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/features2d/SIFT.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/features2d/SimpleBlobDetector.html)
* [No Frames](http://docs.google.com/SimpleBlobDetector.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#1t3h5sf)

org.opencv.features2d

## Class SimpleBlobDetector

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.features2d.Feature2D](http://docs.google.com/org/opencv/features2d/Feature2D.html)
      * org.opencv.features2d.SimpleBlobDetector
* public class SimpleBlobDetector  
  extends [Feature2D](http://docs.google.com/org/opencv/features2d/Feature2D.html)

Class for extracting blobs from an image. : The class implements a simple algorithm for extracting blobs from an image: 1. Convert the source image to binary images by applying thresholding with several thresholds from minThreshold (inclusive) to maxThreshold (exclusive) with distance thresholdStep between neighboring thresholds. 2. Extract connected components from every binary image by findContours and calculate their centers. 3. Group centers from several binary images by their coordinates. Close centers form one group that corresponds to one blob, which is controlled by the minDistBetweenBlobs parameter. 4. From the groups, estimate final centers of blobs and their radiuses and return as locations and sizes of keypoints. This class performs several filtrations of returned blobs. You should set filterBy\\* to true/false to turn on/off corresponding filtration. Available filtrations:

* + **By color**. This filter compares the intensity of a binary image at the center of a blob to blobColor. If they differ, the blob is filtered out. Use blobColor = 0 to extract dark blobs and blobColor = 255 to extract light blobs.
  + **By area**. Extracted blobs have an area between minArea (inclusive) and maxArea (exclusive).
  + **By circularity**. Extracted blobs have circularity (\(\frac{4\*\pi\*Area}{perimeter \* perimeter}\)) between minCircularity (inclusive) and maxCircularity (exclusive).
  + **By ratio of the minimum inertia to maximum inertia**. Extracted blobs have this ratio between minInertiaRatio (inclusive) and maxInertiaRatio (exclusive).
  + **By convexity**. Extracted blobs have convexity (area / area of blob convex hull) between minConvexity (inclusive) and maxConvexity (exclusive).

Default values of parameters are tuned to extract dark circular blobs.

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [SimpleBlobDetector](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html#__fromPtr__(long))(long addr) |
| static [SimpleBlobDetector](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html) | [**create**](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html#create())() |
| java.lang.String | [**getDefaultName**](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html#getDefaultName())() Returns the algorithm string identifier. |

### Methods inherited from class org.opencv.features2d.[**Feature2D**](http://docs.google.com/org/opencv/features2d/Feature2D.html)[compute](http://docs.google.com/org/opencv/features2d/Feature2D.html#compute(java.util.List,%20java.util.List,%20java.util.List)), [compute](http://docs.google.com/org/opencv/features2d/Feature2D.html#compute(org.opencv.core.Mat,%20org.opencv.core.MatOfKeyPoint,%20org.opencv.core.Mat)), [defaultNorm](http://docs.google.com/org/opencv/features2d/Feature2D.html#defaultNorm()), [descriptorSize](http://docs.google.com/org/opencv/features2d/Feature2D.html#descriptorSize()), [descriptorType](http://docs.google.com/org/opencv/features2d/Feature2D.html#descriptorType()), [detect](http://docs.google.com/org/opencv/features2d/Feature2D.html#detect(java.util.List,%20java.util.List)), [detect](http://docs.google.com/org/opencv/features2d/Feature2D.html#detect(java.util.List,%20java.util.List,%20java.util.List)), [detect](http://docs.google.com/org/opencv/features2d/Feature2D.html#detect(org.opencv.core.Mat,%20org.opencv.core.MatOfKeyPoint)), [detect](http://docs.google.com/org/opencv/features2d/Feature2D.html#detect(org.opencv.core.Mat,%20org.opencv.core.MatOfKeyPoint,%20org.opencv.core.Mat)), [detectAndCompute](http://docs.google.com/org/opencv/features2d/Feature2D.html#detectAndCompute(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.MatOfKeyPoint,%20org.opencv.core.Mat)), [detectAndCompute](http://docs.google.com/org/opencv/features2d/Feature2D.html#detectAndCompute(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.MatOfKeyPoint,%20org.opencv.core.Mat,%20boolean)), [empty](http://docs.google.com/org/opencv/features2d/Feature2D.html#empty()), [read](http://docs.google.com/org/opencv/features2d/Feature2D.html#read(java.lang.String)), [write](http://docs.google.com/org/opencv/features2d/Feature2D.html#write(java.lang.String))

### 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()), [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

### Method Detail

#### \_\_fromPtr\_\_ public static [SimpleBlobDetector](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html) \_\_fromPtr\_\_(long addr)

#### create public static [SimpleBlobDetector](http://docs.google.com/org/opencv/features2d/SimpleBlobDetector.html) create()

#### getDefaultName public java.lang.String getDefaultName() **Description copied from class:**[**Algorithm**](http://docs.google.com/org/opencv/core/Algorithm.html#getDefaultName()) Returns the algorithm string identifier. This string is used as top level xml/yml node tag when the object is saved to a file or string.**Overrides:** [getDefaultName](http://docs.google.com/org/opencv/features2d/Feature2D.html#getDefaultName()) in class [Feature2D](http://docs.google.com/org/opencv/features2d/Feature2D.html) Returns:automatically generated

* [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/features2d/SIFT.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/features2d/SimpleBlobDetector.html)
* [No Frames](http://docs.google.com/SimpleBlobDetector.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#1t3h5sf)

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