

Reference Manual

Generated by Doxygen 1.4.7

Sun Oct 22 15:33:47 2006

Contents

1	MSER Detector documentation	1
1.1	Introduction	1
1.2	Configuration	1
1.3	Installation	2
1.4	Examples	2
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	extrema Namespace Reference	11
6.2	std Namespace Reference	20
6.3	utls Namespace Reference	21
7	Class Documentation	23
7.1	utls::Ary< PixelType > Struct Template Reference	23
7.2	extrema::BoundaryExtrema Struct Reference	27
7.3	extrema::BoundaryPoint Struct Reference	28
7.4	extrema::BoundaryRegion Struct Reference	29
7.5	extrema::CPreprocess Class Reference	30
7.6	extrema::CPreprocessRGBPlanes Class Reference	32

7.7	extrema::CPreprocessRGBSeq Class Reference	33
7.8	extrema::ExtremaImage Struct Reference	34
7.9	extrema::ExtremaParams Struct Reference	35
7.10	extrema::ExtremaStats Struct Reference	36
7.11	utls::Matrix2 Class Reference	38
7.12	extrema::Region Struct Reference	41
7.13	extrema::RLEExtrema Struct Reference	44
7.14	extrema::RLEItem Struct Reference	45
7.15	extrema::RLERegion Struct Reference	46
7.16	extrema::s_borderpixel Struct Reference	47
7.17	extrema::s_region Struct Reference	48
7.18	extrema::s_region_equiv Struct Reference	49
7.19	extrema::s_sortpixels Struct Reference	50
7.20	extrema::s_suballocator Struct Reference	51
7.21	extrema::s_thresh_def Struct Reference	52
7.22	extrema::s_thresh_par Struct Reference	53
7.23	utls::t_frgh Union Reference	55
7.24	extrema::t_ipoint Struct Reference	56
7.25	utls::t_rgb Union Reference	57
8	File Documentation	59
8.1	ary.h File Reference	59
8.2	boundary.h File Reference	60
8.3	common.oxy File Reference	61
8.4	dtypes.h File Reference	62
8.5	extremaConfig.h File Reference	63
8.6	extremaParams.h File Reference	64
8.7	extremaTypes.h File Reference	65
8.8	getExtrema.h File Reference	67
8.9	libExtrema.h File Reference	68
8.10	matrix.h File Reference	70
8.11	optThresh.h File Reference	71
8.12	preprocess.h File Reference	72
8.13	sortPixels.h File Reference	73
8.14	suballoc.h File Reference	74
8.15	timeutls.h File Reference	75

Chapter 1

MSER Detector documentation

1.1 Introduction

This is a documentation for the Maximally Stable Extremal Regions detector. The source files have following structure:

LL	- directory with LL library
doc	- documentation
extrema	- MSER detector sources
imageutls	- wrappers for reading different image file formats
optionGM	- command line option processing library
utls	- various utilities used by MSER detector

Directories **imageutls** and **optionGM** are included only to allow build of a standalone binary. MSER detector itself depends only on the **LL** and **utls** library. Main part of the MSER detector is implemented in **extrema** directory. Whole detector is compiled into library **libExtrema**. The main interface file is [libExtrema.h](#).

1.2 Configuration

Configuration is done through the file **config** in the root directory of the package. The interesting variables follows:

- Compiler optimisation options can be set with the variable `OPTIMIZATION_FLAGS`. Standard setting is:

```
OPTIMIZATION_FLAGS = -O3 -fomit-frame-pointer
```

- Example binary **extrema-bin** located in the directories **extrema** can open images of different file formats. Images are open by the standard image libraries *libpng*, *libjpeg* and *libtiff*. While these are usually present in modern distributions, one can set the usage with the variables `IMAGE_LIBS_CFLAGS` and `IMAGE_LIBS_LFLAGS`:

```
IMAGE_LIBS_CFLAGS = -DWITH_LIBPNG -DWITH_LIBJPEG -DWITH_LIBTIFF  
IMAGE_LIBS_LFLAGS = -lpng -ltiff -ljpeg -lz
```

1.3 Installation

Sources can be built by

```
make all
```

in the root directory.

After succesful build, library files **LL.a** and **libExtrema.a** are located in directories **LL** and **extrema**. Include file with MSER detector's interface i.e. exported functions is located in file **libExtrema.h**.

1.4 Examples

Examples using MSER detector interface will be available soon. However there is an example of a standalone detector that uses older interface. The example is located in **extrema/extrema.cpp** (**extrema-mt/extrema.cpp**).

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

extrema	11
std	20
utls	21

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

utls::Ary< PixelType >	23
extrema::BoundaryExtrema	27
extrema::BoundaryPoint	28
extrema::CPreprocess	30
extrema::CPreprocessRGBPlanes	32
extrema::CPreprocessRGBSeq	33
extrema::ExtremaImage	34
extrema::ExtremaParams	35
extrema::ExtremaStats	36
utls::Matrix2	38
extrema::Region	41
extrema::BoundaryRegion	29
extrema::RLERegion	46
extrema::RLEExtrema	44
extrema::RLEItem	45
extrema::s_borderpixel	47
extrema::s_region	48
extrema::s_region_equiv	49
extrema::s_sortpixels	50
extrema::s_suballocator	51
extrema::s_thresh_def	52
extrema::s_thresh_par	53
utls::t_frgb	55
extrema::t_ipoint	56
utls::t_rgb	57

Chapter 4

Class Index

4.1 Class List

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

utls::Ary< PixelType >	23
extrema::BoundaryExtrema (Old interface structure that holds result of <code>getBoundaryExtrema</code>)	27
extrema::BoundaryPoint (A structure that holds coordinates of a point in BoundaryRegion i.e. the boundary representation of a region)	28
extrema::BoundaryRegion (Description of a boundary region)	29
extrema::CPreprocess	30
extrema::CPreprocessRGBPlanes	32
extrema::CPreprocessRGBSeq	33
extrema::ExtremaImage (A structure holding image parameters)	34
extrema::ExtremaParams (A structure with MSER detector parameters)	35
extrema::ExtremaStats (Timing statistics of the detector, gathered only if <code>TIME_STATS</code> is set)	36
utls::Matrix2	38
extrema::Region (A structure with common statistics of BoundaryRegion and RLERegion)	41
extrema::RLEExtrema (Old interface structure that holds result of <code>getRLEExtrema</code>)	44
extrema::RLEItem (A structure that holds coordinates of a RLE element in RLERegion i.e. the RLE representation of a region)	45
extrema::RLERegion (Description of a RLE region)	46
extrema::s_borderpixel (Structure with pixel of the extended boundary)	47
extrema::s_region (Internal region structure)	48
extrema::s_region_equiv (Internal structure with a node of the label equivalency tree)	49
extrema::s_sortpixels (Internal structure with intensity histogram)	50
extrema::s_suballocator (Internal structure with suballocator's data)	51
extrema::s_thresh_def (Internal structure holding threshold parameters)	52
extrema::s_thresh_par (Internal structure with processed detector's parameters)	53
utls::t_frgb	55
extrema::t_ipoint (Internal structure, holds 2D point coordinates)	56
utls::t_rgb	57

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

ary.h	59
boundary.h	60
common.oxy	61
dtypes.h	62
extremaConfig.h	63
extremaParams.h	64
extremaTypes.h	65
getExtrema.h	67
libExtrema.h	68
matrix.h	70
optThresh.h	71
preprocess.h	72
sortPixels.h	73
suballoc.h	74
timeutls.h	75

Chapter 6

Namespace Documentation

6.1 extrema Namespace Reference

Classes

- struct [ExtremaImage](#)
A structure holding image parameters.
- struct [ExtremaParams](#)
A structure with MSER detector parameters.
- struct [t_ipoint](#)
Internal structure, holds 2D point coordinates.
- struct [s_sortpixels](#)
Internal structure with intensity histogram.
- struct [s_region](#)
Internal region structure.
- struct [s_region_equiv](#)
Internal structure with a node of the label equivalency tree.
- struct [s_thresh_par](#)
Internal structure with processed detector's parameters.
- struct [s_borderpixel](#)
Structure with pixel of the extended boundary.
- struct [s_thresh_def](#)
Internal structure holding threshold parameters.
- struct [BoundaryPoint](#)
A structure that holds coordinates of a point in [BoundaryRegion](#) i.e. the boundary representation of a region.

- struct [RLEItem](#)
A structure that holds coordinates of a RLE element in [RLERegion](#) i.e. the RLE representation of a region.
- struct [Region](#)
A structure with common statistics of [BoundaryRegion](#) and [RLERegion](#).
- struct [RLERegion](#)
Description of a RLE region.
- struct [BoundaryRegion](#)
Description of a boundary region.
- struct [ExtremaStats](#)
Timing statistics of the detector, gathered only if `TIME_STATS` is set.
- struct [RLEExtrema](#)
Old interface structure that holds result of `getRLEExtrema`.
- struct [BoundaryExtrema](#)
Old interface structure that holds result of `getBoundaryExtrema`.
- class [CPreprocess](#)
- class [CPreprocessRGBSeq](#)
- class [CPreprocessRGBPlanes](#)
- struct [s_suballocator](#)
Internal structure with suballocator's data.

Typedefs

- typedef unsigned int [t_label](#)
- typedef [extrema::s_sortpixels](#) [t_sortpixels](#)
Internal structure with intensity histogram.
- typedef unsigned int [t_mregion](#)
- typedef [extrema::s_region](#) [t_region](#)
Internal region structure.
- typedef [extrema::s_region_equiv](#) [t_region_equiv](#)
Internal structure with a node of the label equivalency tree.
- typedef [extrema::s_thresh_par](#) [t_thresh_par](#)
Internal structure with processed detector's parameters.
- typedef [extrema::s_borderpixel](#) [t_borderpixel](#)
Structure with pixel of the extended boundary.
- typedef std::vector< [t_borderpixel](#) > [point_vector](#)
Vector with extended boundary.

- typedef [extrema::s_thresh_def t_thresh_def](#)
Internal structure holding threshold paramaters.
- typedef [extrema::s_suballocator t_suballocator](#)
Internal structure with suballocator's data.

Enumerations

- enum [EXTREMA_PREPROCESS](#) {
[PREPROCESS_CHANNEL_none](#) = 0x00000000,
[PREPROCESS_CHANNEL_intensity](#) = 0x00000001,
[PREPROCESS_CHANNEL_saturation](#) = 0x00000002,
[PREPROCESS_CHANNEL_hue](#) = 0x00000003,
[PREPROCESS_CHANNEL_redblue](#) = 0x00000004,
[PREPROCESS_CHANNEL_red](#) = 0x00000005,
[PREPROCESS_CHANNEL_green](#) = 0x00000006,
[PREPROCESS_CHANNEL_blue](#) = 0x00000007,
[PREPROCESS_CHANNEL_greenmagenta](#) = 0x00000008,
[PREPROCESS_CHANNEL_intensity_half](#) = 0x00000009,
[PREPROCESS_CHANNEL_MASK](#) = 0x0000ffff,
[PREPROCESS_INTENSITY_none](#) = 0x00000000,
[PREPROCESS_INTENSITY_MASK](#) = 0xffff0000 }
An enumeration that encodes different preprocessings of an image.

Functions

- void [RegionBoundaries](#) ([utls::Bary](#) *img, [t_LL](#) regions)
- void [InitRegionRecycling](#) ()
- void [DestRegionRecycling](#) ()
- [t_LL](#) [GetExtrema](#) ([utls::Bary](#) *img, [t_sortpixels](#) pixels, const [ExtremaParams](#) &par, bool invert)
- void [DestRegions](#) ([t_LL](#) regions)
- [BoundaryExtrema](#) [getBoundaryExtrema](#) (const [ExtremaParams](#) ¶ms, const [ExtremaImage](#) &image, int both=3)
Old interface function, produces BoundaryRegions for a given image. Computes MSERs: both=1 +, both = 2 -, or both = 3 + and -.
- [RLEExtrema](#) [getRLEExtrema](#) (const [ExtremaParams](#) &par, const [ExtremaImage](#) &image, int both=3)
Old interface function, produces RLERegions for a given image. Computes MSERs: both=1 +, both = 2 -, or both = 3 + and -.
- void [extremaPrepareImage](#) (const [ExtremaParams](#) ¶ms, const [ExtremaImage](#) &image)
Prepares image for detection of MSER regions.

- void [extremaAttachImage](#) (const [ExtremaParams](#) ¶ms, [utls::BAry](#) *image)
Assigns already prepared image for detection of MSER regions.
- void [extremaInvertImage](#) ()
Inverts image in internal structure.
- void [extremaBoundaryRegions](#) (const [ExtremaParams](#) ¶ms, bool inverted, vector< [BoundaryRegion](#) > &result)
Detects MSERs inverted or not inverted image.
- void [extremaBoundaryEllRegions](#) (const [ExtremaParams](#) ¶ms, bool inverted, vector< [BoundaryRegion](#) > &result)
Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.
- void [extremaRLERegions](#) (const [ExtremaParams](#) ¶ms, bool inverted, vector< [RLERegion](#) > &result)
Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.
- void [extremaCleanup](#) (bool detach_only=false)
Cleans up internal image structure.
- const [ExtremaStats](#) & [extremaStats](#) ()
Returns timing statistics.
- void [exportRLEVector](#) (FILE *fid, vector< [RLERegion](#) > &rle_vector)
- void [exportBoundaryVector](#) (FILE *fid, vector< [BoundaryRegion](#) > &boundary_vector)
- void [exportBoundaryVectorGF](#) (FILE *fid, vector< [BoundaryRegion](#) > &boundary_vector)
- void [exportAffVector](#) (FILE *fid, vector< [RLERegion](#) > &rle_vector, double factor, int krysl_compat)
- void [RLE2Ellipse](#) (const vector< [RLEItem](#) > &rle, double &barX, double &barY, double &sumX2, double &sumXY, double &sumY2)
- void [ReducedBoundary2RLE](#) (vector< [BoundaryPoint](#) > &reduced_boundary, vector< [RLEItem](#) > &rle)
- void [FastSetOptThresholds4StableRegion](#) (t_region *p_r)
- void [CalcHistogram](#) ([utls::BAry](#) *&img, [t_sortpixels](#) &pixels)
- void [BinSortPixels](#) ([utls::BAry](#) *&img, [t_sortpixels](#) &pixels)
- void [InvertImageAndHistogram](#) ([utls::BAry](#) *img, [t_sortpixels](#) &pixels)
- void [InitSuballocator](#) (t_suballocator *s, size_t blocksize, size_t itemsize, int clear_blocks=0)
- void [DestSuballocator](#) (t_suballocator *s)
- void [SuballocatorAddBlock](#) (t_suballocator *s)

Variables

- [t_thresh_par](#) [g_thresh_params](#)

6.1.1 Typedef Documentation

6.1.1.1 typedef unsigned int [extrema::t_label](#)

6.1.1.2 typedef struct [extrema::s_sortpixels](#) [extrema::t_sortpixels](#)

Internal structure with intensity histogram.

6.1.1.3 typedef unsigned int [extrema::t_mregion](#)

6.1.1.4 typedef struct [extrema::s_region](#) [extrema::t_region](#)

Internal region structure.

6.1.1.5 typedef struct [extrema::s_region_equiv](#) [extrema::t_region_equiv](#)

Internal structure with a node of the label equivalency tree.

6.1.1.6 typedef struct [extrema::s_thresh_par](#) [extrema::t_thresh_par](#)

Internal structure with processed detector's parameters.

6.1.1.7 typedef struct [extrema::s_borderpixel](#) [extrema::t_borderpixel](#)

Structure with pixel of the extended boundary.

6.1.1.8 typedef std::vector<[t_borderpixel](#)> [extrema::point_vector](#)

Vector with extended boundary.

6.1.1.9 typedef struct [extrema::s_thresh_def](#) [extrema::t_thresh_def](#)

Internal structure holding threshold paramaters.

6.1.1.10 typedef struct [extrema::s_suballocator](#) [extrema::t_suballocator](#)

Internal structure with suballocator's data.

6.1.2 Enumeration Type Documentation

6.1.2.1 enum [extrema::EXTREMA_PREPROCESS](#)

An enumeration that encodes different preprocessings of an image.

Enumerator:

PREPROCESS_CHANNEL_none

PREPROCESS_CHANNEL_intensity

PREPROCESS_CHANNEL_saturation
PREPROCESS_CHANNEL_hue
PREPROCESS_CHANNEL_redblue
PREPROCESS_CHANNEL_red
PREPROCESS_CHANNEL_green
PREPROCESS_CHANNEL_blue
PREPROCESS_CHANNEL_greenmagenta
PREPROCESS_CHANNEL_intensity_half
PREPROCESS_CHANNEL_MASK
PREPROCESS_INTENSITY_none
PREPROCESS_INTENSITY_MASK

6.1.3 Function Documentation

6.1.3.1 void extrema::RegionBoundaries ([utils::BAry](#) * *img*, *t_LL regions*)

6.1.3.2 void extrema::InitRegionRecycling ()

6.1.3.3 void extrema::DestRegionRecycling ()

6.1.3.4 *t_LL* extrema::GetExtrema ([utils::BAry](#) * *img*, [t_sortpixels](#) *pixels*, const ExtremaParams & *par*, bool *invert*)

6.1.3.5 void extrema::DestRegions (*t_LL regions*)

6.1.3.6 [BoundaryExtrema](#) extrema::getBoundaryExtrema (const ExtremaParams & *params*, const ExtremaImage & *image*, int *both* = 3)

Old interface function, produces BoundaryRegions for a given image. Computes MSERs: both=1 +, both = 2 -, or both = 3 + and -.

6.1.3.7 [RLEExtrema](#) extrema::getRLEExtrema (const ExtremaParams & *par*, const ExtremaImage & *image*, int *both* = 3)

Old interface function, produces RLERegions for a given image. Computes MSERs: both=1 +, both = 2 -, or both = 3 + and -.

6.1.3.8 void extrema::extremaPrepareImage (const ExtremaParams & *params*, const ExtremaImage & *image*)

Prepares image for detection of MSER regions.

Parameters:

params a structure ExtremaPars with detector parameters.

image a structure [ExtremaImage](#) with image data.

Preparation involves preprocessing i.e. performs demanded preprocess operation given by [ExtremaParams.preprocess](#). Image is copied into internal structure and it's boundary is extended by one pixel.

See also:

[EXTREMA_PREPROCESS](#), [ExtremaParams](#).

6.1.3.9 void extrema::extremaAttachImage (const ExtremaParams & *params*, [utls::BAry](#) * *image*)

Assigns already prepared image for detection of MSER regions.

Parameters:

params a structure [ExtremaParams](#) with detector parameters.

image a structure [utls::BAry](#) (byte image) with image, please note image boundaries should be extended by one pixel, i.e. for an image of size width x height one should use constructor BAry(-1,-1,width,height), that results in array of width+2 x height+2 elements.

See also:

[utls::BAry](#), [ExtremaParams](#).

6.1.3.10 void extrema::extremaInvertImage ()

Inverts image in internal structure.

6.1.3.11 void extrema::extremaBoundaryRegions (const ExtremaParams & *params*, bool *inverted*, vector< [BoundaryRegion](#) > & *result*)

Detects MSERs inverted or not inverted image.

Parameters:

params a structure [ExtremaParams](#) with detector parameters.

inverted a boolean that signalise if internal image structure was or was not inverted.

result a vector of [BoundaryRegion](#) structures containing regions.

Note:

This function does not compute and thus fill values of centroid and second moments in [Region](#) structure.

See also:

[BoundaryRegion](#), [ExtremaParams](#)

6.1.3.12 void extrema::extremaBoundaryEllRegions (const ExtremaParams & *params*, bool *inverted*, vector< [BoundaryRegion](#) > & *result*)

Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.

Parameters:

params a structure [ExtremaParams](#) with detector parameters.

inverted a boolean that signalise if internal image structure was or was not inverted.

result a vector of [BoundaryRegion](#) structures containing regions.

See also:

[BoundaryRegion](#), [ExtremaParams](#)

6.1.3.13 void extrema::extremaRLERegions (const ExtremaParams & *params*, bool *inverted*, vector< RLERegion > & *result*)

Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.

Parameters:

params a structure [ExtremaParams](#) with detector parameters.

inverted a boolean that signalise if internal image structure was or was not inverted.

result a vector of [RLERegion](#) structures containing regions.

See also:

[RLERegion](#), [ExtremaParams](#)

6.1.3.14 void extrema::extremaCleanup (bool *detach_only* = false)

Cleans up internal image structure.

Parameters:

detach_only a boolean that specifies if the image is deallocated (false) or not.

6.1.3.15 const ExtremaStats& extrema::extremaStats ()

Returns timing statistics.

Return values:

A structure [ExtremaStats](#).

- 6.1.3.16 void extrema::exportRLEVector (FILE **fid*, vector< RLERegion > &*rle_vector*)
- 6.1.3.17 void extrema::exportBoundaryVector (FILE **fid*, vector< BoundaryRegion > &*boundary_vector*)
- 6.1.3.18 void extrema::exportBoundaryVectorGF (FILE **fid*, vector< BoundaryRegion > &*boundary_vector*)
- 6.1.3.19 void extrema::exportAffVector (FILE **fid*, vector< RLERegion > &*rle_vector*, double *factor*, int *krys_compat*)
- 6.1.3.20 void extrema::RLE2Ellipse (const vector< RLEItem > &*rle*, double &*barX*, double &*barY*, double &*sumX2*, double &*sumXY*, double &*sumY2*)
- 6.1.3.21 void extrema::ReducedBoundary2RLE (vector< BoundaryPoint > &*reduced_boundary*, vector< RLEItem > &*rle*)
- 6.1.3.22 void extrema::FastSetOptThresholds4StableRegion (t_region **p_r*)
- 6.1.3.23 void extrema::CalcHistogram (utls::BAry *&*img*, t_sortpixels &*pixels*)
- 6.1.3.24 void extrema::BinSortPixels (utls::BAry *&*img*, t_sortpixels &*pixels*)
- 6.1.3.25 void extrema::InvertImageAndHistogram (utls::BAry **img*, t_sortpixels &*pixels*)
- 6.1.3.26 void extrema::InitSuballocator (t_suballocator **s*, size_t *blocksize*, size_t *itemsz*, int *clear_blocks* = 0)
- 6.1.3.27 void extrema::DestSuballocator (t_suballocator **s*)
- 6.1.3.28 void extrema::SuballocatorAddBlock (t_suballocator **s*)

6.1.4 Variable Documentation

- 6.1.4.1 t_thresh_par extrema::g_thresh_params

6.2 std Namespace Reference

6.3 utls Namespace Reference

Classes

- struct [Ary](#)
- union [t_rgb](#)
- union [t_frgb](#)
- class [Matrix2](#)

Typedefs

- typedef [Ary](#)< unsigned char > [BAry](#)
- typedef [Ary](#)< int > [IAry](#)
- typedef [Ary](#)< unsigned int > [LAry](#)
- typedef [Ary](#)< float > [FAry](#)
- typedef [Ary](#)< double > [DAry](#)
- typedef [Ary](#)< void * > [PAry](#)
- typedef unsigned char [t_byte](#)

6.3.1 Typedef Documentation

6.3.1.1 typedef [Ary](#)<unsigned char> [utls::BAry](#)

6.3.1.2 typedef [Ary](#)<int> [utls::IAry](#)

6.3.1.3 typedef [Ary](#)<unsigned int> [utls::LAry](#)

6.3.1.4 typedef [Ary](#)<float> [utls::FAry](#)

6.3.1.5 typedef [Ary](#)<double> [utls::DAry](#)

6.3.1.6 typedef [Ary](#)<void *> [utls::PAry](#)

6.3.1.7 typedef unsigned char [utls::t_byte](#)

Chapter 7

Class Documentation

7.1 `utls::Ary< PixelType >` Struct Template Reference

```
#include <ary.h>
```

Public Types

- typedef PixelType `value`
- typedef PixelType * `pointer`
- typedef PixelType ** `row_pointer`
- typedef int `size_type`
- typedef int `coord_type`

Public Member Functions

- `Ary` ()
- `Ary` (`coord_type` fr, `coord_type` lr, `coord_type` fc, `coord_type` lc, `pointer` attach_to=0)
- `Ary` (`coord_type` nrows, `coord_type` ncols, `pointer` attach_to=0)
- `Ary` (const `Ary` &other, bool do_copy=true, bool do_attach=false)
- `~Ary` ()
- `pointer` ptr () const
- `size_type` rows () const
- `size_type` cols () const
- `size_type` size () const
- bool `isin` (`coord_type` row, `coord_type` col) const
- void `cons` (`coord_type` firstrow, `coord_type` lastrow, `coord_type` firstcol, `coord_type` lastcol, `pointer` attach_to)
- void `clear` ()
- void `set` (const PixelType &`value`)
- void `detach` ()
- `Ary` * `copy` () const
- void `copy` (const `Ary` &from)
- void `copy` (const `Ary` *from)
- void `deallocate` ()

Public Attributes

- [coord_type lb1](#)
- [coord_type lb2](#)
- [coord_type ub1](#)
- [coord_type ub2](#)
- [size_type num_rows](#)
- [size_type num_cols](#)
- [pointer data](#)
- [row_pointer el](#)
- [int tag](#)
- [void * user_data](#)

```
template<typename PixelType> struct utls::Ary< PixelType >
```

7.1.1 Member Typedef Documentation

7.1.1.1 `template<typename PixelType> typedef int utls::Ary< PixelType >::coord_type`

7.1.1.2 `template<typename PixelType> typedef PixelType* utls::Ary< PixelType >::pointer`

7.1.1.3 `template<typename PixelType> typedef PixelType** utls::Ary< PixelType >::row_pointer`

7.1.1.4 `template<typename PixelType> typedef int utls::Ary< PixelType >::size_type`

7.1.1.5 `template<typename PixelType> typedef PixelType utls::Ary< PixelType >::value`

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `template<typename PixelType> utls::Ary< PixelType >::Ary () [inline]`

7.1.2.2 `template<typename PixelType> utls::Ary< PixelType >::Ary (coord_type fr, coord_type lr, coord_type fc, coord_type lc, pointer attach_to = 0) [inline]`

7.1.2.3 `template<typename PixelType> utls::Ary< PixelType >::Ary (coord_type nrows, coord_type ncols, pointer attach_to = 0) [inline]`

7.1.2.4 `template<typename PixelType> utls::Ary< PixelType >::Ary (const Ary< PixelType > & other, bool do_copy = true, bool do_attach = false) [inline]`

7.1.2.5 `template<typename PixelType> utls::Ary< PixelType >::~~Ary () [inline]`

7.1.3 Member Function Documentation

7.1.3.1 `template<typename PixelType> void utls::Ary< PixelType >::clear () [inline]`

7.1.3.2 `template<typename PixelType> size_type utls::Ary< PixelType >::cols () const [inline]`

7.1.3.3 `template<typename PixelType> void utls::Ary< PixelType >::cons (coord_type firstrow, coord_type lastrow, coord_type firstcol, coord_type lastcol, pointer attach_to) [inline]`

7.1.3.4 `template<typename PixelType> void utls::Ary< PixelType >::copy (const Ary< PixelType > *from) [inline]`

7.1.3.5 `template<typename PixelType> void utls::Ary< PixelType >::copy (const Ary< PixelType > &from) [inline]`

7.1.3.6 `template<typename PixelType> Ary* utls::Ary< PixelType >::copy () const [inline]`

7.1.3.7 `template<typename PixelType> void utls::Ary< PixelType >::deallocate () [inline]`

7.1.3.8 `template<typename PixelType> void utls::Ary< PixelType >::detach () [inline]`

7.1.3.9 `template<typename PixelType> bool utls::Ary< PixelType >::isin (coord_type row, coord_type col) const [inline]`

Generated on Sun Oct 22 15:33:47 2006 by Doxygen

7.1.3.10 `template<typename PixelType> pointer utls::Ary< PixelType >::ptr () const [inline]`

7.1.3.11 `template<typename PixelType> size_type utls::Ary< PixelType >::rows () const [inline]`

- [ary.h](#)

7.2 extrema::BoundaryExtrema Struct Reference

Old interface structure that holds result of getBoundaryExtrema.

```
#include <libExtrema.h>
```

Public Attributes

- vector< [BoundaryRegion](#) > [MSERplus](#)
- vector< [BoundaryRegion](#) > [MSERmin](#)

7.2.1 Detailed Description

Old interface structure that holds result of getBoundaryExtrema.

7.2.2 Member Data Documentation

7.2.2.1 vector<[BoundaryRegion](#)> [extrema::BoundaryExtrema::MSERmin](#)

7.2.2.2 vector<[BoundaryRegion](#)> [extrema::BoundaryExtrema::MSERplus](#)

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

- [libExtrema.h](#)

7.3 extrema::BoundaryPoint Struct Reference

A structure that holds coordinates of a point in [BoundaryRegion](#) i.e. the boundary representation of a region.

```
#include <libExtrema.h>
```

Public Attributes

- int [line](#)
- int [col](#)

7.3.1 Detailed Description

A structure that holds coordinates of a point in [BoundaryRegion](#) i.e. the boundary representation of a region.

7.3.2 Member Data Documentation

7.3.2.1 int [extrema::BoundaryPoint::col](#)

7.3.2.2 int [extrema::BoundaryPoint::line](#)

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

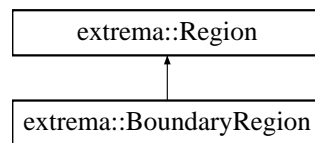
- [libExtrema.h](#)

7.4 extrema::BoundaryRegion Struct Reference

Description of a boundary region.

```
#include <libExtrema.h>
```

Inheritance diagram for extrema::BoundaryRegion::



Public Member Functions

- bool [operator<](#) (const [BoundaryRegion](#) &a) const
Stability ordering operator.

Public Attributes

- vector< [BoundaryPoint](#) > [boundary](#)
Vector that holds boundary representation of the region.

7.4.1 Detailed Description

Description of a boundary region.

7.4.2 Member Function Documentation

7.4.2.1 bool extrema::BoundaryRegion::operator< (const [BoundaryRegion](#) &a) const
[inline]

Stability ordering operator.

7.4.3 Member Data Documentation

7.4.3.1 vector<[BoundaryPoint](#)> extrema::BoundaryRegion::boundary

Vector that holds boundary representation of the region.

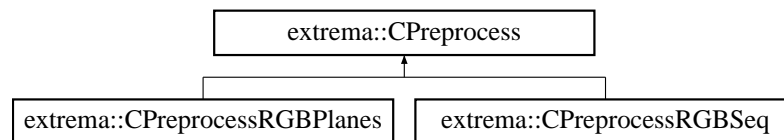
The documentation for this struct was generated from the following file:

- [libExtrema.h](#)

7.5 extrema::CPreprocess Class Reference

```
#include <preprocess.h>
```

Inheritance diagram for extrema::CPreprocess::



Public Member Functions

- double [min](#) (double a, double b) const
- double [max](#) (double a, double b) const
- int [rgb_to_none](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_intensity](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_intensity_half](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_saturation](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_hue](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_red](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_green](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_blue](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [rgb_to_redblue](#) (const unsigned char *r, const unsigned char *g, const unsigned char *b) const
- int [inten_to_none](#) (int intensity) const

7.5.1 Member Function Documentation

- 7.5.1.1 `int extrema::CPreprocess::inten_to_none (int intensity) const` [inline]
- 7.5.1.2 `double extrema::CPreprocess::max (double a, double b) const` [inline]
- 7.5.1.3 `double extrema::CPreprocess::min (double a, double b) const` [inline]
- 7.5.1.4 `int extrema::CPreprocess::rgb_to_blue (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.5 `int extrema::CPreprocess::rgb_to_green (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.6 `int extrema::CPreprocess::rgb_to_hue (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.7 `int extrema::CPreprocess::rgb_to_intensity (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.8 `int extrema::CPreprocess::rgb_to_intensity_half (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.9 `int extrema::CPreprocess::rgb_to_none (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.10 `int extrema::CPreprocess::rgb_to_red (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.11 `int extrema::CPreprocess::rgb_to_redblue (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]
- 7.5.1.12 `int extrema::CPreprocess::rgb_to_saturation (const unsigned char * r, const unsigned char * g, const unsigned char * b) const` [inline]

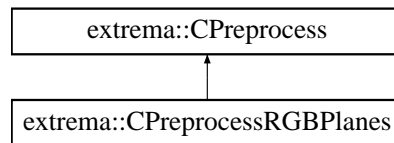
The documentation for this class was generated from the following file:

- [preprocess.h](#)

7.6 extrema::CPreprocessRGBPlanes Class Reference

```
#include <preprocess.h>
```

Inheritance diagram for extrema::CPreprocessRGBPlanes::



Public Member Functions

- void [preprocess](#) (const [ExtremaImage](#) &image, int preprocess_type, int swap, [utls::BAry](#) *&img)

7.6.1 Member Function Documentation

7.6.1.1 void extrema::CPreprocessRGBPlanes::preprocess (const [ExtremaImage](#) & *image*, int *preprocess_type*, int *swap*, [utls::BAry](#) *& *img*)

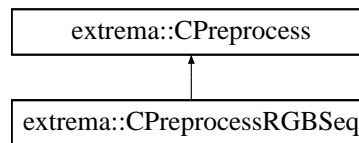
The documentation for this class was generated from the following file:

- [preprocess.h](#)

7.7 extrema::CPreprocessRGBSeq Class Reference

```
#include <preprocess.h>
```

Inheritance diagram for extrema::CPreprocessRGBSeq::



Public Member Functions

- void [preprocess](#) (const [ExtremaImage](#) &image, int preprocess_type, [utls::BAry](#) *&img)

7.7.1 Member Function Documentation

7.7.1.1 void extrema::CPreprocessRGBSeq::preprocess (const [ExtremaImage](#) & *image*, int *preprocess_type*, [utls::BAry](#) *& *img*)

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

- [preprocess.h](#)

7.8 extrema::ExtremaImage Struct Reference

A structure holding image parameters.

```
#include <extremaParams.h>
```

Public Attributes

- unsigned int [width](#)
Width of the image.
- unsigned int [height](#)
Height of the image.
- unsigned int [channels](#)
Number of channels of the image.
- unsigned char * [data](#)
Pointer to image data.

7.8.1 Detailed Description

A structure holding image parameters.

7.8.2 Member Data Documentation

7.8.2.1 unsigned int [extrema::ExtremaImage::channels](#)

Number of channels of the image.

7.8.2.2 unsigned char* [extrema::ExtremaImage::data](#)

Pointer to image data.

7.8.2.3 unsigned int [extrema::ExtremaImage::height](#)

Height of the image.

7.8.2.4 unsigned int [extrema::ExtremaImage::width](#)

Width of the image.

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

- [extremaParams.h](#)

7.9 extrema::ExtremaParams Struct Reference

A structure with MSER detector parameters.

```
#include <extremaParams.h>
```

Public Member Functions

- [ExtremaParams \(\)](#)

Public Attributes

- bool [relative](#)
- int [preprocess](#)
- int [min_size](#)
- double [max_area](#)
- double [min_margin](#)
- bool [verbose](#)
- int [debug](#)

7.9.1 Detailed Description

A structure with MSER detector parameters.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 [extrema::ExtremaParams::ExtremaParams \(\)](#) `[inline]`

7.9.3 Member Data Documentation

7.9.3.1 `int` [extrema::ExtremaParams::debug](#)

7.9.3.2 `double` [extrema::ExtremaParams::max_area](#)

7.9.3.3 `double` [extrema::ExtremaParams::min_margin](#)

7.9.3.4 `int` [extrema::ExtremaParams::min_size](#)

7.9.3.5 `int` [extrema::ExtremaParams::preprocess](#)

7.9.3.6 `bool` [extrema::ExtremaParams::relative](#)

7.9.3.7 `bool` [extrema::ExtremaParams::verbose](#)

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

- [extremaParams.h](#)

7.10 extrema::ExtremaStats Struct Reference

Timing statistics of the detector, gathered only if TIME_STATS is set.

```
#include <libExtrema.h>
```

Public Member Functions

- void [DumpTimeStats](#) (bool both_runs=true)

Public Attributes

- int [num_extrema_replaced](#)
- double [initial_time](#)
- double [preprocess_time](#)
- double [replace_time](#)
- double [chisto_time](#)
- double [histo_time](#)
- double [extrema_p_time](#)
- double [output_p_time](#)
- double [extrema_m_time](#)
- double [output_m_time](#)
- double [total_time](#)

7.10.1 Detailed Description

Timing statistics of the detector, gathered only if TIME_STATS is set.

7.10.2 Member Function Documentation

7.10.2.1 void extrema::ExtremaStats::DumpTimeStats (bool *both_runs* = true)

7.10.3 Member Data Documentation

7.10.3.1 double extrema::ExtremaStats::chisto_time

7.10.3.2 double extrema::ExtremaStats::extrema_m_time

7.10.3.3 double extrema::ExtremaStats::extrema_p_time

7.10.3.4 double extrema::ExtremaStats::histo_time

7.10.3.5 double extrema::ExtremaStats::initial_time

7.10.3.6 int extrema::ExtremaStats::num_extrema_replaced

7.10.3.7 double extrema::ExtremaStats::output_m_time

7.10.3.8 double extrema::ExtremaStats::output_p_time

7.10.3.9 double extrema::ExtremaStats::preprocess_time

7.10.3.10 double extrema::ExtremaStats::replace_time

7.10.3.11 double extrema::ExtremaStats::total_time

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

- [libExtrema.h](#)

7.11 utls::Matrix2 Class Reference

```
#include <matrix.h>
```

Public Member Functions

- [Matrix2](#) (const double a11=0, const double a12=0, const double a21=0, const double a22=0)
- [Matrix2](#) (const double *A)
- [Matrix2](#) (const [Matrix2](#) &other)
- [~Matrix2](#) ()
- void [dump](#) (void) const
- [Matrix2 operator *](#) (const double s) const
- [Matrix2 operator/](#) (const double s) const
- [Matrix2 operator+](#) (const double s) const
- [Matrix2 operator-](#) (const double s) const
- void [operator=](#) (const [Matrix2](#) &other)
- [Matrix2 operator *](#) (const [Matrix2](#) &m) const
- [Matrix2 operator+](#) (const [Matrix2](#) &m) const
- [Matrix2 operator-](#) (const [Matrix2](#) &m) const
- [Matrix2 inv](#) (void) const
- [Matrix2 transpose](#) (void) const
- double [det](#) (void) const
- double [trace](#) (void) const
- [Matrix2 sqrt](#) (void) const
- double [fnorm](#) (void) const
- double * [operator\[\]](#) (int row_idx) const
- void [eye](#) ()
- void [inv_i](#) (void)
- void [transpose_i](#) (void)
- void [sqrt_i](#) (void)
- void [fnormalize](#) (void)
- void [schur_sym](#) ([Matrix2](#) &Q, [Matrix2](#) &T) const
- [Matrix2 chol](#) () const
- void [QR](#) ([Matrix2](#) &Q, [Matrix2](#) &R) const
- void [svd](#) ([Matrix2](#) &U, [Matrix2](#) &S, [Matrix2](#) &V) const

Public Attributes

- double [a](#) [2][2]

7.11.1 Constructor & Destructor Documentation

7.11.1.1 `utls::Matrix2::Matrix2 (const double a11 = 0, const double a12 = 0, const double a21 = 0, const double a22 = 0)`

7.11.1.2 `utls::Matrix2::Matrix2 (const double * A)`

7.11.1.3 `utls::Matrix2::Matrix2 (const Matrix2 & other)`

7.11.1.4 `utls::Matrix2::~~Matrix2 ()`

7.11.2 Member Function Documentation

7.11.2.1 `Matrix2 utls::Matrix2::chol () const`

7.11.2.2 `double utls::Matrix2::det (void) const`

7.11.2.3 `void utls::Matrix2::dump (void) const`

7.11.2.4 `void utls::Matrix2::eye ()`

7.11.2.5 `double utls::Matrix2::fnorm (void) const`

7.11.2.6 `void utls::Matrix2::fnormalize (void)`

7.11.2.7 `Matrix2 utls::Matrix2::inv (void) const`

7.11.2.8 `void utls::Matrix2::inv_i (void)`

7.11.2.9 `Matrix2 utls::Matrix2::operator * (const Matrix2 & m) const`

7.11.2.10 `Matrix2 utls::Matrix2::operator * (const double s) const`

7.11.2.11 `Matrix2 utls::Matrix2::operator+ (const Matrix2 & m) const`

7.11.2.12 `Matrix2 utls::Matrix2::operator+ (const double s) const`

7.11.2.13 `Matrix2 utls::Matrix2::operator- (const Matrix2 & m) const`

7.11.2.14 `Matrix2 utls::Matrix2::operator- (const double s) const`

7.11.2.15 `Matrix2 utls::Matrix2::operator/ (const double s) const`

7.11.2.16 `void utls::Matrix2::operator= (const Matrix2 & other)`

7.11.2.17 `]`

`double* utls::Matrix2::operator[] (int row_idx) const`

7.11.2.18 void utls::Matrix2::QR ([Matrix2](#) & Q , [Matrix2](#) & R) const

7.11.2.19 void utls::Matrix2::schur_sym ([Matrix2](#) & Q , [Matrix2](#) & T) const

7.11.2.20 [Matrix2](#) utls::Matrix2::sqrt (void) const

7.11.2.21 void utls::Matrix2::sqrt_i (void)

7.11.2.22 void utls::Matrix2::svd ([Matrix2](#) & U , [Matrix2](#) & S , [Matrix2](#) & V) const

7.11.2.23 double utls::Matrix2::trace (void) const

7.11.2.24 [Matrix2](#) utls::Matrix2::transpose (void) const

7.11.2.25 void utls::Matrix2::transpose_i (void)

7.11.3 Member Data Documentation

7.11.3.1 double [utls::Matrix2::a](#)[2][2]

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

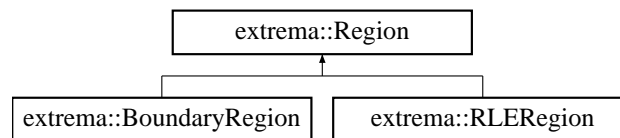
- [matrix.h](#)

7.12 extrema::Region Struct Reference

A structure with common statistics of [BoundaryRegion](#) and [RLERegion](#).

```
#include <libExtrema.h>
```

Inheritance diagram for extrema::Region::



Public Attributes

- int [label](#)
Unique region's label.
- int [minI](#)
Minimum intensity.
- int [maxI](#)
Maximum intensity.
- int [margin](#)
Stability i.e. the length of stable intensities range.
- int [threshold](#)
Thresholded intensity.
- int [extremumX](#)
Position of initial point of the region, a zero-based X coordinate.
- int [extremumY](#)
Position of initial point of the region, a zero-based Y coordinate .
- int [area](#)
A region area at the thresholded intensity level.
- int [border](#)
A region border length at the thresholded intensity level.
- double [cx](#)
Centroid of the region at the thresholded level.
- double [cy](#)
Centroid of the region at the thresholded level.
- double [sxx](#)

Second moments of the region at the thresholded level.

- double `sxy`

Second moments of the region at the thresholded level.

- double `syy`

Second moments of the region at the thresholded level.

- int `rid`

Unique region id, i.e. index of the region.

7.12.1 Detailed Description

A structure with common statistics of `BoundaryRegion` and `RLERegion`.

7.12.2 Member Data Documentation

7.12.2.1 int `extrema::Region::area`

A region area at the thresholded intensity level.

7.12.2.2 int `extrema::Region::border`

A region border length at the thresholded intensity level.

7.12.2.3 double `extrema::Region::cx`

Centroid of the region at the thresholded level.

7.12.2.4 double `extrema::Region::cy`

Centroid of the region at the thresholded level.

7.12.2.5 int `extrema::Region::extremumX`

Position of initial point of the region, a zero-based X coordinate.

7.12.2.6 int `extrema::Region::extremumY`

Position of initial point of the region, a zero-based Y coordinate .

7.12.2.7 int `extrema::Region::label`

Unique region's label.

7.12.2.8 int extrema::Region::margin

Stability i.e. the length of stable intensities range.

7.12.2.9 int extrema::Region::maxI

Maximum intensity.

7.12.2.10 int extrema::Region::minI

Minimum intensity.

7.12.2.11 int extrema::Region::rid

Unique region id, i.e. index of the region.

7.12.2.12 double extrema::Region::sxx

Second moments of the region at the thresholded level.

7.12.2.13 double extrema::Region::sxy

Second moments of the region at the thresholded level.

7.12.2.14 double extrema::Region::syy

Second moments of the region at the thresholded level.

7.12.2.15 int extrema::Region::threshold

Thresholded intensity.

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

- [libExtrema.h](#)

7.13 extrema::RLEExtrema Struct Reference

Old interface structure that holds result of getRLEExtrema.

```
#include <libExtrema.h>
```

Public Attributes

- [vector< RLERegion > MSERplus](#)
- [vector< RLERegion > MSERmin](#)

7.13.1 Detailed Description

Old interface structure that holds result of getRLEExtrema.

7.13.2 Member Data Documentation

7.13.2.1 [vector<RLERegion> extrema::RLEExtrema::MSERmin](#)

7.13.2.2 [vector<RLERegion> extrema::RLEExtrema::MSERplus](#)

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

- [libExtrema.h](#)

7.14 extrema::RLEItem Struct Reference

A structure that holds coordinates of a RLE element in [RLERegion](#) i.e. the RLE representation of a region.

```
#include <libExtrema.h>
```

Public Attributes

- int [line](#)
- int [col1](#)
- int [col2](#)

7.14.1 Detailed Description

A structure that holds coordinates of a RLE element in [RLERegion](#) i.e. the RLE representation of a region.

7.14.2 Member Data Documentation

7.14.2.1 int [extrema::RLEItem::col1](#)

7.14.2.2 int [extrema::RLEItem::col2](#)

7.14.2.3 int [extrema::RLEItem::line](#)

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

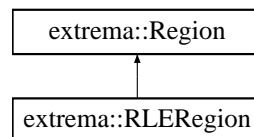
- [libExtrema.h](#)

7.15 extrema::RLERegion Struct Reference

Description of a RLE region.

```
#include <libExtrema.h>
```

Inheritance diagram for extrema::RLERegion::



Public Member Functions

- `bool operator< (const RLERegion &a) const`
Stability ordering operator.

Public Attributes

- `vector< RLEItem > rle`
Vector that holds RLE representation of the region.

7.15.1 Detailed Description

Description of a RLE region.

7.15.2 Member Function Documentation

7.15.2.1 `bool extrema::RLERegion::operator< (const RLERegion &a) const` [inline]

Stability ordering operator.

7.15.3 Member Data Documentation

7.15.3.1 `vector<RLEItem> extrema::RLERegion::rle`

Vector that holds RLE representation of the region.

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

- [libExtrema.h](#)

7.16 extrema::s_borderpixel Struct Reference

Structure with pixel of the extended boundary.

```
#include <extremaTypes.h>
```

Public Member Functions

- bool [operator<](#) (const [s_borderpixel](#) &other) const

Public Attributes

- [t_ipoint](#) pos
- unsigned char [direct](#)

7.16.1 Detailed Description

Structure with pixel of the extended boundary.

7.16.2 Member Function Documentation

7.16.2.1 bool extrema::s_borderpixel::operator< (const [s_borderpixel](#) & *other*) const [inline]

7.16.3 Member Data Documentation

7.16.3.1 unsigned char [extrema::s_borderpixel::direct](#)

7.16.3.2 [t_ipoint](#) [extrema::s_borderpixel::pos](#)

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

- [extremaTypes.h](#)

7.17 extrema::s_region Struct Reference

Internal region structure.

```
#include <extremaTypes.h>
```

Public Attributes

- [t_label](#) label
- [int](#) [minimum_int](#)
- [int](#) [pixel_total](#)
- [int](#) [border_total](#)
- [t_ipoint](#) [minimum_pos](#)
- [int](#) [maximum_int](#)
- [t_label](#) [merge_label](#)
- [t_LL](#) [thresholds](#)
- [int](#) [pixels](#) [[c_maxByte](#)]
- [int](#) [borders](#) [[c_maxByte](#)]

7.17.1 Detailed Description

Internal region structure.

7.17.2 Member Data Documentation

7.17.2.1 [int](#) [extrema::s_region::border_total](#)

7.17.2.2 [int](#) [extrema::s_region::borders](#)[[c_maxByte](#)]

7.17.2.3 [t_label](#) [extrema::s_region::label](#)

7.17.2.4 [int](#) [extrema::s_region::maximum_int](#)

7.17.2.5 [t_label](#) [extrema::s_region::merge_label](#)

7.17.2.6 [int](#) [extrema::s_region::minimum_int](#)

7.17.2.7 [t_ipoint](#) [extrema::s_region::minimum_pos](#)

7.17.2.8 [int](#) [extrema::s_region::pixel_total](#)

7.17.2.9 [int](#) [extrema::s_region::pixels](#)[[c_maxByte](#)]

7.17.2.10 [t_LL](#) [extrema::s_region::thresholds](#)

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

- [extremaTypes.h](#)

7.18 extrema::s_region_equiv Struct Reference

Internal structure with a node of the label equivalency tree.

```
#include <extremaTypes.h>
```

Public Attributes

- unsigned int [pred](#)
- [t_region](#)* [region](#)

7.18.1 Detailed Description

Internal structure with a node of the label equivalency tree.

7.18.2 Member Data Documentation

7.18.2.1 unsigned int [extrema::s_region_equiv::pred](#)

7.18.2.2 [t_region](#)* [extrema::s_region_equiv::region](#)

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

- [extremaTypes.h](#)

7.19 extrema::s_sortpixels Struct Reference

Internal structure with intensity histogram.

```
#include <extremaTypes.h>
```

Public Attributes

- [t_ipoint](#) * [data](#) [c_maxByte]
- int [hist](#) [c_maxByte]

7.19.1 Detailed Description

Internal structure with intensity histogram.

7.19.2 Member Data Documentation

7.19.2.1 [t_ipoint](#)* [extrema::s_sortpixels::data](#)[c_maxByte]

7.19.2.2 int [extrema::s_sortpixels::hist](#)[c_maxByte]

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

- [extremaTypes.h](#)

7.20 extrema::s_suballocator Struct Reference

Internal structure with suballocator's data.

```
#include <suballoc.h>
```

Public Attributes

- [t_LL items](#)
- [t_LL blocks](#)
- [size_t free_items](#)
- [size_t item_size](#)
- [size_t block_size](#)
- [int clear_blocks](#)

7.20.1 Detailed Description

Internal structure with suballocator's data.

7.20.2 Member Data Documentation

7.20.2.1 [size_t extrema::s_suballocator::block_size](#)

7.20.2.2 [t_LL extrema::s_suballocator::blocks](#)

7.20.2.3 [int extrema::s_suballocator::clear_blocks](#)

7.20.2.4 [size_t extrema::s_suballocator::free_items](#)

7.20.2.5 [size_t extrema::s_suballocator::item_size](#)

7.20.2.6 [t_LL extrema::s_suballocator::items](#)

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

- [suballoc.h](#)

7.21 extrema::s_thresh_def Struct Reference

Internal structure holding threshold paramaters.

```
#include <extremaTypes.h>
```

Public Attributes

- int [thresh](#)
- int [pos](#)
- int [margin](#)
- [point_vector](#) * [boundary](#)

7.21.1 Detailed Description

Internal structure holding threshold paramaters.

7.21.2 Member Data Documentation

7.21.2.1 [point_vector*](#) [extrema::s_thresh_def::boundary](#)

7.21.2.2 int [extrema::s_thresh_def::margin](#)

7.21.2.3 int [extrema::s_thresh_def::pos](#)

7.21.2.4 int [extrema::s_thresh_def::thresh](#)

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

- [extremaTypes.h](#)

7.22 extrema::s_thresh_par Struct Reference

Internal structure with processed detector's parameters.

```
#include <extremaTypes.h>
```

Public Attributes

- int [min_size](#)
minimum size of the region in pixels
- int [max_size](#)
maximum size of the region in pixels
- double [min_margin](#)
minimum margin and upper boundary for hysteresis thresholding
- bool [relative_margin](#)
margin relative to intensity level
- int [invert](#)
do inverted margin

7.22.1 Detailed Description

Internal structure with processed detector's parameters.

7.22.2 Member Data Documentation

7.22.2.1 int [extrema::s_thresh_par::invert](#)

do inverted margin

7.22.2.2 int [extrema::s_thresh_par::max_size](#)

maximum size of the region in pixels

7.22.2.3 double [extrema::s_thresh_par::min_margin](#)

minimum margin and upper boundary for hysteresis thresholding

7.22.2.4 int [extrema::s_thresh_par::min_size](#)

minimum size of the region in pixels

7.22.2.5 **bool** [extrema::s_thresh_par::relative_margin](#)

margin relative to intensity level

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

- [extremaTypes.h](#)

7.23 utls::t_frgb Union Reference

```
#include <dtypes.h>
```

Public Attributes

- float [arr](#) [3]
- struct {
 - float [r](#)
 - float [g](#)
 - float [b](#)
- } [st](#)

7.23.1 Member Data Documentation

7.23.1.1 float [utls::t_frgb::arr](#)[3]

7.23.1.2 float [utls::t_frgb::b](#)

7.23.1.3 float [utls::t_frgb::g](#)

7.23.1.4 float [utls::t_frgb::r](#)

7.23.1.5 struct { ... } [utls::t_frgb::st](#)

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

- [dtypes.h](#)

7.24 extrema::t_ipoint Struct Reference

Internal structure, holds 2D point coordinates.

```
#include <extremaTypes.h>
```

Public Attributes

- [int x](#)
- [int y](#)

7.24.1 Detailed Description

Internal structure, holds 2D point coordinates.

7.24.2 Member Data Documentation

7.24.2.1 [int extrema::t_ipoint::x](#)

7.24.2.2 [int extrema::t_ipoint::y](#)

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

- [extremaTypes.h](#)

7.25 utls::t_rgb Union Reference

```
#include <dtypes.h>
```

Public Attributes

- [t_byte arr](#) [3]
- struct {
 - [t_byte r](#)
 - [t_byte g](#)
 - [t_byte b](#)
- } [st](#)

7.25.1 Member Data Documentation

7.25.1.1 [t_byte utls::t_rgb::arr](#)[3]

7.25.1.2 [t_byte utls::t_rgb::b](#)

7.25.1.3 [t_byte utls::t_rgb::g](#)

7.25.1.4 [t_byte utls::t_rgb::r](#)

7.25.1.5 struct { ... } [utls::t_rgb::st](#)

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

- [dtypes.h](#)

Chapter 8

File Documentation

8.1 ary.h File Reference

```
#include <assert.h>
#include <cmath>
#include <stdio.h>
#include "dtypes.h"
```

Namespaces

- namespace [utls](#)

Classes

- struct [utls::Ary< PixelType >](#)

Typedefs

- typedef Ary< unsigned char > [utls::BAry](#)
- typedef Ary< int > [utls::IAry](#)
- typedef Ary< unsigned int > [utls::LAry](#)
- typedef Ary< float > [utls::FAry](#)
- typedef Ary< double > [utls::DAry](#)
- typedef Ary< void * > [utls::PAry](#)

8.2 boundary.h File Reference

```
#include <ary.h>
#include <LL.h>
#include "extremaTypes.h"
```

Namespaces

- namespace [extrema](#)

Functions

- void [extrema::RegionBoundaries](#) ([utls::BAry](#) *img, t_LL regions)

8.3 common.oxy File Reference

8.4 dtypes.h File Reference

Namespaces

- namespace [utls](#)

Classes

- union [utls::t_rgb](#)
- union [utls::t_frgr](#)

Typedefs

- typedef unsigned char [utls::t_byte](#)

8.5 extremaConfig.h File Reference

Defines

- #define [TIME_STATS](#) 1

8.5.1 Define Documentation

8.5.1.1 #define TIME_STATS 1

8.6 extremaParams.h File Reference

```
#include "extremaConfig.h"
```

Namespaces

- namespace [extrema](#)

Classes

- struct [extrema::ExtremaImage](#)
A structure holding image parameters.
- struct [extrema::ExtremaParams](#)
A structure with MSER detector parameters.

Defines

- #define [GENERATE_MSER_PLUS](#) 1
- #define [GENERATE_MSER_MINUS](#) 2

Enumerations

- enum [extrema::EXTREMA_PREPROCESS](#) {
 [extrema::PREPROCESS_CHANNEL_none](#) = 0x00000000,
 [extrema::PREPROCESS_CHANNEL_intensity](#) = 0x00000001,
 [extrema::PREPROCESS_CHANNEL_saturation](#) = 0x00000002,
 [extrema::PREPROCESS_CHANNEL_hue](#) = 0x00000003,
 [extrema::PREPROCESS_CHANNEL_redblue](#) = 0x00000004,
 [extrema::PREPROCESS_CHANNEL_red](#) = 0x00000005,
 [extrema::PREPROCESS_CHANNEL_green](#) = 0x00000006,
 [extrema::PREPROCESS_CHANNEL_blue](#) = 0x00000007,
 [extrema::PREPROCESS_CHANNEL_greenmagenta](#) = 0x00000008,
 [extrema::PREPROCESS_CHANNEL_intensity_half](#) = 0x00000009,
 [extrema::PREPROCESS_CHANNEL_MASK](#) = 0x0000ffff,
 [extrema::PREPROCESS_INTENSITY_none](#) = 0x00000000,
 [extrema::PREPROCESS_INTENSITY_MASK](#) = 0xffff0000 }
An enumeration that encodes different preprocessings of an image.

8.6.1 Define Documentation

8.6.1.1 #define GENERATE_MSER_MINUS 2

8.6.1.2 #define GENERATE_MSER_PLUS 1

8.7 extremaTypes.h File Reference

```
#include <LL.h>
#include <vector>
#include "extremaConfig.h"
```

Namespaces

- namespace [extrema](#)

Classes

- struct [extrema::t_ipoint](#)
Internal structure, holds 2D point coordinates.
- struct [extrema::s_sortpixels](#)
Internal structure with intensity histogram.
- struct [extrema::s_region](#)
Internal region structure.
- struct [extrema::s_region_equiv](#)
Internal structure with a node of the label equivalency tree.
- struct [extrema::s_thresh_par](#)
Internal structure with processed detector's parameters.
- struct [extrema::s_borderpixel](#)
Structure with pixel of the extended boundary.
- struct [extrema::s_thresh_def](#)
Internal structure holding threshold paramaters.

Defines

- #define [c_maxByte](#) 256

Typedefs

- typedef unsigned int [extrema::t_label](#)
- typedef [extrema::s_sortpixels](#) [extrema::t_sortpixels](#)
Internal structure with intensity histogram.
- typedef unsigned int [extrema::t_mregion](#)
- typedef [extrema::s_region](#) [extrema::t_region](#)
Internal region structure.

- typedef [extrema::s_region_equiv](#) [extrema::t_region_equiv](#)
Internal structure with a node of the label equivalency tree.
- typedef [extrema::s_thresh_par](#) [extrema::t_thresh_par](#)
Internal structure with processed detector's parameters.
- typedef [extrema::s_borderpixel](#) [extrema::t_borderpixel](#)
Structure with pixel of the extended boundary.
- typedef `std::vector< t_borderpixel >` [extrema::point_vector](#)
Vector with extended boundary.
- typedef [extrema::s_thresh_def](#) [extrema::t_thresh_def](#)
Internal structure holding threshold paramaters.

8.7.1 Define Documentation

8.7.1.1 `#define c_maxByte 256`

8.8 getExtrema.h File Reference

```
#include <LL.h>
#include <ary.h>
#include "extremaParams.h"
#include "extremaTypes.h"
#include "suballoc.h"
```

Namespaces

- namespace [extrema](#)

Functions

- void [extrema::InitRegionRecycling](#) ()
- void [extrema::DestRegionRecycling](#) ()
- t_LL [extrema::GetExtrema](#) (utls::BAry *img, [t_sortpixels](#) pixels, const ExtremaParams &par, bool invert)
- void [extrema::DestRegions](#) (t_LL regions)

Variables

- [t_thresh_par](#) [extrema::g_thresh_params](#)

8.9 libExtrema.h File Reference

```
#include <algorithm>
#include <vector>
#include <stdio.h>
#include <ary.h>
#include "extremaConfig.h"
#include "extremaParams.h"
```

Namespaces

- namespace [extrema](#)
- namespace [std](#)

Classes

- struct [extrema::BoundaryPoint](#)
A structure that holds coordinates of a point in [BoundaryRegion](#) i.e. the boundary representation of a region.
- struct [extrema::RLEItem](#)
A structure that holds coordinates of a RLE element in [RLERegion](#) i.e. the RLE representation of a region.
- struct [extrema::Region](#)
A structure with common statistics of [BoundaryRegion](#) and [RLERegion](#).
- struct [extrema::RLERegion](#)
Description of a RLE region.
- struct [extrema::BoundaryRegion](#)
Description of a boundary region.
- struct [extrema::ExtremaStats](#)
Timing statistics of the detector, gathered only if `TIME_STATS` is set.
- struct [extrema::RLEExtrema](#)
Old interface structure that holds result of `getRLEExtrema`.
- struct [extrema::BoundaryExtrema](#)
Old interface structure that holds result of `getBoundaryExtrema`.

Functions

- BoundaryExtrema [extrema::getBoundaryExtrema](#) (const ExtremaParams ¶ms, const ExtremaImage &image, int both=3)
Old interface function, produces BoundaryRegions for a given image. Computes MSERs: both=1 +, both=2 -, or both=3 + and -.

- RLEExtrema [extrema::getRLEExtrema](#) (const ExtremaParams &par, const ExtremaImage &image, int both=3)
Old interface function, produces RLERegions for a given image. Computes MSERs: both=1 +, both = 2 -, or both = 3 + and -.
- void [extrema::extremaPrepareImage](#) (const ExtremaParams ¶ms, const ExtremaImage &image)
Prepares image for detection of MSER regions.
- void [extrema::extremaAttachImage](#) (const ExtremaParams ¶ms, [utls::BArray](#) *image)
Assigns already prepared image for detection of MSER regions.
- void [extrema::extremaInvertImage](#) ()
Inverts image in internal structure.
- void [extrema::extremaBoundaryRegions](#) (const ExtremaParams ¶ms, bool inverted, vector< BoundaryRegion > &result)
Detects MSERs inverted or not inverted image.
- void [extrema::extremaBoundaryEllRegions](#) (const ExtremaParams ¶ms, bool inverted, vector< BoundaryRegion > &result)
Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.
- void [extrema::extremaRLERegions](#) (const ExtremaParams ¶ms, bool inverted, vector< RLERegion > &result)
Detects MSERs inverted or not inverted image and computes centroids and second moments of each region.
- void [extrema::extremaCleanup](#) (bool detach_only=false)
Cleans up internal image structure.
- const ExtremaStats & [extrema::extremaStats](#) ()
Returns timing statistics.
- void [extrema::exportRLEVector](#) (FILE *fid, vector< RLERegion > &rle_vector)
- void [extrema::exportBoundaryVector](#) (FILE *fid, vector< BoundaryRegion > &boundary_vector)
- void [extrema::exportBoundaryVectorGF](#) (FILE *fid, vector< BoundaryRegion > &boundary_vector)
- void [extrema::exportAffVector](#) (FILE *fid, vector< RLERegion > &rle_vector, double factor, int kryes_compat)
- void [extrema::RLE2Ellipse](#) (const vector< RLEItem > &rle, double &barX, double &barY, double &sumX2, double &sumXY, double &sumY2)
- void [extrema::ReducedBoundary2RLE](#) (vector< BoundaryPoint > &reduced_boundary, vector< RLEItem > &rle)

8.9.1 Detailed Description

Interface of MSER detector.

This file contains main external interface of the MSERs detector.

8.10 matrix.h File Reference

```
#include <math.h>
```

Namespaces

- namespace [utls](#)

Classes

- class [utls::Matrix2](#)

8.11 optThresh.h File Reference

```
#include "extremaTypes.h"
```

Namespaces

- namespace [extrema](#)

Functions

- void [extrema::FastSetOptThresholds4StableRegion](#) (t_region *p_r)

8.12 preprocess.h File Reference

```
#include <math.h>
#include <ary.h>
#include "extremaTypes.h"
#include "extremaParams.h"
```

Namespaces

- namespace [extrema](#)

Classes

- class [extrema::CPreprocess](#)
- class [extrema::CPreprocessRGBSeq](#)
- class [extrema::CPreprocessRGBPlanes](#)

Defines

- #define [M_PI](#) 3.14159265358979323846
- #define [def_preprocess_function_seq](#)(channel_conversion, intensity_processing)
- #define [def_preprocess_function_plan](#)(channel_conversion, intensity_processing)

8.12.1 Define Documentation

8.12.1.1 #define [def_preprocess_function_plan](#)(channel_conversion, intensity_processing)

Value:

```
void preprocess_## channel_conversion ## _ ## intensity_processing \
(const ExtremaImage &image, int preprocess_type, int swap, utls::BAry *&img);
```

8.12.1.2 #define [def_preprocess_function_seq](#)(channel_conversion, intensity_processing)

Value:

```
void preprocess_## channel_conversion ## _ ## intensity_processing \
(const ExtremaImage &image, int preprocess_type, utls::BAry *&img);
```

8.12.1.3 #define [M_PI](#) 3.14159265358979323846

8.13 sortPixels.h File Reference

```
#include <ary.h>
#include "extremaTypes.h"
```

Namespaces

- namespace [extrema](#)

Functions

- void [extrema::CalcHistogram](#) (utls::BAry *&img, [t_sortpixels](#) &pixels)
- void [extrema::BinSortPixels](#) (utls::BAry *&img, [t_sortpixels](#) &pixels)
- void [extrema::InvertImageAndHistogram](#) (utls::BAry *img, [t_sortpixels](#) &pixels)

8.14 suballoc.h File Reference

```
#include <LL.h>
#include <string.h>
```

Namespaces

- namespace [extrema](#)

Classes

- struct [extrema::s_suballocator](#)
Internal structure with suballocator's data.

Typedefs

- typedef [extrema::s_suballocator](#) [extrema::t_suballocator](#)
Internal structure with suballocator's data.

Functions

- void [extrema::InitSuballocator](#) ([t_suballocator](#) *s, [size_t](#) blocksize, [size_t](#) itemsize, int clear_blocks=0)
- void [extrema::DestSuballocator](#) ([t_suballocator](#) *s)
- void [extrema::SuballocatorAddBlock](#) ([t_suballocator](#) *s)

8.15 timeutls.h File Reference

Functions

- double [get_time](#) ()

8.15.1 Function Documentation

8.15.1.1 double [get_time](#) ()

Index

- ~Ary
 - utls::Ary, [25](#)
- ~Matrix2
 - utls::Matrix2, [39](#)
- a
 - utls::Matrix2, [40](#)
- area
 - extrema::Region, [42](#)
- arr
 - utls::t_frpb, [55](#)
 - utls::t_rgb, [57](#)
- Ary
 - utls::Ary, [25](#)
- ary.h, [59](#)
- b
 - utls::t_frpb, [55](#)
 - utls::t_rgb, [57](#)
- BAry
 - utls, [21](#)
- BinSortPixels
 - extrema, [19](#)
- block_size
 - extrema::s_suballocator, [51](#)
- blocks
 - extrema::s_suballocator, [51](#)
- border
 - extrema::Region, [42](#)
- border_total
 - extrema::s_region, [48](#)
- borders
 - extrema::s_region, [48](#)
- boundary
 - extrema::BoundaryRegion, [29](#)
 - extrema::s_thresh_def, [52](#)
- boundary.h, [60](#)
- c_maxByte
 - extremaTypes.h, [66](#)
- CalcHistogram
 - extrema, [19](#)
- channels
 - extrema::ExtremaImage, [34](#)
- chisto_time
 - extrema::ExtremaStats, [37](#)
- chol
 - utls::Matrix2, [39](#)
- clear
 - utls::Ary, [25](#)
- clear_blocks
 - extrema::s_suballocator, [51](#)
- col
 - extrema::BoundaryPoint, [28](#)
- col1
 - extrema::RLEItem, [45](#)
- col2
 - extrema::RLEItem, [45](#)
- cols
 - utls::Ary, [25](#)
- common.oxy, [61](#)
- cons
 - utls::Ary, [25](#)
- coord_type
 - utls::Ary, [25](#)
- copy
 - utls::Ary, [25](#)
- cx
 - extrema::Region, [42](#)
- cy
 - extrema::Region, [42](#)
- DAry
 - utls, [21](#)
- data
 - extrema::ExtremaImage, [34](#)
 - extrema::s_sortpixels, [50](#)
 - utls::Ary, [25](#)
- deallocate
 - utls::Ary, [25](#)
- debug
 - extrema::ExtremaParams, [35](#)
- def_preprocess_function_plan
 - preprocess.h, [72](#)
- def_preprocess_function_seq
 - preprocess.h, [72](#)
- DestRegionRecycling
 - extrema, [16](#)
- DestRegions
 - extrema, [16](#)

- DestSuballocator
 - extrema, 19
- det
 - utls::Matrix2, 39
- detach
 - utls::Ary, 25
- direct
 - extrema::s_borderpixel, 47
- dtypes.h, 62
- dump
 - utls::Matrix2, 39
- DumpTimeStats
 - extrema::ExtremaStats, 37
- el
 - utls::Ary, 25
- exportAffVector
 - extrema, 19
- exportBoundaryVector
 - extrema, 19
- exportBoundaryVectorGF
 - extrema, 19
- exportRLEVector
 - extrema, 18
- extrema, 11
 - BinSortPixels, 19
 - CalcHistogram, 19
 - DestRegionRecycling, 16
 - DestRegions, 16
 - DestSuballocator, 19
 - exportAffVector, 19
 - exportBoundaryVector, 19
 - exportBoundaryVectorGF, 19
 - exportRLEVector, 18
 - EXTREMA_PREPROCESS, 15
 - extremaAttachImage, 17
 - extremaBoundaryEllRegions, 17
 - extremaBoundaryRegions, 17
 - extremaCleanup, 18
 - extremaInvertImage, 17
 - extremaPrepareImage, 16
 - extremaRLERegions, 18
 - extremaStats, 18
 - FastSetOptThresholds4StableRegion, 19
 - g_thresh_params, 19
 - getBoundaryExtrema, 16
 - GetExtrema, 16
 - getRLEExtrema, 16
 - InitRegionRecycling, 16
 - InitSuballocator, 19
 - InvertImageAndHistogram, 19
 - point_vector, 15
 - PREPROCESS_CHANNEL_blue, 16
 - PREPROCESS_CHANNEL_green, 16
 - PREPROCESS_CHANNEL_greenmagenta, 16
 - PREPROCESS_CHANNEL_hue, 16
 - PREPROCESS_CHANNEL_intensity, 15
 - PREPROCESS_CHANNEL_intensity_half, 16
 - PREPROCESS_CHANNEL_MASK, 16
 - PREPROCESS_CHANNEL_none, 15
 - PREPROCESS_CHANNEL_red, 16
 - PREPROCESS_CHANNEL_redblue, 16
 - PREPROCESS_CHANNEL_saturation, 15
 - PREPROCESS_INTENSITY_MASK, 16
 - PREPROCESS_INTENSITY_none, 16
 - ReducedBoundary2RLE, 19
 - RegionBoundaries, 16
 - RLE2Ellipse, 19
 - SuballocatorAddBlock, 19
 - t_borderpixel, 15
 - t_label, 15
 - t_mregion, 15
 - t_region, 15
 - t_region_equiv, 15
 - t_sortpixels, 15
 - t_suballocator, 15
 - t_thresh_def, 15
 - t_thresh_par, 15
- extrema::BoundaryExtrema, 27
- extrema::BoundaryExtrema
 - MSERmin, 27
 - MSERplus, 27
- extrema::BoundaryPoint, 28
- extrema::BoundaryPoint
 - col, 28
 - line, 28
- extrema::BoundaryRegion, 29
- extrema::BoundaryRegion
 - boundary, 29
 - operator<, 29
- extrema::CPreprocess, 30
 - inten_to_none, 31
 - max, 31
 - min, 31
 - rgb_to_blue, 31
 - rgb_to_green, 31
 - rgb_to_hue, 31
 - rgb_to_intensity, 31
 - rgb_to_intensity_half, 31
 - rgb_to_none, 31
 - rgb_to_red, 31
 - rgb_to_redblue, 31
 - rgb_to_saturation, 31
- extrema::CPreprocessRGBPlanes, 32
- extrema::CPreprocessRGBPlanes
 - preprocess, 32

- extrema::CPreprocessRGBSeq, 33
- extrema::CPreprocessRGBSeq
 - preprocess, 33
- extrema::ExtremaImage, 34
- extrema::ExtremaImage
 - channels, 34
 - data, 34
 - height, 34
 - width, 34
- extrema::ExtremaParams, 35
- extrema::ExtremaParams
 - debug, 35
 - ExtremaParams, 35
 - max_area, 35
 - min_margin, 35
 - min_size, 35
 - preprocess, 35
 - relative, 35
 - verbose, 35
- extrema::ExtremaStats, 36
- extrema::ExtremaStats
 - chisto_time, 37
 - DumpTimeStats, 37
 - extrema_m_time, 37
 - extrema_p_time, 37
 - histo_time, 37
 - initial_time, 37
 - num_extrema_replaced, 37
 - output_m_time, 37
 - output_p_time, 37
 - preprocess_time, 37
 - replace_time, 37
 - total_time, 37
- extrema::Region, 41
 - area, 42
 - border, 42
 - cx, 42
 - cy, 42
 - extremumX, 42
 - extremumY, 42
 - label, 42
 - margin, 42
 - maxI, 43
 - minI, 43
 - rid, 43
 - sxx, 43
 - sxy, 43
 - syy, 43
 - threshold, 43
- extrema::RLEExtrema, 44
 - MSERmin, 44
 - MSERplus, 44
- extrema::RLEItem, 45
 - col1, 45
 - col2, 45
 - line, 45
- extrema::RLERegion, 46
 - operator<, 46
 - rle, 46
- extrema::s_borderpixel, 47
 - direct, 47
 - operator<, 47
 - pos, 47
- extrema::s_region, 48
 - border_total, 48
 - borders, 48
 - label, 48
 - maximum_int, 48
 - merge_label, 48
 - minimum_int, 48
 - minimum_pos, 48
 - pixel_total, 48
 - pixels, 48
 - thresholds, 48
- extrema::s_region_equiv, 49
 - pred, 49
 - region, 49
- extrema::s_sortpixels, 50
 - data, 50
 - hist, 50
- extrema::s_suballocator, 51
 - block_size, 51
 - blocks, 51
 - clear_blocks, 51
 - free_items, 51
 - item_size, 51
 - items, 51
- extrema::s_thresh_def, 52
 - boundary, 52
 - margin, 52
 - pos, 52
 - thresh, 52
- extrema::s_thresh_par, 53
 - invert, 53
 - max_size, 53
 - min_margin, 53
 - min_size, 53
 - relative_margin, 53
- extrema::t_ipoint, 56
 - x, 56
 - y, 56
- extrema_m_time
 - extrema::ExtremaStats, 37
- extrema_p_time
 - extrema::ExtremaStats, 37
- EXTREMA_PREPROCESS
 - extrema, 15
- extremaAttachImage

- extrema, 17
- extremaBoundaryEllRegions
 - extrema, 17
- extremaBoundaryRegions
 - extrema, 17
- extremaCleanup
 - extrema, 18
- extremaConfig.h, 63
- extremaConfig.h
 - TIME_STATS, 63
- extremaInvertImage
 - extrema, 17
- ExtremaParams
 - extrema::ExtremaParams, 35
- extremaParams.h, 64
- extremaParams.h
 - GENERATE_MSER_MINUS, 64
 - GENERATE_MSER_PLUS, 64
- extremaPrepareImage
 - extrema, 16
- extremaRLERegions
 - extrema, 18
- extremaStats
 - extrema, 18
- extremaTypes.h, 65
- extremaTypes.h
 - c_maxByte, 66
- extremumX
 - extrema::Region, 42
- extremumY
 - extrema::Region, 42
- eye
 - utls::Matrix2, 39
- FAry
 - utls, 21
- FastSetOptThresholds4StableRegion
 - extrema, 19
- fnorm
 - utls::Matrix2, 39
- fnormalize
 - utls::Matrix2, 39
- free_items
 - extrema::s_suballocator, 51
- g
 - utls::t_frgb, 55
 - utls::t_rgb, 57
- g_thresh_params
 - extrema, 19
- GENERATE_MSER_MINUS
 - extremaParams.h, 64
- GENERATE_MSER_PLUS
 - extremaParams.h, 64
- get_time
 - timeutls.h, 75
- getBoundaryExtrema
 - extrema, 16
- GetExtrema
 - extrema, 16
- getExtrema.h, 67
- getRLEExtrema
 - extrema, 16
- height
 - extrema::ExtremaImage, 34
- hist
 - extrema::s_sortpixels, 50
- histo_time
 - extrema::ExtremaStats, 37
- IAry
 - utls, 21
- initial_time
 - extrema::ExtremaStats, 37
- InitRegionRecycling
 - extrema, 16
- InitSuballocator
 - extrema, 19
- inten_to_none
 - extrema::CPreprocess, 31
- inv
 - utls::Matrix2, 39
- inv_i
 - utls::Matrix2, 39
- invert
 - extrema::s_thresh_par, 53
- InvertImageAndHistogram
 - extrema, 19
- isin
 - utls::Ary, 25
- item_size
 - extrema::s_suballocator, 51
- items
 - extrema::s_suballocator, 51
- label
 - extrema::Region, 42
 - extrema::s_region, 48
- LAry
 - utls, 21
- lb1
 - utls::Ary, 25
- lb2
 - utls::Ary, 25
- libExtrema.h, 68
- line
 - extrema::BoundaryPoint, 28

- extrema::RLEItem, 45
- M_PI
 - preprocess.h, 72
- margin
 - extrema::Region, 42
 - extrema::s_thresh_def, 52
- matrix.h, 70
- Matrix2
 - utls::Matrix2, 39
- max
 - extrema::CPreprocess, 31
- max_area
 - extrema::ExtremaParams, 35
- max_size
 - extrema::s_thresh_par, 53
- maxI
 - extrema::Region, 43
- maximum_int
 - extrema::s_region, 48
- merge_label
 - extrema::s_region, 48
- min
 - extrema::CPreprocess, 31
- min_margin
 - extrema::ExtremaParams, 35
 - extrema::s_thresh_par, 53
- min_size
 - extrema::ExtremaParams, 35
 - extrema::s_thresh_par, 53
- minI
 - extrema::Region, 43
- minimum_int
 - extrema::s_region, 48
- minimum_pos
 - extrema::s_region, 48
- MSERmin
 - extrema::BoundaryExtrema, 27
 - extrema::RLEExtrema, 44
- MSERplus
 - extrema::BoundaryExtrema, 27
 - extrema::RLEExtrema, 44
- num_cols
 - utls::Ary, 25
- num_extrema_replaced
 - extrema::ExtremaStats, 37
- num_rows
 - utls::Ary, 25
- operator *
 - utls::Matrix2, 39
- operator+
 - utls::Matrix2, 39
- operator-
 - utls::Matrix2, 39
- operator/
 - utls::Matrix2, 39
- operator<
 - extrema::BoundaryRegion, 29
 - extrema::RLERegion, 46
 - extrema::s_borderpixel, 47
- operator=
 - utls::Matrix2, 39
- operator[]
 - utls::Matrix2, 39
- optThresh.h, 71
- output_m_time
 - extrema::ExtremaStats, 37
- output_p_time
 - extrema::ExtremaStats, 37
- PAry
 - utls, 21
- pixel_total
 - extrema::s_region, 48
- pixels
 - extrema::s_region, 48
- point_vector
 - extrema, 15
- pointer
 - utls::Ary, 25
- pos
 - extrema::s_borderpixel, 47
 - extrema::s_thresh_def, 52
- pred
 - extrema::s_region_equiv, 49
- preprocess
 - extrema::CPreprocessRGBPlanes, 32
 - extrema::CPreprocessRGBSeq, 33
 - extrema::ExtremaParams, 35
- preprocess.h, 72
 - def_preprocess_function_plan, 72
 - def_preprocess_function_seq, 72
 - M_PI, 72
- PREPROCESS_CHANNEL_blue
 - extrema, 16
- PREPROCESS_CHANNEL_green
 - extrema, 16
- PREPROCESS_CHANNEL_greenmagenta
 - extrema, 16
- PREPROCESS_CHANNEL_hue
 - extrema, 16
- PREPROCESS_CHANNEL_intensity
 - extrema, 15
- PREPROCESS_CHANNEL_intensity_half
 - extrema, 16
- PREPROCESS_CHANNEL_MASK

- extrema, 16
- PREPROCESS_CHANNEL_none
 - extrema, 15
- PREPROCESS_CHANNEL_red
 - extrema, 16
- PREPROCESS_CHANNEL_redblue
 - extrema, 16
- PREPROCESS_CHANNEL_saturation
 - extrema, 15
- PREPROCESS_INTENSITY_MASK
 - extrema, 16
- PREPROCESS_INTENSITY_none
 - extrema, 16
- preprocess_time
 - extrema::ExtremaStats, 37
- ptr
 - utls::Ary, 25
- QR
 - utls::Matrix2, 39
- r
 - utls::t_frgb, 55
 - utls::t_rgb, 57
- ReducedBoundary2RLE
 - extrema, 19
- region
 - extrema::s_region_equiv, 49
- RegionBoundaries
 - extrema, 16
- relative
 - extrema::ExtremaParams, 35
- relative_margin
 - extrema::s_thresh_par, 53
- replace_time
 - extrema::ExtremaStats, 37
- rgb_to_blue
 - extrema::CPreprocess, 31
- rgb_to_green
 - extrema::CPreprocess, 31
- rgb_to_hue
 - extrema::CPreprocess, 31
- rgb_to_intensity
 - extrema::CPreprocess, 31
- rgb_to_intensity_half
 - extrema::CPreprocess, 31
- rgb_to_none
 - extrema::CPreprocess, 31
- rgb_to_red
 - extrema::CPreprocess, 31
- rgb_to_redblue
 - extrema::CPreprocess, 31
- rgb_to_saturation
 - extrema::CPreprocess, 31
- rid
 - extrema::Region, 43
- rle
 - extrema::RLERegion, 46
- RLE2Ellipse
 - extrema, 19
- row_pointer
 - utls::Ary, 25
- rows
 - utls::Ary, 25
- schur_sym
 - utls::Matrix2, 40
- set
 - utls::Ary, 25
- size
 - utls::Ary, 25
- size_type
 - utls::Ary, 25
- sortPixels.h, 73
- sqrt
 - utls::Matrix2, 40
- sqrt_i
 - utls::Matrix2, 40
- st
 - utls::t_frgb, 55
 - utls::t_rgb, 57
- std, 20
- suballoc.h, 74
- SuballocatorAddBlock
 - extrema, 19
- svd
 - utls::Matrix2, 40
- sxx
 - extrema::Region, 43
- sxy
 - extrema::Region, 43
- syy
 - extrema::Region, 43
- t_borderpixel
 - extrema, 15
- t_byte
 - utls, 21
- t_label
 - extrema, 15
- t_mregion
 - extrema, 15
- t_region
 - extrema, 15
- t_region_equiv
 - extrema, 15
- t_sortpixels
 - extrema, 15

- t_suballocator
 - extrema, 15
- t_thresh_def
 - extrema, 15
- t_thresh_par
 - extrema, 15
- tag
 - utls::Ary, 25
- thresh
 - extrema::s_thresh_def, 52
- threshold
 - extrema::Region, 43
- thresholds
 - extrema::s_region, 48
- TIME_STATS
 - extremaConfig.h, 63
- timeutls.h, 75
 - get_time, 75
- total_time
 - extrema::ExtremaStats, 37
- trace
 - utls::Matrix2, 40
- transpose
 - utls::Matrix2, 40
- transpose_i
 - utls::Matrix2, 40
- ub1
 - utls::Ary, 25
- ub2
 - utls::Ary, 25
- user_data
 - utls::Ary, 25
- utls, 21
 - BAry, 21
 - DAry, 21
 - FAry, 21
 - IAry, 21
 - LAry, 21
 - PAry, 21
 - t_byte, 21
- utls::Ary, 23
 - ~Ary, 25
 - Ary, 25
 - clear, 25
 - cols, 25
 - cons, 25
 - coord_type, 25
 - copy, 25
 - data, 25
 - deallocate, 25
 - detach, 25
 - el, 25
 - isin, 25
 - lb1, 25
 - lb2, 25
 - num_cols, 25
 - num_rows, 25
 - pointer, 25
 - ptr, 25
 - row_pointer, 25
 - rows, 25
 - set, 25
 - size, 25
 - size_type, 25
 - tag, 25
 - ub1, 25
 - ub2, 25
 - user_data, 25
 - value, 25
- utls::Matrix2, 38
 - ~Matrix2, 39
 - a, 40
 - chol, 39
 - det, 39
 - dump, 39
 - eye, 39
 - fnorm, 39
 - fnormalize, 39
 - inv, 39
 - inv_i, 39
 - Matrix2, 39
 - operator *, 39
 - operator+, 39
 - operator-, 39
 - operator/, 39
 - operator=, 39
 - operator[], 39
 - QR, 39
 - schur_sym, 40
 - sqrt, 40
 - sqrt_i, 40
 - svd, 40
 - trace, 40
 - transpose, 40
 - transpose_i, 40
- utls::t_fr gb, 55
 - arr, 55
 - b, 55
 - g, 55
 - r, 55
 - st, 55
- utls::t_r gb, 57
 - arr, 57
 - b, 57
 - g, 57
 - r, 57
 - st, 57

value
 utls::Ary, [25](#)

verbose
 extrema::ExtremaParams, [35](#)

width
 extrema::ExtremaImage, [34](#)

x
 extrema::t_ipoint, [56](#)

y
 extrema::t_ipoint, [56](#)