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/AKAZE.html)
* [Next Class](http://docs.google.com/org/opencv/features2d/BOWImgDescriptorExtractor.html)
* [Frames](http://docs.google.com/index.html?org/opencv/features2d/BFMatcher.html)
* [No Frames](http://docs.google.com/BFMatcher.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#2et92p0) |
* [Constr](#tyjcwt) |
* [Method](#3dy6vkm)
* Detail:
* Field |
* [Constr](#17dp8vu) |
* [Method](#35nkun2)

org.opencv.features2d

## Class BFMatcher

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.features2d.DescriptorMatcher](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html)
      * org.opencv.features2d.BFMatcher
* public class BFMatcher  
  extends [DescriptorMatcher](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html)  
  Brute-force descriptor matcher. For each descriptor in the first set, this matcher finds the closest descriptor in the second set by trying each one. This descriptor matcher supports masking permissible matches of descriptor sets.

### Field Summary

### Fields inherited from class org.opencv.features2d.[**DescriptorMatcher**](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html)[BRUTEFORCE](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#BRUTEFORCE), [BRUTEFORCE\_HAMMING](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#BRUTEFORCE_HAMMING), [BRUTEFORCE\_HAMMINGLUT](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#BRUTEFORCE_HAMMINGLUT), [BRUTEFORCE\_L1](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#BRUTEFORCE_L1), [BRUTEFORCE\_SL2](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#BRUTEFORCE_SL2), [FLANNBASED](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#FLANNBASED)

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**BFMatcher**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#BFMatcher())() Brute-force matcher constructor (obsolete). |
| [**BFMatcher**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#BFMatcher(int))(int normType) Brute-force matcher constructor (obsolete). |
| [**BFMatcher**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#BFMatcher(int,%20boolean))(int normType, boolean crossCheck) Brute-force matcher constructor (obsolete). |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#__fromPtr__(long))(long addr) |
| static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) | [**create**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#create())() Brute-force matcher create method. |
| static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) | [**create**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#create(int))(int normType) Brute-force matcher create method. |
| static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) | [**create**](http://docs.google.com/org/opencv/features2d/BFMatcher.html#create(int,%20boolean))(int normType, boolean crossCheck) Brute-force matcher create method. |

### Methods inherited from class org.opencv.features2d.[**DescriptorMatcher**](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html)[add](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#add(java.util.List)), [clear](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#clear()), [clone](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#clone()), [clone](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#clone(boolean)), [create](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#create(java.lang.String)), [empty](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#empty()), [getTrainDescriptors](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#getTrainDescriptors()), [isMaskSupported](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#isMaskSupported()), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20java.util.List,%20int)), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20java.util.List,%20int,%20java.util.List)), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20java.util.List,%20int,%20java.util.List,%20boolean)), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20int)), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20int,%20org.opencv.core.Mat)), [knnMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#knnMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20int,%20org.opencv.core.Mat,%20boolean)), [match](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#match(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.MatOfDMatch)), [match](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#match(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.MatOfDMatch,%20org.opencv.core.Mat)), [match](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#match(org.opencv.core.Mat,%20org.opencv.core.MatOfDMatch)), [match](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#match(org.opencv.core.Mat,%20org.opencv.core.MatOfDMatch,%20java.util.List)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20java.util.List,%20float)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20java.util.List,%20float,%20java.util.List)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20java.util.List,%20float,%20java.util.List,%20boolean)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20float)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20float,%20org.opencv.core.Mat)), [radiusMatch](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#radiusMatch(org.opencv.core.Mat,%20org.opencv.core.Mat,%20java.util.List,%20float,%20org.opencv.core.Mat,%20boolean)), [read](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#read(java.lang.String)), [train](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#train()), [write](http://docs.google.com/org/opencv/features2d/DescriptorMatcher.html#write(java.lang.String))

### Methods inherited from class org.opencv.core.[**Algorithm**](http://docs.google.com/org/opencv/core/Algorithm.html)[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

### Constructor Detail

#### BFMatcher public BFMatcher() Brute-force matcher constructor (obsolete). Please use BFMatcher.create()

#### BFMatcher public BFMatcher(int normType) Brute-force matcher constructor (obsolete). Please use BFMatcher.create()Parameters:normType - automatically generated

#### BFMatcher public BFMatcher(int normType, boolean crossCheck) Brute-force matcher constructor (obsolete). Please use BFMatcher.create()Parameters:normType - automatically generatedcrossCheck - automatically generated

### Method Detail

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

#### create public static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) create() Brute-force matcher create method. preferable choices for SIFT and SURF descriptors, NORM\_HAMMING should be used with ORB, BRISK and BRIEF, NORM\_HAMMING2 should be used with ORB when WTA\_K==3 or 4 (see ORB::ORB constructor description). nearest neighbors for each query descriptor. If crossCheck==true, then the knnMatch() method with k=1 will only return pairs (i,j) such that for i-th query descriptor the j-th descriptor in the matcher's collection is the nearest and vice versa, i.e. the BFMatcher will only return consistent pairs. Such technique usually produces best results with minimal number of outliers when there are enough matches. This is alternative to the ratio test, used by D. Lowe in SIFT paper.Returns:automatically generated

#### create public static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) create(int normType) Brute-force matcher create method.Parameters:normType - One of NORM\_L1, NORM\_L2, NORM\_HAMMING, NORM\_HAMMING2. L1 and L2 norms are preferable choices for SIFT and SURF descriptors, NORM\_HAMMING should be used with ORB, BRISK and BRIEF, NORM\_HAMMING2 should be used with ORB when WTA\_K==3 or 4 (see ORB::ORB constructor description). nearest neighbors for each query descriptor. If crossCheck==true, then the knnMatch() method with k=1 will only return pairs (i,j) such that for i-th query descriptor the j-th descriptor in the matcher's collection is the nearest and vice versa, i.e. the BFMatcher will only return consistent pairs. Such technique usually produces best results with minimal number of outliers when there are enough matches. This is alternative to the ratio test, used by D. Lowe in SIFT paper. Returns:automatically generated

#### create public static [BFMatcher](http://docs.google.com/org/opencv/features2d/BFMatcher.html) create(int normType, boolean crossCheck) Brute-force matcher create method.Parameters:normType - One of NORM\_L1, NORM\_L2, NORM\_HAMMING, NORM\_HAMMING2. L1 and L2 norms are preferable choices for SIFT and SURF descriptors, NORM\_HAMMING should be used with ORB, BRISK and BRIEF, NORM\_HAMMING2 should be used with ORB when WTA\_K==3 or 4 (see ORB::ORB constructor description).crossCheck - If it is false, this is will be default BFMatcher behaviour when it finds the k nearest neighbors for each query descriptor. If crossCheck==true, then the knnMatch() method with k=1 will only return pairs (i,j) such that for i-th query descriptor the j-th descriptor in the matcher's collection is the nearest and vice versa, i.e. the BFMatcher will only return consistent pairs. Such technique usually produces best results with minimal number of outliers when there are enough matches. This is alternative to the ratio test, used by D. Lowe in SIFT paper. 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/AKAZE.html)
* [Next Class](http://docs.google.com/org/opencv/features2d/BOWImgDescriptorExtractor.html)
* [Frames](http://docs.google.com/index.html?org/opencv/features2d/BFMatcher.html)
* [No Frames](http://docs.google.com/BFMatcher.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#2et92p0) |
* [Constr](#tyjcwt) |
* [Method](#3dy6vkm)
* Detail:
* Field |
* [Constr](#17dp8vu) |
* [Method](#35nkun2)

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