JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/calib3d/StereoSGBM.html)
* [No Frames](http://docs.google.com/StereoSGBM.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#tyjcwt)
* Detail:
* [Field](#2s8eyo1) |
* Constr |
* [Method](#35nkun2)

org.opencv.calib3d

## Class StereoSGBM

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.calib3d.StereoMatcher](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)
      * org.opencv.calib3d.StereoSGBM
* public class StereoSGBM  
  extends [StereoMatcher](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)

The class implements the modified H. Hirschmuller algorithm CITE: HH08 that differs from the original one as follows:

* + By default, the algorithm is single-pass, which means that you consider only 5 directions instead of 8. Set mode=StereoSGBM::MODE\_HH in createStereoSGBM to run the full variant of the algorithm but beware that it may consume a lot of memory.
  + The algorithm matches blocks, not individual pixels. Though, setting blockSize=1 reduces the blocks to single pixels.
  + Mutual information cost function is not implemented. Instead, a simpler Birchfield-Tomasi sub-pixel metric from CITE: BT98 is used. Though, the color images are supported as well.
  + Some pre- and post- processing steps from K. Konolige algorithm StereoBM are included, for example: pre-filtering (StereoBM::PREFILTER\_XSOBEL type) and post-filtering (uniqueness check, quadratic interpolation and speckle filtering).

**Note:**

* + (Python) An example illustrating the use of the StereoSGBM matching algorithm can be found at opencv\_source\_code/samples/python/stereo\_match.py

### Field SummaryFields

| Modifier and Type | Field and Description |
| --- | --- |
| static int | [**MODE\_HH**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#MODE_HH) |
| static int | [**MODE\_HH4**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#MODE_HH4) |
| static int | [**MODE\_SGBM**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#MODE_SGBM) |
| static int | [**MODE\_SGBM\_3WAY**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#MODE_SGBM_3WAY) |

### Fields inherited from class org.opencv.calib3d.[**StereoMatcher**](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)[DISP\_SCALE](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#DISP_SCALE), [DISP\_SHIFT](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#DISP_SHIFT)

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#__fromPtr__(long))(long addr) |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create())() Creates StereoSGBM object rectification algorithms can shift images, so this parameter needs to be adjusted accordingly. |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int))(int minDisparity) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int))(int minDisparity, int numDisparities) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize, int speckleRange) Creates StereoSGBM object |
| static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) | [**create**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#create(int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int))(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize, int speckleRange, int mode) Creates StereoSGBM object |
| int | [**getMode**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#getMode())() |
| int | [**getP1**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#getP1())() |
| int | [**getP2**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#getP2())() |
| int | [**getPreFilterCap**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#getPreFilterCap())() |
| int | [**getUniquenessRatio**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#getUniquenessRatio())() |
| void | [**setMode**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#setMode(int))(int mode) |
| void | [**setP1**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#setP1(int))(int P1) |
| void | [**setP2**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#setP2(int))(int P2) |
| void | [**setPreFilterCap**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#setPreFilterCap(int))(int preFilterCap) |
| void | [**setUniquenessRatio**](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html#setUniquenessRatio(int))(int uniquenessRatio) |

### Methods inherited from class org.opencv.calib3d.[**StereoMatcher**](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)[compute](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#compute(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.Mat)), [getBlockSize](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getBlockSize()), [getDisp12MaxDiff](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getDisp12MaxDiff()), [getMinDisparity](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getMinDisparity()), [getNumDisparities](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getNumDisparities()), [getSpeckleRange](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getSpeckleRange()), [getSpeckleWindowSize](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#getSpeckleWindowSize()), [setBlockSize](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setBlockSize(int)), [setDisp12MaxDiff](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setDisp12MaxDiff(int)), [setMinDisparity](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setMinDisparity(int)), [setNumDisparities](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setNumDisparities(int)), [setSpeckleRange](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setSpeckleRange(int)), [setSpeckleWindowSize](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html#setSpeckleWindowSize(int))

### Methods inherited from class org.opencv.core.[**Algorithm**](http://docs.google.com/org/opencv/core/Algorithm.html)[clear](http://docs.google.com/org/opencv/core/Algorithm.html#clear()), [empty](http://docs.google.com/org/opencv/core/Algorithm.html#empty()), [getDefaultName](http://docs.google.com/org/opencv/core/Algorithm.html#getDefaultName()), [getNativeObjAddr](http://docs.google.com/org/opencv/core/Algorithm.html#getNativeObjAddr()), [save](http://docs.google.com/org/opencv/core/Algorithm.html#save(java.lang.String))

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### MODE\_HH public static final int MODE\_HHSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.calib3d.StereoSGBM.MODE_HH)

#### MODE\_HH4 public static final int MODE\_HH4See Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.calib3d.StereoSGBM.MODE_HH4)

#### MODE\_SGBM public static final int MODE\_SGBMSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.calib3d.StereoSGBM.MODE_SGBM)

#### MODE\_SGBM\_3WAY public static final int MODE\_SGBM\_3WAYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.calib3d.StereoSGBM.MODE_SGBM_3WAY)

### Method Detail

#### \_\_fromPtr\_\_ public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) \_\_fromPtr\_\_(long addr)

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create() Creates StereoSGBM object rectification algorithms can shift images, so this parameter needs to be adjusted accordingly. zero. In the current implementation, this parameter must be divisible by 16. somewhere in the 3..11 range. the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value.Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly. zero. In the current implementation, this parameter must be divisible by 16. somewhere in the 3..11 range. the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16. somewhere in the 3..11 range. the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range. the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below. the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively). disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check. computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check.preFilterCap - Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function. value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check.preFilterCap - Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function.uniquenessRatio - Margin in percentage by which the best (minimum) computed cost function value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough. and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check.preFilterCap - Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function.uniquenessRatio - Margin in percentage by which the best (minimum) computed cost function value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough.speckleWindowSize - Maximum size of smooth disparity regions to consider their noise speckles and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range. filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize, int speckleRange) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check.preFilterCap - Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function.uniquenessRatio - Margin in percentage by which the best (minimum) computed cost function value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough.speckleWindowSize - Maximum size of smooth disparity regions to consider their noise speckles and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range.speckleRange - Maximum disparity variation within each connected component. If you do speckle filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough. algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### create public static [StereoSGBM](http://docs.google.com/org/opencv/calib3d/StereoSGBM.html) create(int minDisparity, int numDisparities, int blockSize, int P1, int P2, int disp12MaxDiff, int preFilterCap, int uniquenessRatio, int speckleWindowSize, int speckleRange, int mode) Creates StereoSGBM objectParameters:minDisparity - Minimum possible disparity value. Normally, it is zero but sometimes rectification algorithms can shift images, so this parameter needs to be adjusted accordingly.numDisparities - Maximum disparity minus minimum disparity. The value is always greater than zero. In the current implementation, this parameter must be divisible by 16.blockSize - Matched block size. It must be an odd number >=1 . Normally, it should be somewhere in the 3..11 range.P1 - The first parameter controlling the disparity smoothness. See below.P2 - The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1 . See stereo\_match.cpp sample where some reasonably good P1 and P2 values are shown (like 8\\*number\_of\_image\_channels\\*blockSize\\*blockSize and 32\\*number\_of\_image\_channels\\*blockSize\\*blockSize , respectively).disp12MaxDiff - Maximum allowed difference (in integer pixel units) in the left-right disparity check. Set it to a non-positive value to disable the check.preFilterCap - Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval. The result values are passed to the Birchfield-Tomasi pixel cost function.uniquenessRatio - Margin in percentage by which the best (minimum) computed cost function value should "win" the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough.speckleWindowSize - Maximum size of smooth disparity regions to consider their noise speckles and invalidate. Set it to 0 to disable speckle filtering. Otherwise, set it somewhere in the 50-200 range.speckleRange - Maximum disparity variation within each connected component. If you do speckle filtering, set the parameter to a positive value, it will be implicitly multiplied by 16. Normally, 1 or 2 is good enough.mode - Set it to StereoSGBM::MODE\_HH to run the full-scale two-pass dynamic programming algorithm. It will consume O(W\\*H\\*numDisparities) bytes, which is large for 640x480 stereo and huge for HD-size pictures. By default, it is set to false . The first constructor initializes StereoSGBM with all the default parameters. So, you only have to set StereoSGBM::numDisparities at minimum. The second constructor enables you to set each parameter to a custom value. Returns:automatically generated

#### getMode public int getMode()

#### getP1 public int getP1()

#### getP2 public int getP2()

#### getPreFilterCap public int getPreFilterCap()

#### getUniquenessRatio public int getUniquenessRatio()

#### setMode public void setMode(int mode)

#### setP1 public void setP1(int P1)

#### setP2 public void setP2(int P2)

#### setPreFilterCap public void setPreFilterCap(int preFilterCap)

#### setUniquenessRatio public void setUniquenessRatio(int uniquenessRatio)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/opencv/calib3d/StereoMatcher.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/calib3d/StereoSGBM.html)
* [No Frames](http://docs.google.com/StereoSGBM.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#tyjcwt)
* Detail:
* [Field](#2s8eyo1) |
* Constr |
* [Method](#35nkun2)

Generated on 2021-04-02 03:15:03 / OpenCV 3.4.14