
up_diff	Maximal upper brightness/color difference between the currently observed pixel and one of its neighbors belonging to the component, or a seed pixel being added to the component. In the case of 8-bit color images it is a packed value
comp	Returned connected component for the repainted domain. Note that the function does not fill `comp->contour` field. The boundary of the filled component can be retrieved from the output mask image using **FindContours**
flags	An integer type number with default value *4*

mask	Operation mask, should be a single-channel 8-bit image, 2 pixels wider and 2 pixels taller than `image`. If not NULL, the function uses and updates the mask, so the user takes responsibility of initializing the `mask` content. Floodfilling can't go across non-zero pixels in the mask, for example, an edge detector output can be used as a mask to stop filling at edges. It is possible to use the same mask in multiple calls to the function to make sure the filled area do not overlap. **Note**: because the mask is larger than the filled image, a pixel in `mask` that corresponds to (x,y) pixel in `image` will have coordinates $(x+1,y+1)$

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.38 GetCentralMoment

- num **GetCentralMoment**(CvMoments moments, int x_order, int y_order)

  Retrieves the central moment from the moment state structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

moments	Pointer to the moment state structure
x_order	x order of the retrieved moment, $\texttt{x\_order} >= 0$
y_order	y order of the retrieved moment, $\texttt{y\_order} >= 0$ and $\texttt{x\_order} + \texttt{y\_order} <= 3$

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.39 GetHuMoments

- **GetHuMoments**(CvMoments moments,CvHuMoments hu_moments)

Calculates the seven Hu invariants.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

moments	The input moments, computed with **Moments**
hu_moments	A wrappered OpenCV object of type CvHuMoments

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.40 GetMinMaxHistValue

- float,float,int int **GetMinMaxHistValue**(CvHistogram hist)

  Finds the minimum and maximum histogram bins.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

hist	Histogram

Returns

float - A floating point number which must be float
float - A floating point number which must be float
int - An integer type number defined by architecture
int - An integer type number defined by architecture

The documentation was generated from :

- lua_imgproc.cpp

3.4.41 GetNormalizedCentralMoment

- num **GetNormalizedCentralMoment**(CvMoments moments, int x_order, int y_order)

  Retrieves the normalized central moment from the moment state structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

moments	Pointer to the moment state structure
x_order	x order of the retrieved moment, $\mathtt{x\_order} >= 0$
y_order	y order of the retrieved moment, $\mathtt{y\_order} >= 0$ and $\mathtt{x\_order} + \mathtt{y\_order} <= 3$

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.42 GetQuadrangleSubPix

- **GetQuadrangleSubPix**(CvArr src, CvArr dst, CvMat map_matrix)

  Retrieves the pixel quadrangle from an image with sub-pixel accuracy.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Extracted quadrangle
map_matrix	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.43 GetRectSubPix

- **GetRectSubPix**(CvArr src, CvArr dst, CvPoint2D32f center)

  Retrieves the pixel rectangle from an image with sub-pixel accuracy.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Extracted rectangle
center	Floating point coordinates of the extracted rectangle center within the source image. The center must be inside the image

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.44 GetSpatialMoment

- num **GetSpatialMoment**(CvMoments moments, int x_order, int y_order)

  Retrieves the spatial moment from the moment state structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

moments	The moment state, calculated by **Moments**
x_order	x order of the retrieved moment, $\mathtt{x\_order} >= 0$
y_order	y order of the retrieved moment, $\mathtt{y\_order} >= 0$ and $\mathtt{x\_order} + \mathtt{y\_order} <= 3$

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.45 GoodFeaturesToTrack

- int **GoodFeaturesToTrack**(CvArr image, CvArr eigen_image, CvArr temp_image, CvPoint2D32f[] corners, int count, num quality_level, num min_distance, CvMat mask=nil, int block_size=3, int use_harris=0, num k=0.04)

  Determines strong corners on an image.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	The source 8-bit or floating-point 32-bit, single-channel image
eigen_image	A wrappered OpenCV object of type CvArr
temp_image	A wrappered OpenCV object of type CvArr
corners	Output parameter; detected corners
count	An integer type number
quality_level	A floating point number
min_distance	A floating point number
mask	Region of interest. The function selects points either in the specified region or in the whole image if the mask is NULL
block_size	An integer type number with default value *3*
use_harris	An integer type number with default value *0*
k	Free parameter of Harris detector; used only if $(\mathtt{useHarris}! = 0)$

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_imgproc.cpp

3.4.46 HoughLines2

- **HoughLines2**(CvArr image, userdata line_storage, int method, num rho, num theta, int threshold, num param1=0, param2=0)

Finds lines in a binary image using a Hough transform.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	The 8-bit, single-channel, binary source image. In the case of a probabilistic method, the image is modified by the function
line_storage	An unspecified C pointer to memory
method	An integer type number
rho	Distance resolution in pixel-related units
theta	Angle resolution measured in radians
threshold	Threshold parameter. A line is returned by the function if the corresponding accumulator value is greater than `threshold`
param1	A floating point number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.47 InitSubdivDelaunay2D

- **InitSubdivDelaunay2D**(CvSubdiv2D subdiv, CvRect rect)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

subdiv	A wrappered OpenCV object of type CvSubdiv2D
rect	A wrappered OpenCV object of type CvRect

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.48 InitUndistortMap

- **InitUndistortMap**(CvMat camera_matrix, CvMat distortion_coeffs, CvArr mapx, CvArr mapy)

  Computes an undistortion map.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

camera_matrix	A wrappered OpenCV object of type CvMat
distortion_coeffs	A wrappered OpenCV object of type CvMat
mapx	A wrappered OpenCV object of type CvArr
mapy	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.49 InitUndistortRectifyMap

- **InitUndistortRectifyMap**(CvMat camera_matrix, CvMat distortion_coeffs, CvMat R, CvMat new_camera_matrix, CvArr mapx, CvArr mapy)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

camera_matrix	A wrappered OpenCV object of type CvMat
distortion_coeffs	A wrappered OpenCV object of type CvMat
R	A wrappered OpenCV object of type CvMat

new_camera_matrix	A wrappered OpenCV object of type CvMat
mapx	A wrappered OpenCV object of type CvArr
mapy	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.50 Inpaint

- **Inpaint**(CvArr src, CvArr inpaint_mask, CvArr dst, num inpaintRange, int flags)

  Inpaints the selected region in the image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	The input 8-bit 1-channel or 3-channel image.
inpaint_mask	A wrappered OpenCV object of type CvArr
dst	The output image of the same format and the same size as input.
inpaintRange	A floating point number
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.51 Integral

- **Integral**(CvArr image, CvArr sum, CvArr sqsum=nil, CvArr tilted_sum=nil)

Calculates the integral of an image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	The source image, $W \times H$, 8-bit or floating-point (32f or 64f)
sum	The integral image, $(W+1) \times (H+1)$, 32-bit integer or double precision floating-point (64f)
sqsum	The integral image for squared pixel values, $(W+1) \times (H+1)$, double precision floating-point (64f)
tilted_sum	A wrappered OpenCV object of type CvArr with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.52 LSHAdd

- **LSHAdd**(CvLSH lsh, CvMat data, CvMat indices=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

lsh	A wrappered OpenCV object of type CvLSH
data	A wrappered OpenCV object of type CvMat
indices	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.53 LSHRemove

- **LSHRemove**(CvLSH lsh, CvMat indices)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

lsh	A wrappered OpenCV object of type CvLSH
indices	A wrappered OpenCV object of type CvMat

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.54 Laplace

- **Laplace**(CvArr src, CvArr dst, int aperature_size=3)

  Calculates the Laplacian of an image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
aperature_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.55 LinearPolar

- **LinearPolar**(CvArr src, CvArr dst, CvPoint2D32f center, num maxRadius, int flags=CV_INTER_LINEAR+CV_WARP_FILL_OUTLIERS)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
center	A wrappered OpenCV object of type CvPoint2D32f
maxRadius	A floating point number
flags	An integer type number with default value *CV_INTER_LINEAR*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.56 LogPolar

- **LogPolar**(CvArr src, CvArr dst, CvPoint2D32f center, num M, int flags=CV_INTER_LINEAR+CV_WARP_FI

  Remaps an image to log-polar space.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
center	The transformation center; where the output precision is maximal
M	Magnitude scale parameter. See below
flags	An integer type number with default value *CV_INTER_LINEAR*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.57 MatchShapes

- num **MatchShapes**(userdata object1, userdata object2, int method, num parameter=0)

  Compares two shapes.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

object1	First contour or grayscale image
object2	Second contour or grayscale image
method	An integer type number
parameter	Method-specific parameter (is not used now)

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.58 MatchTemplate

- **MatchTemplate**(CvArr image, CvArr templ, CvArr result, int method)

  Compares a template against overlapped image regions.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Image where the search is running; should be 8-bit or 32-bit floating-point
templ	Searched template; must be not greater than the source image and the same data type as the image
result	A wrappered OpenCV object of type CvArr
method	Specifies the way the template must be compared with the image regions (see below)

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.59 MinEnclosingCircle

- int,num **MinEnclosingCircle**(CvArr points, CvPoint2D32f center)

  Finds the circumscribed circle of minimal area for a given 2D point set.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

points	Sequence or array of 2D points
center	Output parameter; the center of the enclosing circle

Returns

int - An integer type number defined by architecture
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.60 Moments

- **Moments**(CvArr arr, CvMoments moments, int binary=0)

  Calculates all of the moments up to the third order of a polygon or rasterized shape.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

arr	Image (1-channel or 3-channel with COI set) or polygon (CvSeq of points or a vector of points)
moments	Pointer to returned moment's state structure
binary	(For images only) If the flag is non-zero, all of the zero pixel values are treated as zeroes, and all of the others are treated as 1's

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.61 MorphologyEx

- **MorphologyEx**(CvMat src, CvMat dst, CvArr temp, IplConvKernel element, int operation, int iterations=1)

  Performs advanced morphological transformations.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
temp	Temporary image, required in some cases
element	Structuring element
operation	An integer type number
iterations	Number of times erosion and dilation are applied

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.62 MultiplyAcc

- **MultiplyAcc**(CvArr image1, CvArr image2, CvArr acc, CvArr mask=nil)

  Adds the product of two input images to the accumulator.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image1	First input image, 1- or 3-channel, 8-bit or 32-bit floating point (each channel of multi-channel image is processed independently)
image2	Second input image, the same format as the first one
acc	Accumulator with the same number of channels as input images, 32-bit or 64-bit floating-point
mask	Optional operation mask

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.63 NormalizeHist

- **NormalizeHist**(CvHistogram hist, num factor)

  Normalizes the histogram.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

hist	Histogram
factor	Normalization factor

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.64 PointPolygonTest

- num **PointPolygonTest**(CvArr contour, CvPoint2D32f pt, int measure_dist)

  Point in contour test.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

contour	Input contour
pt	The point tested against the contour
measure_dist	If it is non-zero, the function estimates the distance from the point to the nearest contour edge

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.65 PreCornerDetect

- **PreCornerDetect**(CvArr image, CvArr corners, int aperature_size=3)

  Calculates the feature map for corner detection.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input image
corners	Image to store the corner candidates
aperature_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.66 PyrDown

- **PyrDown**(CvArr src, CvArr dst, int filter=CV_GAUSSIAN_5x5)

  Downsamples an image.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source image
dst	The destination image, should have a half as large width and height than the source
filter	Type of the filter used for convolution; only `CV_GAUSSIAN_5x5` is currently supported

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.67 PyrMeanShiftFiltering

- **PyrMeanShiftFiltering**(CvArr src, CvArr dst, num sp, num sr, int max_level=1, CvTermCriteria termcrit=TermCriteria(CV_TERMCRIT_ITER+CV_TERMCRIT_EPS,5,1))

  Does meanshift image segmentation

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source 8-bit, 3-channel image.
dst	The destination image of the same format and the same size as the source.
sp	The spatial window radius.

sr	The color window radius.
max_level	Maximum level of the pyramid for the segmentation.
termcrit	Termination criteria: when to stop meanshift iterations.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.68 PyrSegmentation

- **PyrSegmentation**(IplImage src, IplImage dst, CvMemStorage storage, CvSeq comp, int level, num threshold1, num threshold2)

  Implements image segmentation by pyramids.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	The source image
dst	The destination image
storage	Storage; stores the resulting sequence of connected components
comp	Pointer to the output sequence of the segmented components
level	Maximum level of the pyramid for the segmentation
threshold1	Error threshold for establishing the links
threshold2	Error threshold for the segments clustering

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.69 PyrUp

- **PyrUp**(CvArr src, CvArr dst, int filter=CV_GAUSSIAN_5x5)

Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
filter	An integer type number with default value *CV_GAUSSIAN_5x5*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.70 ReleaseFeatureTree

- **ReleaseFeatureTree**(CvFeatureTree tr)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tr	A wrappered OpenCV object of type CvFeatureTree

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.71 ReleaseHist

- **ReleaseHist**(CvHistogram hist)

Releases the histogram.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

hist	Double pointer to the released histogram

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.72 ReleaseLSH

- **ReleaseLSH**(CvLSH lsh)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

lsh	A wrappered OpenCV object of type CvLSH

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.73 ReleasePyramid

- **ReleasePyramid**(CvMat pyramid, int extra_layers)

  Non details detected.

Detailed Description
Full function’s description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

pyramid	A wrappered OpenCV object of type CvMat
extra_layers	An integer type number

Returns

none - function doesn’t return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.74 ReleaseStructuringElement

- **ReleaseStructuringElement**(IplConvKernel kernel)

  Deletes a structuring element.

Detailed Description
Full function’s description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

kernel	A wrappered OpenCV object of type IplConvKernel

Returns

none - function doesn’t return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.75 Remap

- **Remap**(CvArr src, CvArr dst, CvArr mapx, CvArr mapy, int flags=CV_INTER_LINEAR+CV_WARP_FILL_OU
  CvScalar fillval=ScalarAll(0))

  Applies a generic geometrical transformation to the image.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	Source image
dst	Destination image
mapx	The map of x-coordinates (CV_32FC1 image)
mapy	The map of y-coordinates (CV_32FC1 image)
flags	An integer type number with default value *CV_INTER_LINEAR*
fillval	A value used to fill outliers

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.76 Resize

- **Resize**(CvArr src, CvArr dst, int interpolation=CV_INTER_LINEAR)

  Resizes an image.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	Source image
dst	Destination image
interpolation	An integer type number with default value *CV_INTER_LINEAR*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.77 RunningAvg

- **RunningAvg**(CvArr image, CvArr acc, num alpha, CvArr mask=nil)

  Updates the running average.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input image, 1- or 3-channel, 8-bit or 32-bit floating point (each channel of multi-channel image is processed independently)
acc	Accumulator with the same number of channels as input image, 32-bit or 64-bit floating-point
alpha	Weight of input image
mask	Optional operation mask

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.78 SampleLine

- int,userdata **SampleLine**(CvMat image, CvPoint pt1, CvPoint pt2, int connectivity=8)

  Reads the raster line to the buffer.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Image to sample the line from
pt1	Starting line point
pt2	Ending line point
connectivity	The line connectivity, 4 or 8

Returns

int - An integer type number defined by architecture
userdata - An unspecified C pointer to memory

The documentation was generated from :

- lua_imgproc.cpp

3.4.79 Smooth

- **Smooth**(CvArr src, CvArr dst, int smoothtype=CV_GAUSIAN, int size1=3, int size2=0, num sigma1=0, num sigma2=0)

  Smooths the image in one of several ways.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source image
dst	The destination image
smoothtype	An integer type number with default value *CV_GAUSIAN*
size1	An integer type number with default value *3*
size2	An integer type number with default value *0*
sigma1	A floating point number with default value *0*
sigma2	A floating point number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.80 Sobel

- **Sobel**(CvArr src, CvArr dst, int xorder, int yorder, int aperature_size=3)

  Calculates the first, second, third or mixed image derivatives using an extended Sobel operator

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	The source image
dst	The destination image; will have the same size and the same number of channels as `src`
xorder	Order of the derivative x
yorder	Order of the derivative y
aperature_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.81 SquareAcc

- **SquareAcc**(CvArr image, CvArr sqsum, CvArr mask=nil)

  Adds the square of the source image to the accumulator.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	Input image, 1- or 3-channel, 8-bit or 32-bit floating point (each channel of multi-channel image is processed independently)
sqsum	Accumulator with the same number of channels as input image, 32-bit or 64-bit floating-point
mask	Optional operation mask

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.82 StartReadChainPoints

- **StartReadChainPoints**(CvChain chain, CvChainPtReader reader)

Initializes the chain reader.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

chain	A wrappered OpenCV object of type CvChain
reader	A wrappered OpenCV object of type CvChainPtReader

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.83 Subdiv2DLocate

- int,CvSubdiv2DEdge,CvSubdiv2DPoint **Subdiv2DLocate**(CvSubdiv2D subdiv, CvPoint2D32f pt)

  Returns the location of a point within a Delaunay triangulation.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

subdiv	Delaunay or another subdivision
pt	The point to locate

Returns

int - An integer type number defined by architecture
CvSubdiv2DEdge - A wrappered OpenCV object of type CvSubdiv2DEdge
CvSubdiv2DPoint - A wrappered OpenCV object of type CvSubdiv2DPoint

The documentation was generated from :

- lua_imgproc.cpp

3.4.84 SubstituteContour

- **SubstituteContour**(CvContourScanner scanner, CvSeq new_contour)

  Replaces a retrieved contour.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

scanner	Contour scanner initialized by **StartFindContours**
new_contour	Substituting contour

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.85 ThreshHist

- **ThreshHist**(CvHistogram hist, num factor)

  Thresholds the histogram.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

hist	Pointer to the histogram
factor	A floating point number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.86 Threshold

- num **Threshold**(CvArr src, CvArr dst, num threshold, num max_value, int threshold_type)

  Applies a fixed-level threshold to array elements.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source array (single-channel, 8-bit or 32-bit floating point)
dst	Destination array; must be either the same type as `src` or 8-bit
threshold	Threshold value
max_value	A floating point number
threshold_type	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.87 TriangleArea

- num **TriangleArea**(CvPoint2D32f a,CvPoint2D32f b,CvPoint2D32f c)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

a	A wrappered OpenCV object of type CvPoint2D32f
b	A wrappered OpenCV object of type CvPoint2D32f
c	A wrappered OpenCV object of type CvPoint2D32f

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_imgproc.cpp

3.4.88 Undistort2

- **Undistort2**(CvArr src, CvArr dst, CvMat camera_matrix, CvMat distortion_coeffs, CvMat new_camera_matrix=nil)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	A wrappered OpenCV object of type CvArr
dst	A wrappered OpenCV object of type CvArr
camera_matrix	A wrappered OpenCV object of type CvMat
distortion_coeffs	A wrappered OpenCV object of type CvMat
new_camera_matrix	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.89 UndistortPoints

- **UndistortPoints**(CvMat src, CvMat dst, CvMat camera_matrix, CvMat distortion_coeffs, CvMat R=nil, CvMat P=nil)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

src	A wrappered OpenCV object of type CvMat
dst	A wrappered OpenCV object of type CvMat
camera_matrix	A wrappered OpenCV object of type CvMat
distortion_coeffs	A wrappered OpenCV object of type CvMat
R	A wrappered OpenCV object of type CvMat with default value *nil*
P	A wrappered OpenCV object of type CvMat with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.90 WarpAffine

- **WarpAffine**(CvArr src, CvArr dst, CvMat map_matrix, int flags=(CV_INTER_LINEAR+CV_WARP_FILL_OUTLIERS), CvScalar fillval=ScalarAll(0))

  Applies an affine transformation to an image.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
map_matrix	A wrappered OpenCV object of type CvMat
flags	An integer type number with default value *(CV_INTER_LINEAR*
fillval	A value used to fill outliers

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.91 WarpPerspective

- **WarpPerspective**(CvArr src, CvArr dst, CvMat map_matrix, int flags=CV_INTER_LINEAR+CV_WARP_FIL CvScalar fillval=ScalarAll(0))

  Applies a perspective transformation to an image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

src	Source image
dst	Destination image
map_matrix	A wrappered OpenCV object of type CvMat
flags	An integer type number with default value *CV_INTER_LINEAR*
fillval	A value used to fill outliers

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.92 Watershed

- **Watershed**(CvArr image, CvArr markers)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

image	A wrappered OpenCV object of type CvArr
markers	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.93 cvLSHQuery

- **cvLSHQuery**(CvLSH lsh, CvMat query_points, CvMat indices, CvMat dist, int k, int emax)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

lsh	A wrappered OpenCV object of type CvLSH
query_points	A wrappered OpenCV object of type CvMat
indices	A wrappered OpenCV object of type CvMat
dist	A wrappered OpenCV object of type CvMat
k	An integer type number
emax	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_imgproc.cpp

3.4.94 cvLSHSize

- int **cvLSHSize**(CvLSH lsh)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

lsh	A wrappered OpenCV object of type CvLSH

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_imgproc.cpp

3.5 Module features2d

3.5.1 SURFParams

- **SURFParams**(CvArr img, CvArr mask, CvSeq keypoints, CvSeq descriptors, CvMemStorage storage, CvSURFParams params, int useProvidedKeyPts=0)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

img	A wrappered OpenCV object of type CvArr
mask	A wrappered OpenCV object of type CvArr
keypoints	A wrappered OpenCV object of type CvSeq
descriptors	A wrappered OpenCV object of type CvSeq
storage	A wrappered OpenCV object of type CvMemStorage
params	A wrappered OpenCV object of type CvSURFParams
useProvidedKeyPts	An integer type number with default value *0*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_features2d.cpp

3.6 Module video

3.6.1 BGCodeBookClearStale

- **BGCodeBookClearStale**(CvBGCodeBookModel model, int staleThresh, CvRect roi=cv.Rect(0,0,0,0), CvArr mask=nil)

Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

model	A wrappered OpenCV object of type CvBGCodeBookModel
staleThresh	An integer type number
roi	A wrappered OpenCV object of type CvRect with default value *cv*
mask	A wrappered OpenCV object of type CvArr with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.2 BGCodeBookDiff

- num **BGCodeBookDiff**(CvBGCodeBookModel model, CvArr image, CvArr fgmask,CvRect roi=cv.Rect(0,0,0,0))

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

model	A wrappered OpenCV object of type CvBGCodeBookModel
image	A wrappered OpenCV object of type CvArr
fgmask	A wrappered OpenCV object of type CvArr
roi	A wrappered OpenCV object of type CvRect with default value *cv*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.3 BGCodeBookUpdate

- **BGCodeBookUpdate**(CvBGCodeBookModel model, CvArr image, CvRect roi=cv.Rect(0,0,0,0), CvArr mask=nil)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

model	A wrappered OpenCV object of type CvBGCodeBookModel
image	A wrappered OpenCV object of type CvArr
roi	A wrappered OpenCV object of type CvRect with default value *cv*
mask	A wrappered OpenCV object of type CvArr with default value *nil*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.4 CalcAffineFlowPyrLK

- string **CalcAffineFlowPyrLK**(CvArr prev, CvArr curr, CvArr prev_pyr, CvArr curr_pyr, CvPoint2D32f[] prev_features, CvPoint2D32f[] curr_features, num[] matrices, int count, CvSize win_size, int level, CvTermCriteria criteria, int flags)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prev	A wrappered OpenCV object of type CvArr
curr	A wrappered OpenCV object of type CvArr
prev_pyr	A wrappered OpenCV object of type CvArr
curr_pyr	A wrappered OpenCV object of type CvArr
prev_features	A wrappered OpenCV object of type CvPoint2D32f which is an array
curr_features	A wrappered OpenCV object of type CvPoint2D32f which is an array
matrices	A floating point number which is an array

count	An integer type number
win_size	A wrappered OpenCV object of type CvSize
level	An integer type number
criteria	A wrappered OpenCV object of type CvTermCriteria
flags	An integer type number

Returns

string - An array of characters

The documentation was generated from :

- lua_video.cpp

3.6.5 CalcGlobalOrientation

- num **CalcGlobalOrientation**(CvArr orientation, CvArr mask, CvArr mhi, num timestamp, num duration)

  Calculates the global motion orientation of some selected region.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

orientation	Motion gradient orientation image; calculated by the function **CalcMotionGradient**
mask	Mask image. It may be a conjunction of a valid gradient mask, obtained with **CalcMotionGradient** and the mask of the region, whose direction needs to be calculated
mhi	Motion history image
timestamp	Current time in milliseconds or other units, it is better to store time passed to **UpdateMotionHistory** before and reuse it here, because running **UpdateMotionHistory** and **CalcMotionGradient** on large images may take some time
duration	Maximal duration of motion track in milliseconds, the same as **UpdateMotionHistory**

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.6 CalcMotionGradient

- **CalcMotionGradient**(CvArr mhi, CvArr mask, CvArr orientation, num delta1, num delta2, int aperature_size=3)

  Calculates the gradient orientation of a motion history image.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

mhi	Motion history image
mask	Mask image; marks pixels where the motion gradient data is correct; output parameter
orientation	Motion gradient orientation image; contains angles from 0 to 360 degrees
delta1	See below
delta2	See below
aperature_size	An integer type number with default value *3*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.7 CalcOpticalFlowBM

- **CalcOpticalFlowBM**(CvArr prev, CvArr cur, CvSize block_size, CvSize shift_size, CvSize max_range, int use_previous, CvArr velx, CvArr vely)

  Calculates the optical flow for two images by using the block matching method.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prev	First image, 8-bit, single-channel
cur	A wrappered OpenCV object of type CvArr
block_size	A wrappered OpenCV object of type CvSize
shift_size	A wrappered OpenCV object of type CvSize
max_range	Size of the scanned neighborhood in pixels around the block
use_previous	An integer type number
velx	A wrappered OpenCV object of type CvArr
vely	Vertical component of the optical flow of the same size `velx`, 32-bit floating-point, single-channel

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.8 CalcOpticalFlowFarneback

- **CalcOpticalFlowFarneback**(CvArr prev, CvArr next, CvArr flow, num pyr_scale, int levels, int winsize, int iterations, int poly_n, num poly_sigma, int flags)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

prev	A wrappered OpenCV object of type CvArr
next	A wrappered OpenCV object of type CvArr
flow	A wrappered OpenCV object of type CvArr
pyr_scale	A floating point number
levels	An integer type number
winsize	An integer type number
iterations	An integer type number
poly_n	An integer type number
poly_sigma	A floating point number
flags	An integer type number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.9 CalcOpticalFlowHS

- **CalcOpticalFlowHS**(CvArr prev, CvArr cur, int use_previous, CvArr velx, CvArr vely, num lambda, CvTermCriteria criteria)

  Calculates the optical flow for two images.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

prev	First image, 8-bit, single-channel
cur	A wrappered OpenCV object of type CvArr
use_previous	An integer type number
velx	Horizontal component of the optical flow of the same size as input images, 32-bit floating-point, single-channel
vely	Vertical component of the optical flow of the same size as input images, 32-bit floating-point, single-channel
lambda	Lagrangian multiplier
criteria	Criteria of termination of velocity computing

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.10 CalcOpticalFlowLK

- **CalcOpticalFlowLK**(CvArr prev, CvArr cur, CvSize win_size, CvArr velx, CvArr vely)

  Calculates the optical flow for two images.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prev	First image, 8-bit, single-channel
cur	A wrappered OpenCV object of type CvArr
win_size	A wrappered OpenCV object of type CvSize
velx	Horizontal component of the optical flow of the same size as input images, 32-bit floating-point, single-channel
vely	Vertical component of the optical flow of the same size as input images, 32-bit floating-point, single-channel

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.11 CalcOpticalFlowPyrLK

- string **CalcOpticalFlowPyrLK**(CvArr prev, CvArr curr, CvArr prev_pyr, CvArr curr_pyr, CvPoint2D32f[] prev_features, CvPoint2D32f[] curr_features, int count, CvSize win_size, int level, CvTermCriteria criteria, int flags)

  Calculates the optical flow for a sparse feature set using the iterative Lucas-Kanade method with pyramids.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prev	First frame, at time `t`
curr	Second frame, at time `t + dt`
prev_pyr	A wrappered OpenCV object of type CvArr
curr_pyr	A wrappered OpenCV object of type CvArr
prev_features	A wrappered OpenCV object of type CvPoint2D32f which is an array
curr_features	A wrappered OpenCV object of type CvPoint2D32f which is an array
count	Number of feature points
win_size	A wrappered OpenCV object of type CvSize
level	Maximal pyramid level number. If `0` , pyramids are not used (single level), if `1` , two levels are used, etc

criteria	Specifies when the iteration process of finding the flow for each point on each pyramid level should be stopped
flags	An integer type number

Returns

string - An array of characters

The documentation was generated from :

- lua_video.cpp

3.6.12 CamShift

- num **CamShift**(CvArr prob_image, CvRect window, CvTermCriteria criteria, CvConnectedComp comp, CvBox2D box=nil)

  Finds the object center, size, and orientation

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prob_image	A wrappered OpenCV object of type CvArr
window	Initial search window
criteria	Stop criteria for the underlying **meanShift**
comp	A wrappered OpenCV object of type CvConnectedComp
box	A wrappered OpenCV object of type CvBox2D with default value *nil*

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.13 ChangeDetection

- num **ChangeDetection**(IplImage prev_frame, IplImage curr_frame, IplImage change_mask)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

prev_frame	A wrappered OpenCV object of type IplImage
curr_frame	A wrappered OpenCV object of type IplImage
change_mask	A wrappered OpenCV object of type IplImage

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.14 EstimateRigidTransform

- int **EstimateRigidTransform**(CvArr A, CvArr B, CvMat M, int full_affine)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

A	A wrappered OpenCV object of type CvArr
B	A wrappered OpenCV object of type CvArr
M	A wrappered OpenCV object of type CvMat
full_affine	An integer type number

Returns

int - An integer type number defined by architecture

The documentation was generated from :

- lua_video.cpp

3.6.15 MeanShift

- num **MeanShift**(CvArr prob_image, CvRect window, CvTermCriteria criteria, CvConnectedComp comp)

  Finds the object center on back projection.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

prob_image	Back projection of the object histogram (see **CalcBackProject**)
window	Initial search window
criteria	Criteria applied to determine when the window search should be finished
comp	Resultant structure that contains the converged search window coordinates (`comp->rect` field) and the sum of all of the pixels inside the window (`comp->area` field)

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.16 RefineForegroundMaskBySegm

- **RefineForegroundMaskBySegm**(CvSeq segments, CvBGStatModel bg_model)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

segments	A wrappered OpenCV object of type CvSeq
bg_model	A wrappered OpenCV object of type CvBGStatModel

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.17 ReleaseBGCodeBookModel

- **ReleaseBGCodeBookModel**(CvBGCodeBookModel model)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

model	A wrappered OpenCV object of type CvBGCodeBookModel

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.18 ReleaseBGStatModel

- **ReleaseBGStatModel**(CvBGStatModel model)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

model	A wrappered OpenCV object of type CvBGStatModel

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.19 ReleaseKalman

- **ReleaseKalman**(CvKalman kalman)

  Deallocates the Kalman filter structure.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

kalman	double pointer to the Kalman filter structure

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.6.20 UpdateBGStatModel

- num **UpdateBGStatModel**(IplImage current_frame, CvBGStatModel bg_model, num learningRate=-1)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

current_frame	A wrappered OpenCV object of type IplImage
bg_model	A wrappered OpenCV object of type CvBGStatModel
learningRate	A floating point number with default value

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_video.cpp

3.6.21 UpdateMotionHistory

- **UpdateMotionHistory**(CvArr silhouette, CvArr mhi, num timestamp, num duration)

  Updates the motion history image by a moving silhouette.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

silhouette	Silhouette mask that has non-zero pixels where the motion occurs
mhi	Motion history image, that is updated by the function (single-channel, 32-bit floating-point)
timestamp	Current time in milliseconds or other units
duration	Maximal duration of the motion track in the same units as `timestamp`

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_video.cpp

3.7 Module legacy

3.7.1 3dTracker2dTrackedObject

- **3dTracker2dTrackedObject**(int id, CvPoint2D32f p)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

id	An integer type number
p	A wrappered OpenCV object of type CvPoint2D32f

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.2 3dTracker3dTrackedObject

- **3dTracker3dTrackedObject**(int id, CvPoint3D32f p)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

id	An integer type number
p	A wrappered OpenCV object of type CvPoint3D32f

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.3 CalcDecompCoeff

- num **CalcDecompCoeff**(IplImage obj, IplImage eigObj, IplImage avg)

  Calculates the decomposition coefficient of an input object.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obj	Input object.
eigObj	Eigen object.
avg	Averaged object.

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.4 CalcPGH

- **CalcPGH**(CvSeq contour, CvHistogram hist)

  Calculates a pair-wise geometrical histogram for a contour.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

contour	Input contour. Currently, only integer point coordinates are allowed
hist	Calculated histogram; must be two-dimensional

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.5 CreateGLCMDescriptors

- **CreateGLCMDescriptors**(CvGLCM destGLCM, int descriptorOptimizationType=CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENEST)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

destGLCM	A wrappered OpenCV object of type CvGLCM
descriptorOptimizationType	An integer type number with default value *CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENEST*

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.6 CreateHandMask

- **CreateHandMask**(CvSeq hand_points, IplImage img_mask, CvRect roi)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

hand_points	A wrappered OpenCV object of type CvSeq
img_mask	A wrappered OpenCV object of type IplImage
roi	A wrappered OpenCV object of type CvRect

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.7 DeInterlace

- **DeInterlace**(CvArr frame, CvArr fieldEven, CvArr field0dd)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

frame	A wrappered OpenCV object of type CvArr
fieldEven	A wrappered OpenCV object of type CvArr
field0dd	A wrappered OpenCV object of type CvArr

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.8 EViterbi

- num **EViterbi**(CvImgObsInfo obs_info, CvEHMM hmm)

  Executes the Viterbi algorithm for the embedded HMM.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info	A wrappered OpenCV object of type CvImgObsInfo
hmm	HMM structure.

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.9 EstimateHMMStateParams

- **EstimateHMMStateParams**(CvImgObsInfo[] obs_info_array, int num_img, CvEHMM hmm)

  Estimates all of the parameters of every HMM state.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info_array	A wrappered OpenCV object of type CvImgObsInfo which is an array
num_img	An integer type number
hmm	HMM.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.10 EstimateObsProb

- **EstimateObsProb**(CvImgObsInfo obs_info, CvEHMM hmm)

  Computes the probability of every observation of several images.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info	A wrappered OpenCV object of type CvImgObsInfo
hmm	HMM structure.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.11 EstimateTransProb

- **EstimateTransProb**(CvImgObsInfo[] obs_info_array, int num_img, CvEHMM hmm)

  Computes transition probability matrices for the embedded HMM.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info_array	A wrappered OpenCV object of type CvImgObsInfo which is an array
num_img	An integer type number
hmm	HMM.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.12 GetGLCMDescriptor

- num **GetGLCMDescriptor**(CvGLCM glcm, int step, int descriptor)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

glcm	A wrappered OpenCV object of type CvGLCM
step	An integer type number
descriptor	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.13 GetGLCMDescriptorStatistics

- num,num **GetGLCMDescriptorStatistics**(CvGLCM glcm, int descriptor)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

glcm	A wrappered OpenCV object of type CvGLCM
descriptor	An integer type number

Returns

num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.14 InitMixSegm

- **InitMixSegm**(CvImgObsInfo[] obs_info_array, int num_img, CvEHMM hmm)

  Segments all observations within every internal state of HMM using state mixture components.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info_array	A wrappered OpenCV object of type CvImgObsInfo which is an array
num_img	An integer type number
hmm	HMM.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.15 MatchContourTrees

- num **MatchContourTrees**(CvContourTree tree1, CvContourTree tree2, int method, num threshold)

  Compares two contours using their tree representations.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

tree1	First contour tree
tree2	Second contour tree
method	Similarity measure, only `CV_CONTOUR_TREES_MATCH_I1` is supported
threshold	Similarity threshold

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.16 MixSegmL2

- **MixSegmL2**(CvImgObsInfo[] obs_info_array, int num_img, CvEHMM hmm)

  Segments the observations from all of the training images using the mixture components of the newly assigned states.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info_array	A wrappered OpenCV object of type CvImgObsInfo which is an array
num_img	An integer type number
hmm	HMM.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.17 Release2DHMM

- **Release2DHMM**(CvEHMM hmm)

  Releases a 2D, embedded HMM.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

hmm	Address of the pointer to the HMM to be released.

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.18 ReleaseFaceTracker

- **ReleaseFaceTracker**(CvFaceTracker ppFaceTracker)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

ppFaceTracker	A wrappered OpenCV object of type CvFaceTracker

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.19 ReleaseGLCM

- **ReleaseGLCM**(CvGLCM glcm)

  Non details detected.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

glcm	A wrappered OpenCV object of type CvGLCM

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.20 ReleaseObsInfo

- **ReleaseObsInfo**(CvImgObsInfo obs_info)

  Releases the observation vector structures.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

obs_info	A wrappered OpenCV object of type CvImgObsInfo

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.7.21 SnakeImage

- num,num,num **SnakeImage**(IplImage image, CvPoint[] points, int lenght, num alpha, num beta, num gamma, int coeff_usage, CvSize win, CvTermCriteria criteria, int calc_gradient=1)

  Changes the contour position to minimize its energy.

Detailed Description

Full function's description can be found at the OpenCV official wiki, see link below. (view full description)

Parameters

image	The source image or external energy field
points	Contour points (snake)
lenght	An integer type number
alpha	A floating point number
beta	Weight[s] of curvature energy, similar to `alpha`
gamma	Weight[s] of image energy, similar to `alpha`
coeff_usage	An integer type number
win	Size of neighborhood of every point used to search the minimum, both `win.width` and `win.height` must be odd
criteria	Termination criteria
calc_gradient	Gradient flag; if not 0, the function calculates the gradient magnitude for every image pixel and consideres it as the energy field, otherwise the input image itself is considered

Returns

num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.22 TrackFace

- int,num **TrackFace**(CvFaceTracker pFaceTracking, IplImage imgGray, CvRect[] pRects, int nRects, CvPoint ptRotate)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

pFaceTracking	A wrappered OpenCV object of type CvFaceTracker
imgGray	A wrappered OpenCV object of type IplImage
pRects	A wrappered OpenCV object of type CvRect which is an array
nRects	An integer type number
ptRotate	A wrappered OpenCV object of type CvPoint

Returns

int - An integer type number defined by architecture
num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_legacy.cpp

3.7.23 UniformImgSegm

- **UniformImgSegm**(CvImgObsInfo obs_info, CvEHMM ehmm)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

obs_info	A wrappered OpenCV object of type CvImgObsInfo
ehmm	A wrappered OpenCV object of type CvEHMM

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_legacy.cpp

3.8 Module objdetect

3.8.1 ReleaseHaarClassifierCascade

- **ReleaseHaarClassifierCascade**(CvHaarClassifierCascade cascade)

  Releases the haar classifier cascade.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

cascade	Double pointer to the released cascade. The pointer is cleared by the function

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_objdetect.cpp

3.8.2 ReleaseLatentSvmDetector

- **ReleaseLatentSvmDetector**(CvLatentSvmDetector detector)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

detector	A wrappered OpenCV object of type CvLatentSvmDetector

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_objdetect.cpp

3.8.3 RunHaarClassifierCascade

- num **RunHaarClassifierCascade**(CvHaarClassifierCascade cascade, CvPoint pt, int start_stage=0)

  Runs a cascade of boosted classifiers at the given image location.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

cascade	Haar classifier cascade
pt	The upper left point of window in which the features will be computed. Size of the window is equal to size of training images.
start_stage	Initial zero-based index of the cascade stage to start from. The function assumes that all the previous stages are passed. This feature is used internally by **HaarDetectObjects** for better processor cache utilization

Returns

num - A floating point number defined by lua.h header, in default it is type double

The documentation was generated from :

- lua_objdetect.cpp

3.8.4 SetImagesHaarClassifierCascade

- **SetImagesHaarClassifierCascade**(CvHaarClassifierCascade cascade, CvArr sum, CvArr sqsum, CvArr tilted_sum, num scale)

  Non details detected.

Detailed Description
Full function's description can be found at the OpenCV official wiki, see link below.
(view full description)

Parameters

cascade	A wrappered OpenCV object of type CvHaarClassifierCascade
sum	A wrappered OpenCV object of type CvArr
sqsum	A wrappered OpenCV object of type CvArr
tilted_sum	A wrappered OpenCV object of type CvArr
scale	A floating point number

Returns

none - function doesn't return anything.

The documentation was generated from :

- lua_objdetect.cpp

Chapter 4

Object reference manual

There are implemented plenty of OpenCV structures in LuaCV, even dynamical structures are partially implemented (CvSeq, CvGraph, ...). Due to Lua callback functions (__newindex,__index) is possible to set and get variables from some of OpenCV C structure attributes from Lua. It's not efficient but it's possible e.g. each OpenCV pointer has his own metatable with callbacks.

4.1 Basic Lua and LuaCV types

These are base Lua and LuaCV types, Lua is in default set to have number type as double, but due to many OpenCV functions, there is need for integer type

Types

- **number** - Base Lua number given by lua.conf setting
- **float** - C++ float number
- **string** - Base Lua string
- **userdata** - Raw C++ pointer without metatable
- **int** - C++ integer number

4.2 Cv3dTracker2dTrackedObject

No description detected

Attributes

- number - id
- CvPoint2D32f - p

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/Cv3dTracker2dTrackedObject.cpp

4.3 Cv3dTrackerTrackedObject

No description detected

Attributes

- number - id
- CvPoint3D32f - p

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/Cv3dTrackerTrackedObject.cpp

4.4 CvArr

Arbitrary array

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvArr.cpp

4.5 CvBGCodeBookElem

No description detected

Attributes

- UserData - next
- number - stale
- number - tLastUpdate

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGCodeBookElem.cpp

4.6 CvBGCodeBookModel

No description detected

Attributes

- UserData - freeList
- CvSize - size
- CvMemStorage - storage
- number - t

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGCodeBookModel.cpp

4.7 CvBGPixelCCStatTable

No description detected

Attributes

- number - Pv
- number - Pvb

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGPixelCCStatTable.cpp

4.8 CvBGPixelCStatTable

No description detected

Attributes

- number - Pv
- number - Pvb

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGPixelCStatTable.cpp

4.9 CvBGPixelStat

No description detected

Attributes

- CvBGPixelCStatTable - ctable
- CvBGPixelCCStatTable - cctable
- number - is_trained_st_model
- number - is_trained_dyn_model
- number - Pbcc
- number - Pbc

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGPixelStat.cpp

4.10 CvBGStatModel

No description detected

Attributes

- UserData - background
- UserData - foreground
- CvSeq - foreground_regions
- number - layer_count
- CvMemStorage - storage
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBGStatModel.cpp

4.11 CvBox2D

No description detected

Attributes

- number - angle
- CvPoint2D32f - center
- CvSize2D32f - size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvBox2D.cpp

4.12 CvCapture

Video capturing structure.

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvCapture.cpp

4.13 CvChain

No description detected

Attributes

- UserData - block_max
- number - delta_elems
- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- CvPoint - origin
- UserData - ptr
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvChain.cpp

4.14 CvChainPtReader

No description detected

Attributes

- CvSeqBlock - block
- UserData - block_min
- UserData - block_max
- number - code
- number - delta_index
- number - header_size
- UserData - ptr
- CvPoint - pt
- UserData - prev_elem
- CvSeq - seq

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvChainPtReader.cpp

4.15 CvConnectedComp

No description detected

Attributes

- number - area
- CvSeq - contour
- CvRect - rect
- CvScalar - value

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvConnectedComp.cpp

4.16 CvContour

No description detected

Attributes

- UserData - block_max
- number - color
- number - delta_elems
- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- UserData - ptr
- CvRect - rect
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvContour.cpp

4.17 CvContourScanner

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvContourScanner.cpp

4.18 CvContourTree

No description detected

Attributes

- UserData - block_max
- number - delta_elems
- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- UserData - ptr
- CvPoint - p1
- CvPoint - p2
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvContourTree.cpp

4.19 CvEHMM

No description detected

Attributes

- number - level
- number - num_states

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvEHMM.cpp

4.20 CvFaceTracker

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvFaceTracker.cpp

4.21 CvFeatureTree

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvFeatureTree.cpp

4.22 CvFGDStatModel

No description detected

Attributes

- UserData - Ftd
- UserData - Fbd
- UserData - background
- UserData - foreground
- CvSeq - foreground_regions

- number - layer_count
- CvBGPixelStat - pixel_stat
- UserData - prev_frame
- CvFGDStatModelParams - params
- CvMemStorage - storage
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvFGDStatModel.cpp

4.23 CvFGDStatModelParams

No description detected

Attributes

- number - alpha1
- number - alpha2
- number - alpha3
- number - delta
- number - is_obj_without_holes
- number - Lc
- number - Lcc
- number - N1c
- number - N2c
- number - N1cc
- number - N2cc
- number - minArea
- number - perform_morphing
- number - T

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvFGDStatModelParams.cpp

4.24 CvFont

No description detected

Attributes

- UserData - ascii
- UserData - cyrillic
- number - dx
- number - font_face
- UserData - greek
- number - hscale
- number - line_type
- number - shear
- number - thickness
- number - vscale

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvFont.cpp

4.25 CvGaussBGModel

No description detected

Attributes

- UserData - background
- number - countFrames
- UserData - foreground
- CvSeq - foreground_regions
- CvGaussBGPoint - g_point
- number - layer_count
- CvGaussBGStatModelParams - params
- CvMemStorage - storage
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvGaussBGModel.cpp

4.26 CvGaussBGPoint

No description detected

Attributes

- CvGaussBGValues - g_values

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvGaussBGPoint.cpp

4.27 CvGaussBGStatModelParams

No description detected

Attributes

- number - bg_threshold
- number - minArea
- number - n_gauss
- number - std_threshold
- number - variance_init
- number - weight_init
- number - win_size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvGaussBGStatModelParams.cpp

4.28 CvGaussBGValues

No description detected

Attributes

- number - match_sum
- number - weight

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvGaussBGValues.cpp

4.29 CvGLCM

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvGLCM.cpp

4.30 CvGraph

Oriented or unoriented weighted graph.

Attributes

- number - active_count
- UserData - block_max
- number - delta_elems
- CvSet - edges
- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- CvSetElem - free_elems

- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- UserData - ptr
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvGraph.cpp

4.31 CvGraphEdge

No description detected

Attributes

- number - flags
- number - weight

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvGraphEdge.cpp

4.32 CvGraphScanner

Graph traversal state.

Attributes

- CvGraphEdge - edge
- CvGraphVtx - dst
- CvGraph - graph
- number - index

- number - mask
- CvSeq - stack
- CvGraphVtx - vtx

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvGraphScanner.cpp

4.33 CvGraphVtx

No description detected

Attributes

- number - flags
- CvGraphEdge - first

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvGraphVtx.cpp

4.34 CvGraphVtx2D

No description detected

Attributes

- number - flags
- CvGraphEdge - first
- CvPoint2D32f - ptr

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvGraphVtx2D.cpp

4.35 CvHaarClassifier

No description detected

Attributes

- number - count
- CvHaarFeature - haar_feature

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvHaarClassifier.cpp

4.36 CvHaarClassifierCascade

No description detected

Attributes

- number - count
- number - flags
- CvSize - orig_window_size
- CvSize - real_window_size
- number - scale
- CvHaarStageClassifier - stage_classifier

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvHaarClassifierCascade.cpp

4.37 CvHaarFeature

No description detected

Attributes

- number - tilted

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvHaarFeature.cpp

4.38 CvHaarStageClassifier

No description detected

Attributes

- number - count
- number - child
- CvHaarClassifier - classifier
- number - next
- number - parent
- number - threshold

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below. (view full description)

The documentation for this object was generated from :

- objects/CvHaarStageClassifier.cpp

4.39 CvHistogram

Multi-dimensional histogram.

Attributes

- UserData - bins
- CvMatND - mat
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvHistogram.cpp

4.40 CvHuMoments

No description detected

Attributes

- number - hu1
- number - hu2
- number - hu3
- number - hu4
- number - hu5
- number - hu6
- number - hu7

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvHuMoments.cpp

4.41 CvImgObsInfo

No description detected

Attributes

- number - obs_x
- number - obs_y
- number - obs_size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvImgObsInfo.cpp

4.42 CvKalman

Kalman filter state.

Attributes

- number - CP
- number - DP
- number - MP
- UserData - control_matrix
- UserData - error_cov_pre
- UserData - error_cov_post
- UserData - gain
- UserData - measurement_noise_cov
- UserData - measurement_matrix
- UserData - process_noise_cov
- UserData - state_pre
- UserData - state_post
- UserData - transition_matrix
- UserData - temp1
- UserData - temp2
- UserData - temp3
- UserData - temp4
- UserData - temp5

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvKalman.cpp

4.43 CvLatentSvmDetector

No description detected

Attributes

- number - num_filters
- number - num_components
- number - score_threshold

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLatentSvmDetector.cpp

4.44 CvLineIterator

No description detected

Attributes

- number - err
- number - minus_delta
- number - minus_step
- number - plus_delta
- number - plus_step
- UserData - ptr

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLineIterator.cpp

4.45 CvLSH

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLSH.cpp

4.46 CvLSHOperations

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLSHOperations.cpp

4.47 CvLSVMFilterObject

No description detected

Attributes

- number - numFeatures
- number - sizeX
- number - sizeY
- CvLSVMFilterPosition - V

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLSVMFilterObject.cpp

4.48 CvLSVMFilterPosition

No description detected

Attributes

- number - l
- number - x

- number - y

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvLSVMFilterPosition.cpp

4.49 CvMat

A multi-channel matrix.

Attributes

- number - cols
- number - height
- number - rows
- number - step
- number - type
- number - width

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMat.cpp

4.50 CvMatND

Multi-dimensional dense multi-channel array.

Attributes

- number - dims
- number - hdr_refcount
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMatND.cpp

4.51 CvMemBlock

Memory storage block.

Attributes

- CvMemBlock - next
- CvMemBlock - prev

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMemBlock.cpp

4.52 CvMemStorage

Growing memory storage.

Attributes

- CvMemBlock - bottom
- number - block_size
- number - free_space
- CvMemStorage - parent
- number - signature
- CvMemBlock - top

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMemStorage.cpp

4.53 CvMemStoragePos

Memory storage position.

Attributes

- number - free_space
- CvMemBlock - top

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMemStoragePos.cpp

4.54 CvMoments

No description detected

Attributes

- number - inv_sqrt_m00
- number - m00
- number - m10
- number - m01
- number - m20
- number - m11
- number - m02
- number - m30
- number - m21
- number - m12
- number - m03
- number - mu20
- number - mu11
- number - mu02
- number - mu30
- number - mu21
- number - mu12
- number - mu03

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMoments.cpp

4.55 CvMSERParams

No description detected

Attributes

- number - delta
- number - maxArea
- number - minArea
- number - maxVariation
- number - minDiversity
- number - maxEvolution
- number - areaThreshold
- number - minMargin
- number - edgeBlurSize

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvMSERParams.cpp

4.56 CvNArrayIterator

No description detected

Attributes

- number - count
- number - dims
- CvSize - size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvNArrayIterator.cpp

4.57 CvObjectDetection

No description detected

Attributes

- CvRect - rect
- number - score

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvObjectDetection.cpp

4.58 CvPoint

2D point with integer coordinates (usually zero-based).

Attributes

- number - x
- number - y

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPoint.cpp

4.59 CvPoint2D32f

2D point with floating-point coordinates

Attributes

- number - x
- number - y

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPoint2D32f.cpp

4.60 CvPoint2D64f

2D point with double precision floating-point coordinates

Attributes

- number - x
- number - y

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPoint2D64f.cpp

4.61 CvPoint3D32f

3D point with floating-point coordinates

Attributes

- number - x
- number - y
- number - z

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPoint3D32f.cpp

4.62 CvPoint3D64f

3D point with double precision floating-point coordinates

Attributes

- number - x
- number - y

- number - z

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPoint3D64f.cpp

4.63 CvPOSITObject

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvPOSITObject.cpp

4.64 CvQuadEdge2D

Quad-edge of planar subdivision.

Attributes

- number - flags

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvQuadEdge2D.cpp

4.65 CvRect

Offset (usually the top-left corner) and size of a rectangle.

Attributes

- number - height
- number - width
- number - x
- number - y

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvRect.cpp

4.66 CvRNG

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvRNG.cpp

4.67 CvScalar

A container for 1-,2-,3- or 4-tuples of doubles.

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvScalar.cpp

4.68 CvSeq

Growable sequence of elements.

Attributes

- UserData - block_max
- number - delta_elems
- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- UserData - ptr
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSeq.cpp

4.69 CvSeqBlock

Continuous sequence block.

Attributes

- number - count
- UserData - data
- CvSeqBlock - next
- CvSeqBlock - prev
- number - start_index

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSeqBlock.cpp

4.70 CvSeqReader

No description detected

Attributes

- CvSeqBlock - block
- UserData - block_min
- UserData - block_max
- number - delta_index
- number - header_size
- UserData - ptr
- UserData - prev_elem
- CvSeq - seq

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSeqReader.cpp

4.71 CvSeqWriter

No description detected

Attributes

- CvSeqBlock - block
- UserData - block_min
- UserData - block_max
- number - header_size
- UserData - ptr
- CvSeq - seq

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSeqWriter.cpp

4.72 CvSet

Collection of nodes.

Attributes

- number - active_count
- UserData - block_max
- number - delta_elems
- number - elem_size
- number - flags
- CvSeqBlock - free_blocks
- CvSetElem - free_elems
- CvSeqBlock - first
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- UserData - ptr
- CvMemStorage - storage
- number - total
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSet.cpp

4.73 CvSetElem

No description detected

Attributes

- number - flags
- CvSetElem - next_free

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSetElem.cpp

4.74 CvSize

Pixel-accurate size of a rectangle.

Attributes

- number - height
- number - width

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSize.cpp

4.75 CvSize2D32f

Sub-pixel accurate size of a rectangle.

Attributes

- number - height
- number - width

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSize2D32f.cpp

4.76 CvSlice

A sequence slice.

Attributes

- number - end_index
- number - start_index

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSlice.cpp

4.77 CvSparseMat

Multi-dimensional sparse multi-channel array.

Attributes

- number - dims
- number - hdr_refcount
- CvSet - heap
- number - hashsize
- number - idxoffset
- number - type
- number - valoffset

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSparseMat.cpp

4.78 CvSparseMatIterator

No description detected

Attributes

- number - curidx
- CvSparseMat - mat
- CvSparseNode - node

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSparseMatIterator.cpp

4.79 CvSparseNode

No description detected

Attributes

- number - hashval
- CvSparseNode - next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSparseNode.cpp

4.80 CvStarDetectorParams

No description detected

Attributes

- number - maxSize
- number - responseThreshold
- number - lineThresholdProjected
- number - lineThresholdBinarized
- number - suppressNonmaxSize

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvStarDetectorParams.cpp

4.81 CvStarKeypoint

No description detected

Attributes

- CvPoint - pt
- number - response
- number - size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvStarKeypoint.cpp

4.82 CvStereoBMState

The structure for block matching stereo correspondence algorithm.

Attributes

- number - SADWindowSize
- UserData - cost
- UserData - disp
- number - disp12MaxDiff
- number - minDisparity
- number - numberOfDisparities
- number - preFilterType
- number - preFilterSize
- number - preFilterCap
- UserData - preFilteredImg0
- UserData - preFilteredImg1
- CvRect - roi1
- CvRect - roi2
- number - speckleWindowSize
- number - speckleRange
- UserData - slidingSumBuf
- number - textureThreshold
- number - trySmallerWindows
- number - uniquenessRatio

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvStereoBMState.cpp

4.83 CvStereoGCState

The structure for graph cuts-based stereo correspondence algorithm

Attributes

- number - Ithreshold
- number - K
- UserData - dispLeft
- UserData - dispRight
- UserData - edgeBuf
- number - interactionRadius
- number - lambda
- number - lambda1
- number - lambda2
- UserData - left
- number - minDisparity
- number - maxIters
- number - numberOfDisparities
- number - occlusionCost
- UserData - ptrLeft
- UserData - ptrRight
- UserData - right
- UserData - vtxBuf

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvStereoGCState.cpp

4.84 CvSubdiv2D

Planar subdivision.

Attributes

- number - active_count
- UserData - block_max
- CvPoint2D32f - bottomright
- number - delta_elems
- CvSeq - edges

- number - elem_size
- CvSeqBlock - first
- number - flags
- CvSeqBlock - free_blocks
- CvSetElem - free_elems
- number - header_size
- CvSeq - h_prev
- CvSeq - h_next
- number - is_geometry_valid
- UserData - ptr
- number - quad_edges
- CvMemStorage - storage
- UserData - recent_edge
- number - total
- CvPoint2D32f - topleft
- CvSeq - v_prev
- CvSeq - v_next

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSubdiv2D.cpp

4.85 CvSubdiv2DEdge

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSubdiv2DEdge.cpp

4.86 CvSubdiv2DPoint

Point of original or dual subdivision.

Attributes

- number - flags
- UserData - first
- number - id
- CvPoint2D32f - pt

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSubdiv2DPoint.cpp

4.87 CvSURFParams

No description detected

Attributes

- number - extended
- number - hessianThreshold
- number - nOctaves
- number - nOctaveLayers
- number - upright

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSURFParams.cpp

4.88 CvSURFPoint

A SURF keypoint, represented as a tuple `((x, y), laplacian, size, dir, hessian).`

Attributes

- number - dir
- number - hessian

- number - laplacian
- CvPoint2D32f - pt
- number - size

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvSURFPoint.cpp

4.89 CvTermCriteria

Termination criteria for iterative algorithms.

Attributes

- number - epsilon
- number - max_iter
- number - type

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvTermCriteria.cpp

4.90 CvTreeNodeIterator

Opens existing or creates new file storage.

Attributes

- number - level
- UserData - node
- number - max_level

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvTreeNodeIterator.cpp

4.91 CvVideoWriter

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/CvVideoWriter.cpp

4.92 IplConvKernel

An IplConvKernel is a rectangular convolution kernel, created by function **CreateStructuringElementEx**.

Attributes

- number - anchorX
- number - anchorY
- number - nCols
- number - nRows
- number - nShiftR

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/IplConvKernel.cpp

4.93 IplConvKernelFP

No description detected

Attributes

- number - anchorX
- number - anchorY
- number - nCols
- number - nRows

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/IplConvKernelFP.cpp

4.94 IplImage

IPL image header

Attributes

- number - ID
- number - align
- number - depth
- number - dataOrder
- number - height
- number - imageSize
- number - nSize
- number - nChannels
- number - origin
- number - widthStep
- number - width

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/IplImage.cpp

4.95 IplROI

No description detected

Attributes

- number - coi
- number - height
- number - width
- number - xOffset
- number - yOffset

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/IplROI.cpp

4.96 IplTileInfo

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/IplTileInfo.cpp

4.97 Mat

No description detected

Attributes

- No attributes detected.

Detailed Description
Full object's description can be found at the OpenCV official wiki, see the link below.
(view full description)

The documentation for this object was generated from :

- objects/Mat.cpp

Chapter 5

Constant reference manual

This chapter includes all of registered macros from OpenCV to LuaCV. Only simple macros e.g constants are supported because, there are plenty of macro functions in OpenCV C API code.

5.1 Constants from core module

List of constants from core module

Constants

CV_AA	16
CV_AUTOSTEP	0 x7fffffff
CV_AUTO_STEP	0 x7fffffff
CV_BACK	0
CV_C	1
CV_CHOLESKY	3
CV_CMP_EQ	0
CV_CMP_GE	2
CV_CMP_GT	1
CV_CMP_LE	4
CV_CMP_LT	3
CV_CMP_NE	5
CV_CN_MAX	512
CV_CN_SHIFT	3
CV_COVAR_COLS	16
CV_COVAR_NORMAL	1
CV_COVAR_ROWS	8
CV_COVAR_SCALE	4
CV_COVAR_SCRAMBLED	0
CV_COVAR_USE_AVG	2
CV_DEPTH_MAX	(1 << CV_CN_SHIFT)
CV_DIFF	16

CV_DIFF_C	(CV_DIFF | CV_C)
CV_DIFF_L1	(CV_DIFF | CV_L1)
CV_DIFF_L2	(CV_DIFF | CV_L2)
CV_DXT_FORWARD	0
CV_DXT_INVERSE	1
CV_DXT_INVERSE_SCALE	CV_DXT_INV_SCALE
CV_DXT_MUL_CONJ	8 */* conjugate the second argument of cvMulSpectrums */*
CV_DXT_ROWS	4 */* transform each row individually */*
CV_DXT_SCALE	2 */* divide result by size of array */*
CV_ErrModeLeaf	0 */* Print error and exit program */*
CV_ErrModeParent	1 */* Print error and continue */*
CV_ErrModeSilent	2 */* Don't print and continue */*
CV_FILLED	−1
CV_FONT_HERSHEY_COMPLEX	3
CV_FONT_HERSHEY_COMPLEX_SMALL	5
CV_FONT_HERSHEY_DUPLEX	2
CV_FONT_HERSHEY_PLAIN	1
CV_FONT_HERSHEY_SCRIPT_COMPLEX	7
CV_FONT_HERSHEY_SCRIPT_SIMPLEX	6
CV_FONT_HERSHEY_SIMPLEX	0
CV_FONT_HERSHEY_TRIPLEX	4
CV_FONT_ITALIC	16
CV_FONT_VECTOR0	CV_FONT_HERSHEY_SIMPLEX
CV_FRONT	1
CV_GEMM_A_T	1
CV_GEMM_B_T	2
CV_GEMM_C_T	4
CV_GRAPH	CV_SEQ_KIND_GRAPH
CV_GRAPH_FLAG_ORIENTED	(1 << CV_SEQ_FLAG_SHIFT)
CV_HIST_ARRAY	0
CV_HIST_MAGIC_VAL	0x42450000
CV_HIST_RANGES_FLAG	(1 << 11)
CV_HIST_SPARSE	1
CV_HIST_TREE	CV_HIST_SPARSE
CV_HIST_UNIFORM	1
CV_HIST_UNIFORM_FLAG	(1 << 10)
CV_KMEANS_USE_INITIAL_LABELS	1
CV_L1	2
CV_L2	4
CV_LOG2	0.69314718055994530941723212145818
CV_LU	0
CV_MAGIC_MASK	0xFFFF0000
CV_MAT_CN_MASK	((CV_CN_MAX − 1)<< CV_CN_SHIFT)
CV_MAT_CONT_FLAG	(1 << CV_MAT_CONT_FLAG_SHIFT)
CV_MAT_CONT_FLAG_SHIFT	14
CV_MAT_DEPTH_MASK	(CV_DEPTH_MAX − 1)

CV_MAT_MAGIC_VAL	0x42420000
CV_MAT_TYPE_MASK	(CV_DEPTH_MAX*CV_CN_MAX − 1)
CV_MAX_ARR	10
CV_MAX_DIM	32
CV_MAX_DIM_HEAP	(1 << 16)
CV_MINMAX	32
CV_NORMAL	16
CV_NORM_MASK	7
CV_NO_CN_CHECK	2
CV_NO_DEPTH_CHECK	1
CV_NO_SIZE_CHECK	4
CV_ORIENTED_GRAPH	(CV_SEQ_KIND_GRAPH|CV_GRAPH_FLAG_ORIENTED)
CV_PCA_DATA_AS_COL	1
CV_PCA_DATA_AS_ROW	0
CV_PCA_USE_AVG	2
CV_PI	3.1415926535897932384626433832795
CV_QR	4
CV_RAND_NORMAL	1
CV_RAND_UNI	0
CV_REDUCE_AVG	1
CV_REDUCE_MAX	2
CV_REDUCE_MIN	3
CV_REDUCE_SUM	0
CV_RELATIVE	8
CV_RELATIVE_C	(CV_RELATIVE | CV_C)
CV_RELATIVE_L1	(CV_RELATIVE | CV_L1)
CV_RELATIVE_L2	(CV_RELATIVE | CV_L2)
CV_SEQ_CHAIN	(CV_SEQ_KIND_CURVE | CV_SEQ_ELTYPE_CODE)
CV_SEQ_CHAIN_CONTOUR	(CV_SEQ_FLAG_CLOSED | CV_SEQ_CHAIN)
CV_SEQ_CONNECTED_COMP	(CV_SEQ_KIND_GENERIC | CV_SEQ_ELTYPE_CONNECTED_COMP)
CV_SEQ_CONTOUR	CV_SEQ_POLYGON
CV_SEQ_ELTYPE_BITS	12
CV_SEQ_ELTYPE_CODE	CV_8UC1 /* *freeman code: 0..7* */
CV_SEQ_ELTYPE_CONNECTED_COMP	0 /* *connected component* */
CV_SEQ_ELTYPE_GENERIC	0
CV_SEQ_ELTYPE_GRAPH_EDGE	0 /* *&next_o, &next_d, &vtx_o, &vtx_d* */
CV_SEQ_ELTYPE_GRAPH_VERTEX	0 /* *first_edge, &(x,y)* */
CV_SEQ_ELTYPE_INDEX	CV_32SC1 /**#(x,y)* */
CV_SEQ_ELTYPE_MASK	((1 << CV_SEQ_ELTYPE_BITS)− 1)
CV_SEQ_ELTYPE_POINT	CV_32SC2 /**(x,y)* */
CV_SEQ_ELTYPE_POINT3D	CV_32FC3 /**(x,y,z)* */
CV_SEQ_ELTYPE_PPOINT	CV_SEQ_ELTYPE_PTR /**&(x,y)**/
CV_SEQ_ELTYPE_PTR	CV_USRTYPE1
CV_SEQ_ELTYPE_TRIAN_ATR	0 /* *vertex of the binary tree* */

CV_SEQ_FLAG_CLOSED	(1 << CV_SEQ_FLAG_SHIFT)
CV_SEQ_FLAG_CONVEX	(0 << CV_SEQ_FLAG_SHIFT)
CV_SEQ_FLAG_HOLE	(2 << CV_SEQ_FLAG_SHIFT)
CV_SEQ_FLAG_SHIFT	(CV_SEQ_KIND_BITS + CV_SEQ_ELTYPE_BITS)
CV_SEQ_FLAG_SIMPLE	(0 << CV_SEQ_FLAG_SHIFT)
CV_SEQ_INDEX	(CV_SEQ_KIND_GENERIC | CV_SEQ_ELTYPE_INDEX)
CV_SEQ_KIND_BIN_TREE	(2 << CV_SEQ_ELTYPE_BITS)
CV_SEQ_KIND_BITS	2
CV_SEQ_KIND_CURVE	(1 << CV_SEQ_ELTYPE_BITS)
CV_SEQ_KIND_GENERIC	(0 << CV_SEQ_ELTYPE_BITS)
CV_SEQ_KIND_GRAPH	(1 << CV_SEQ_ELTYPE_BITS)
CV_SEQ_KIND_MASK	(((1 << CV_SEQ_KIND_BITS)−1)<<CV_SEQ_ELTYPE_BITS)
CV_SEQ_KIND_SUBDIV2D	(2 << CV_SEQ_ELTYPE_BITS)
CV_SEQ_MAGIC_VAL	0x42990000
CV_SEQ_POINT3D_SET	(CV_SEQ_KIND_GENERIC| CV_SEQ_ELTYPE_POINT3D)
CV_SEQ_POINT_SET	(CV_SEQ_KIND_GENERIC| CV_SEQ_ELTYPE_POINT)
CV_SEQ_POLYGON	(CV_SEQ_FLAG_CLOSED | CV_SEQ_POLYLINE)
CV_SEQ_POLYGON_TREE	(CV_SEQ_KIND_BIN_TREE | CV_SEQ_ELTYPE_TRIAN_ATR)
CV_SEQ_POLYLINE	(CV_SEQ_KIND_CURVE | CV_SEQ_ELTYPE_POINT)
CV_SEQ_SIMPLE_POLYGON	(CV_SEQ_FLAG_SIMPLE | CV_SEQ_POLYGON)
CV_SET_ELEM_FREE_FLAG	(1 << (**sizeof**(**int**)*8−1))
CV_SET_ELEM_IDX_MASK	((1 << 26) − 1)
CV_SET_MAGIC_VAL	0x42980000
CV_SORT_ASCENDING	0
CV_SORT_DESCENDING	16
CV_SORT_EVERY_COLUMN	1
CV_SORT_EVERY_ROW	0
CV_SPARSE_MAT_MAGIC_VAL	0x42440000
CV_STORAGE_MAGIC_VAL	0x42890000
CV_SVD	1
CV_SVD_MODIFY_A	1
CV_SVD_SYM	2
CV_SVD_U_T	2
CV_SVD_V_T	4
CV_TERMCRIT_EPS	2
CV_TERMCRIT_ITER	1
CV_TERMCRIT_NUMBER	CV_TERMCRIT_ITER
CV_USRTYPE1	7

CV_WHOLE_SEQ_END_INDEX	0 x3fffffff
IPL_ALIGN_16BYTES	16
IPL_ALIGN_32BYTES	32
IPL_ALIGN_4BYTES	4
IPL_ALIGN_8BYTES	8
IPL_ALIGN_DWORD	IPL_ALIGN_4BYTES
IPL_ALIGN_QWORD	IPL_ALIGN_8BYTES
IPL_BORDER_CONSTANT	0
IPL_BORDER_REFLECT	2
IPL_BORDER_REFLECT_101	4
IPL_BORDER_REPLICATE	1
IPL_BORDER_WRAP	3
IPL_DATA_ORDER_PIXEL	0
IPL_DATA_ORDER_PLANE	1
IPL_DEPTH_SIGN	0x80000000
IPL_IMAGE_DATA	2
IPL_IMAGE_HEADER	1
IPL_IMAGE_MAGIC_VAL	((**int**)**sizeof**(IplImage))
IPL_IMAGE_ROI	4
IPL_ORIGIN_BL	1
IPL_ORIGIN_TL	0

The documentation for this object was generated from :

- lua_core.cpp

5.2 Constants from calib3d module

List of constants from calib3d module

Constants

CV_CALIB_CB_ADAPTIVE_THRESH	1
CV_CALIB_CB_FILTER_QUADS	4
CV_CALIB_CB_NORMALIZE_IMAGE	2
CV_CALIB_FIX_ASPECT_RATIO	2
CV_CALIB_FIX_FOCAL_LENGTH	16
CV_CALIB_FIX_INTRINSIC	256
CV_CALIB_FIX_K1	32
CV_CALIB_FIX_K2	64
CV_CALIB_FIX_K3	128
CV_CALIB_FIX_PRINCIPAL_POINT	4
CV_CALIB_SAME_FOCAL_LENGTH	512
CV_CALIB_USE_INTRINSIC_GUESS	1
CV_CALIB_ZERO_DISPARITY	1024
CV_CALIB_ZERO_TANGENT_DIST	8

CV_FM_7POINT	1
CV_FM_8POINT	2
CV_FM_LMEDS	CV_LMEDS
CV_FM_LMEDS_ONLY	CV_LMEDS
CV_FM_RANSAC	CV_RANSAC
CV_FM_RANSAC_ONLY	CV_RANSAC
CV_LMEDS	4
CV_RANSAC	8
CV_STEREO_BM_BASIC	0
CV_STEREO_BM_FISH_EYE	1
CV_STEREO_BM_NARROW	2
CV_STEREO_BM_NORMALIZED_RESPONSE	0
CV_STEREO_GC_OCCLUDED	SHRT_MAX

The documentation for this object was generated from :

- lua_calib3d.cpp

5.3 Constants from highgui module

List of constants from highgui module

Constants

CV_CAP_ANY	0
CV_CAP_CMU1394	300
CV_CAP_DC1394	300
CV_CAP_DSHOW	700
CV_CAP_FIREWARE	300
CV_CAP_FIREWIRE	300
CV_CAP_IEEE1394	300
CV_CAP_MIL	100
CV_CAP_PROP_BRIGHTNESS	10
CV_CAP_PROP_CONTRAST	11
CV_CAP_PROP_CONVERT_RGB	16
CV_CAP_PROP_EXPOSURE	15
CV_CAP_PROP_FORMAT	8
CV_CAP_PROP_FOURCC	6
CV_CAP_PROP_FPS	5
CV_CAP_PROP_FRAME_COUNT	7
CV_CAP_PROP_FRAME_HEIGHT	4
CV_CAP_PROP_FRAME_WIDTH	3
CV_CAP_PROP_GAIN	14
CV_CAP_PROP_HUE	13
CV_CAP_PROP_MODE	9
CV_CAP_PROP_POS_AVI_RATIO	2

CV_CAP_PROP_POS_FRAMES	1
CV_CAP_PROP_POS_MSEC	0
CV_CAP_PROP_RECTIFICATION	18
CV_CAP_PROP_SATURATION	12
CV_CAP_PROP_WHITE_BALANCE	17
CV_CAP_QT	500
CV_CAP_STEREO	400
CV_CAP_TYZX	400
CV_CAP_UNICAP	600
CV_CAP_V4L	200
CV_CAP_V4L2	200
CV_CAP_VFW	200
CV_CVTIMG_FLIP	1
CV_CVTIMG_SWAP_RB	2
CV_EVENT_FLAG_ALTKEY	32
CV_EVENT_FLAG_CTRLKEY	8
CV_EVENT_FLAG_LBUTTON	1
CV_EVENT_FLAG_MBUTTON	4
CV_EVENT_FLAG_RBUTTON	2
CV_EVENT_FLAG_SHIFTKEY	16
CV_EVENT_LBUTTONDBLCLK	7
CV_EVENT_LBUTTONDOWN	1
CV_EVENT_LBUTTONUP	4
CV_EVENT_MBUTTONDBLCLK	9
CV_EVENT_MBUTTONDOWN	3
CV_EVENT_MBUTTONUP	6
CV_EVENT_MOUSEMOVE	0
CV_EVENT_RBUTTONDBLCLK	8
CV_EVENT_RBUTTONDOWN	2
CV_EVENT_RBUTTONUP	5
CV_FOURCC_DEFAULT	CV_FOURCC(’I’, ’Y’, ’U’, ’V’) /* Use default codec for specified filename (Linux only) */
CV_FOURCC_PROMPT	−1 /* Open Codec Selection Dialog (Windows only)*/
CV_IMWRITE_JPEG_QUALITY	1
CV_IMWRITE_PNG_COMPRESSION	16
CV_IMWRITE_PXM_BINARY	32
CV_LOAD_IMAGE_ANYCOLOR	4
CV_LOAD_IMAGE_ANYDEPTH	2
CV_LOAD_IMAGE_COLOR	1
CV_LOAD_IMAGE_GRAYSCALE	0
CV_LOAD_IMAGE_UNCHANGED	−1
CV_TYZX_COLOR	402
CV_TYZX_LEFT	400
CV_TYZX_RIGHT	401
CV_TYZX_Z	403

CV_WINDOW_AUTOSIZE	
HG_AUTOSIZE	CV_WINDOW_AUTOSIZE

The documentation for this object was generated from :

- lua_highgui.cpp

5.4 Constants from imgproc module

List of constants from imgproc module

Constants

CV_ADAPTIVE_THRESH_GAUSSIAN_C	1
CV_ADAPTIVE_THRESH_MEAN_C	0
CV_BGR2BGR555	22
CV_BGR2BGR565	12
CV_BGR2BGRA	0
CV_BGR2GRAY	6
CV_BGR2HLS	52
CV_BGR2HSV	40
CV_BGR2Lab	44
CV_BGR2Luv	50
CV_BGR2RGB	4
CV_BGR2RGBA	2
CV_BGR2XYZ	32
CV_BGR2YCrCb	36
CV_BGR5552BGR	24
CV_BGR5552BGRA	28
CV_BGR5552GRAY	31
CV_BGR5552RGB	25
CV_BGR5552RGBA	29
CV_BGR5652BGR	14
CV_BGR5652BGRA	18
CV_BGR5652GRAY	21
CV_BGR5652RGB	15
CV_BGR5652RGBA	19
CV_BGRA2BGR	1
CV_BGRA2BGR555	26
CV_BGRA2BGR565	16
CV_BGRA2GRAY	10
CV_BGRA2RGB	CV_RGBA2BGR
CV_BGRA2RGBA	5
CV_BILATERAL	4
CV_BLUR	1
CV_BLUR_NO_SCALE	0

CV_BayerBG2BGR	46
CV_BayerBG2RGB	CV_BayerRG2BGR
CV_BayerGB2BGR	47
CV_BayerGB2RGB	CV_BayerGR2BGR
CV_BayerGR2BGR	49
CV_BayerGR2RGB	CV_BayerGB2BGR
CV_BayerRG2BGR	48
CV_BayerRG2RGB	CV_BayerBG2BGR
CV_CANNY_L2_GRADIENT	(1 << 31)
CV_CHAIN_APPROX_NONE	1
CV_CHAIN_APPROX_SIMPLE	2
CV_CLOCKWISE	1
CV_COLORCVT_MAX	100
CV_COMP_BHATTACHARYYA	3
CV_COMP_CHISQR	1
CV_COMP_CORREL	0
CV_COMP_INTERSECT	2
CV_CONTOURS_MATCH_I1	1
CV_CONTOURS_MATCH_I2	2
CV_CONTOURS_MATCH_I3	3
CV_COUNTER_CLOCKWISE	2
CV_DIST_L1	1
CV_DIST_L2	2
CV_DIST_MASK_3	3
CV_DIST_MASK_5	5
CV_DIST_MASK_PRECISE	0
CV_FLOODFILL_FIXED_RANGE	(1 << 16)
CV_FLOODFILL_MASK_ONLY	(1 << 17)
CV_GAUSSIAN	2
CV_GRAY2BGR	8
CV_GRAY2BGR555	30
CV_GRAY2BGR565	20
CV_GRAY2BGRA	9
CV_GRAY2RGB	CV_GRAY2BGR
CV_GRAY2RGBA	CV_GRAY2BGRA
CV_HLS2BGR	60
CV_HLS2RGB	61
CV_HOUGH_GRADIENT	3
CV_HOUGH_MULTI_SCALE	2
CV_HOUGH_PROBABILISTIC	1
CV_HOUGH_STANDARD	0
CV_HSV2BGR	54
CV_HSV2RGB	55
CV_INPAINT_NS	0
CV_INPAINT_TELEA	1
CV_INTER_AREA	3
CV_INTER_CUBIC	2

CV_INTER_LINEAR	1
CV_INTER_NN	0
CV_Lab2BGR	56
CV_Lab2RGB	57
CV_Luv2BGR	58
CV_Luv2RGB	59
CV_MAX_SOBEL_KSIZE	7
CV_MEDIAN	3
CV_MOP_BLACKHAT	6
CV_MOP_CLOSE	3
CV_MOP_GRADIENT	4
CV_MOP_OPEN	2
CV_MOP_TOPHAT	5
CV_NEXT_AROUND_LEFT	0x13
CV_POLY_APPROX_DP	0
CV_RETR_CCOMP	2
CV_RETR_LIST	1
CV_RETR_TREE	3
CV_RGB2BGR	CV_BGR2RGB
CV_RGB2BGR555	23
CV_RGB2BGR565	13
CV_RGB2BGRA	CV_BGR2RGBA
CV_RGB2GRAY	7
CV_RGB2HLS	53
CV_RGB2HSV	41
CV_RGB2Lab	45
CV_RGB2Luv	51
CV_RGB2RGBA	CV_BGR2BGRA
CV_RGB2XYZ	33
CV_RGB2YCrCb	37
CV_RGBA2BGR	3
CV_RGBA2BGR555	27
CV_RGBA2BGR565	17
CV_RGBA2BGRA	CV_BGRA2RGBA
CV_RGBA2GRAY	11
CV_RGBA2RGB	CV_BGRA2BGR
CV_SCHARR	−1
CV_SHAPE_CROSS	1
CV_SHAPE_CUSTOM	100
CV_SHAPE_ELLIPSE	2
CV_SHAPE_RECT	0
CV_THRESH_BINARY	0
CV_THRESH_BINARY_INV	1
CV_THRESH_MASK	7
CV_THRESH_OTSU	8 /* use Otsu algorithm to choose the optimal threshold value;
CV_THRESH_TOZERO	3

CV_THRESH_TOZERO_INV	4
CV_THRESH_TRUNC	2
CV_TM_CCOEFF	4
CV_TM_CCOEFF_NORMED	5
CV_TM_CCORR	2
CV_TM_CCORR_NORMED	3
CV_TM_SQDIFF	0
CV_TM_SQDIFF_NORMED	1
CV_WARP_FILL_OUTLIERS	8
CV_WARP_INVERSE_MAP	16
CV_XYZ2BGR	34
CV_XYZ2RGB	35
CV_YCrCb2BGR	38
CV_YCrCb2RGB	39

The documentation for this object was generated from :

- lua_imgproc.cpp

5.5 Constants from features2d module

List of constants from features2d module

Constants

The documentation for this object was generated from :

- lua_features2d.cpp

5.6 Constants from video module

List of constants from video module

Constants

CV_BGFG_FGD_ALPHA_1	0.1f
CV_BGFG_FGD_ALPHA_2	0.005f
CV_BGFG_FGD_ALPHA_3	0.1f
CV_BGFG_FGD_BG_UPDATE_TRESH	0.5f
CV_BGFG_FGD_DELTA	2
CV_BGFG_FGD_LC	128
CV_BGFG_FGD_LCC	64
CV_BGFG_FGD_MINAREA	15.f
CV_BGFG_FGD_N1C	15

CV_BGFG_FGD_N1CC	25
CV_BGFG_FGD_N2C	25
CV_BGFG_FGD_N2CC	40
CV_BGFG_FGD_T	0.9f
CV_BGFG_MOG_BACKGROUND_THRESHOLD	0.7 /* *threshold sum of weights for background test* */
CV_BGFG_MOG_MAX_NGAUSSIANS	500
CV_BGFG_MOG_MINAREA	15.f
CV_BGFG_MOG_NCOLORS	3
CV_BGFG_MOG_NGAUSSIANS	5 /* *= K = number of Gaussians in mixture* */
CV_BGFG_MOG_SIGMA_INIT	30
CV_BGFG_MOG_STD_THRESHOLD	2.5 /* *lambda=2.5 is 99%* */
CV_BGFG_MOG_WEIGHT_INIT	0.05
CV_BGFG_MOG_WINDOW_SIZE	200 /* *Learning rate; alpha = 1/CV_GBG_WINDOW_SIZE* */
CV_LKFLOW_GET_MIN_EIGENVALS	8
CV_LKFLOW_INITIAL_GUESSES	4
CV_LKFLOW_PYR_A_READY	1
CV_LKFLOW_PYR_B_READY	2

The documentation for this object was generated from :

- lua_video.cpp

5.7 Constants from legacy module

List of constants from legacy module

Constants

CV_ARRAY	2
CV_CAMERA_TO_WARP	1
CV_CONTOUR_TREES_MATCH_I1	1 };
CV_DISPARITY_BIRCHFIELD	0
CV_DOMINANT_IPAN	1
CV_GLCMDESC_CLUSTERSHADE	5
CV_GLCMDESC_CLUSTERTENDENCY	4
CV_GLCMDESC_CONTRAST	3
CV_GLCMDESC_CORRELATION	6
CV_GLCMDESC_CORRELATIONINFO1	7
CV_GLCMDESC_CORRELATIONINFO2	8
CV_GLCMDESC_ENERGY	1
CV_GLCMDESC_ENTROPY	0
CV_GLCMDESC_HOMOGENITY	2
CV_GLCMDESC_MAXIMUMPROBABILITY	9

CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENEST	10
CV_GLCMDESC_OPTIMIZATION_ALLOWTRIPLENEST	11
CV_GLCMDESC_OPTIMIZATION_HISTOGRAM	4
CV_GLCM_ALL	0
CV_GLCM_DESC	2
CV_GLCM_GLCM	1
CV_GLCM_OPTIMIZATION_HISTOGRAM	0
CV_GLCM_OPTIMIZATION_LUT	−1
CV_GLCM_OPTIMIZATION_NONE	−2
CV_IDP_BIRCHFIELD_PARAM1	25
CV_IDP_BIRCHFIELD_PARAM2	5
CV_IDP_BIRCHFIELD_PARAM3	12
CV_IDP_BIRCHFIELD_PARAM4	15
CV_IDP_BIRCHFIELD_PARAM5	25
CV_NUM_FACE_ELEMENTS	3
CV_UNDEF_SC_PARAM	12345 *//default value of parameters*
CV_VALUE	1
CV_WARP_TO_CAMERA	2

The documentation for this object was generated from :

- lua_legacy.cpp

5.8 Constants from objdetect module

List of constants from objdetect module

Constants

CV_HAAR_DO_CANNY_PRUNING	1
CV_HAAR_DO_ROUGH_SEARCH	8
CV_HAAR_FIND_BIGGEST_OBJECT	4
CV_HAAR_SCALE_IMAGE	2

The documentation for this object was generated from :

- lua_objdetect.cpp