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/video/BackgroundSubtractor.html)
* [Next Class](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html)
* [Frames](http://docs.google.com/index.html?org/opencv/video/BackgroundSubtractorKNN.html)
* [No Frames](http://docs.google.com/BackgroundSubtractorKNN.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#1t3h5sf)

org.opencv.video

## Class BackgroundSubtractorKNN

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.video.BackgroundSubtractor](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html)
      * org.opencv.video.BackgroundSubtractorKNN
* public class BackgroundSubtractorKNN  
  extends [BackgroundSubtractor](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html)  
  K-nearest neighbours - based Background/Foreground Segmentation Algorithm. The class implements the K-nearest neighbours background subtraction described in CITE: Zivkovic2006 . Very efficient if number of foreground pixels is low.

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [BackgroundSubtractorKNN](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#__fromPtr__(long))(long addr) |
| boolean | [**getDetectShadows**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getDetectShadows())() Returns the shadow detection flag If true, the algorithm detects shadows and marks them. |
| double | [**getDist2Threshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getDist2Threshold())() Returns the threshold on the squared distance between the pixel and the sample The threshold on the squared distance between the pixel and the sample to decide whether a pixel is close to a data sample. |
| int | [**getHistory**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getHistory())() Returns the number of last frames that affect the background model |
| int | [**getkNNSamples**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getkNNSamples())() Returns the number of neighbours, the k in the kNN. |
| int | [**getNSamples**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getNSamples())() Returns the number of data samples in the background model |
| double | [**getShadowThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getShadowThreshold())() Returns the shadow threshold A shadow is detected if pixel is a darker version of the background. |
| int | [**getShadowValue**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#getShadowValue())() Returns the shadow value Shadow value is the value used to mark shadows in the foreground mask. |
| void | [**setDetectShadows**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setDetectShadows(boolean))(boolean detectShadows) Enables or disables shadow detection |
| void | [**setDist2Threshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setDist2Threshold(double))(double \_dist2Threshold) Sets the threshold on the squared distance |
| void | [**setHistory**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setHistory(int))(int history) Sets the number of last frames that affect the background model |
| void | [**setkNNSamples**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setkNNSamples(int))(int \_nkNN) Sets the k in the kNN. |
| void | [**setNSamples**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setNSamples(int))(int \_nN) Sets the number of data samples in the background model. |
| void | [**setShadowThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setShadowThreshold(double))(double threshold) Sets the shadow threshold |
| void | [**setShadowValue**](http://docs.google.com/org/opencv/video/BackgroundSubtractorKNN.html#setShadowValue(int))(int value) Sets the shadow value |

### Methods inherited from class org.opencv.video.[**BackgroundSubtractor**](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html)[apply](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat)), [apply](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat,%20double)), [getBackgroundImage](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html#getBackgroundImage(org.opencv.core.Mat))

### 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

### Method Detail

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

#### getDetectShadows public boolean getDetectShadows() Returns the shadow detection flag If true, the algorithm detects shadows and marks them. See createBackgroundSubtractorKNN for details.Returns:automatically generated

#### getDist2Threshold public double getDist2Threshold() Returns the threshold on the squared distance between the pixel and the sample The threshold on the squared distance between the pixel and the sample to decide whether a pixel is close to a data sample.Returns:automatically generated

#### getHistory public int getHistory() Returns the number of last frames that affect the background modelReturns:automatically generated

#### getkNNSamples public int getkNNSamples() Returns the number of neighbours, the k in the kNN. K is the number of samples that need to be within dist2Threshold in order to decide that that pixel is matching the kNN background model.Returns:automatically generated

#### getNSamples public int getNSamples() Returns the number of data samples in the background modelReturns:automatically generated

#### getShadowThreshold public double getShadowThreshold() Returns the shadow threshold A shadow is detected if pixel is a darker version of the background. The shadow threshold (Tau in the paper) is a threshold defining how much darker the shadow can be. Tau= 0.5 means that if a pixel is more than twice darker then it is not shadow. See Prati, Mikic, Trivedi and Cucchiara, Detecting Moving Shadows...\*, IEEE PAMI,2003.Returns:automatically generated

#### getShadowValue public int getShadowValue() Returns the shadow value Shadow value is the value used to mark shadows in the foreground mask. Default value is 127. Value 0 in the mask always means background, 255 means foreground.Returns:automatically generated

#### setDetectShadows public void setDetectShadows(boolean detectShadows) Enables or disables shadow detectionParameters:detectShadows - automatically generated

#### setDist2Threshold public void setDist2Threshold(double \_dist2Threshold) Sets the threshold on the squared distanceParameters:\_dist2Threshold - automatically generated

#### setHistory public void setHistory(int history) Sets the number of last frames that affect the background modelParameters:history - automatically generated

#### setkNNSamples public void setkNNSamples(int \_nkNN) Sets the k in the kNN. How many nearest neighbours need to match.Parameters:\_nkNN - automatically generated

#### setNSamples public void setNSamples(int \_nN) Sets the number of data samples in the background model. The model needs to be reinitalized to reserve memory.Parameters:\_nN - automatically generated

#### setShadowThreshold public void setShadowThreshold(double threshold) Sets the shadow thresholdParameters:threshold - automatically generated

#### setShadowValue public void setShadowValue(int value) Sets the shadow valueParameters:value - 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/video/BackgroundSubtractor.html)
* [Next Class](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html)
* [Frames](http://docs.google.com/index.html?org/opencv/video/BackgroundSubtractorKNN.html)
* [No Frames](http://docs.google.com/BackgroundSubtractorKNN.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