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/BackgroundSubtractorKNN.html)
* [Next Class](http://docs.google.com/org/opencv/video/DenseOpticalFlow.html)
* [Frames](http://docs.google.com/index.html?org/opencv/video/BackgroundSubtractorMOG2.html)
* [No Frames](http://docs.google.com/BackgroundSubtractorMOG2.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 BackgroundSubtractorMOG2

* 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.BackgroundSubtractorMOG2
* public class BackgroundSubtractorMOG2  
  extends [BackgroundSubtractor](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html)  
  Gaussian Mixture-based Background/Foreground Segmentation Algorithm. The class implements the Gaussian mixture model background subtraction described in CITE: Zivkovic2004 and CITE: Zivkovic2006 .

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [BackgroundSubtractorMOG2](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#__fromPtr__(long))(long addr) |
| void | [**apply**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [Mat](http://docs.google.com/org/opencv/core/Mat.html) fgmask) Computes a foreground mask. |
| void | [**apply**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat,%20double))([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [Mat](http://docs.google.com/org/opencv/core/Mat.html) fgmask, double learningRate) Computes a foreground mask. |
| double | [**getBackgroundRatio**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getBackgroundRatio())() Returns the "background ratio" parameter of the algorithm If a foreground pixel keeps semi-constant value for about backgroundRatio\\*history frames, it's considered background and added to the model as a center of a new component. |
| double | [**getComplexityReductionThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getComplexityReductionThreshold())() Returns the complexity reduction threshold This parameter defines the number of samples needed to accept to prove the component exists. |
| boolean | [**getDetectShadows**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getDetectShadows())() Returns the shadow detection flag If true, the algorithm detects shadows and marks them. |
| int | [**getHistory**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getHistory())() Returns the number of last frames that affect the background model |
| int | [**getNMixtures**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getNMixtures())() Returns the number of gaussian components in the background model |
| double | [**getShadowThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.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/BackgroundSubtractorMOG2.html#getShadowValue())() Returns the shadow value Shadow value is the value used to mark shadows in the foreground mask. |
| double | [**getVarInit**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getVarInit())() Returns the initial variance of each gaussian component |
| double | [**getVarMax**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getVarMax())() |
| double | [**getVarMin**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getVarMin())() |
| double | [**getVarThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getVarThreshold())() Returns the variance threshold for the pixel-model match The main threshold on the squared Mahalanobis distance to decide if the sample is well described by the background model or not. |
| double | [**getVarThresholdGen**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#getVarThresholdGen())() Returns the variance threshold for the pixel-model match used for new mixture component generation Threshold for the squared Mahalanobis distance that helps decide when a sample is close to the existing components (corresponds to Tg in the paper). |
| void | [**setBackgroundRatio**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setBackgroundRatio(double))(double ratio) Sets the "background ratio" parameter of the algorithm |
| void | [**setComplexityReductionThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setComplexityReductionThreshold(double))(double ct) Sets the complexity reduction threshold |
| void | [**setDetectShadows**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setDetectShadows(boolean))(boolean detectShadows) Enables or disables shadow detection |
| void | [**setHistory**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setHistory(int))(int history) Sets the number of last frames that affect the background model |
| void | [**setNMixtures**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setNMixtures(int))(int nmixtures) Sets the number of gaussian components in the background model. |
| void | [**setShadowThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setShadowThreshold(double))(double threshold) Sets the shadow threshold |
| void | [**setShadowValue**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setShadowValue(int))(int value) Sets the shadow value |
| void | [**setVarInit**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setVarInit(double))(double varInit) Sets the initial variance of each gaussian component |
| void | [**setVarMax**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setVarMax(double))(double varMax) |
| void | [**setVarMin**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setVarMin(double))(double varMin) |
| void | [**setVarThreshold**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setVarThreshold(double))(double varThreshold) Sets the variance threshold for the pixel-model match |
| void | [**setVarThresholdGen**](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html#setVarThresholdGen(double))(double varThresholdGen) Sets the variance threshold for the pixel-model match used for new mixture component generation |

### Methods inherited from class org.opencv.video.[**BackgroundSubtractor**](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html)[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 [BackgroundSubtractorMOG2](http://docs.google.com/org/opencv/video/BackgroundSubtractorMOG2.html) \_\_fromPtr\_\_(long addr)

#### apply public void apply([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [Mat](http://docs.google.com/org/opencv/core/Mat.html) fgmask) Computes a foreground mask.**Overrides:** [apply](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat)) in class [BackgroundSubtractor](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html) Parameters:image - Next video frame. Floating point frame will be used without scaling and should be in range \([0,255]\).fgmask - The output foreground mask as an 8-bit binary image. learnt. Negative parameter value makes the algorithm to use some automatically chosen learning rate. 0 means that the background model is not updated at all, 1 means that the background model is completely reinitialized from the last frame.

#### apply public void apply([Mat](http://docs.google.com/org/opencv/core/Mat.html) image, [Mat](http://docs.google.com/org/opencv/core/Mat.html) fgmask, double learningRate) Computes a foreground mask.**Overrides:** [apply](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html#apply(org.opencv.core.Mat,%20org.opencv.core.Mat,%20double)) in class [BackgroundSubtractor](http://docs.google.com/org/opencv/video/BackgroundSubtractor.html) Parameters:image - Next video frame. Floating point frame will be used without scaling and should be in range \([0,255]\).fgmask - The output foreground mask as an 8-bit binary image.learningRate - The value between 0 and 1 that indicates how fast the background model is learnt. Negative parameter value makes the algorithm to use some automatically chosen learning rate. 0 means that the background model is not updated at all, 1 means that the background model is completely reinitialized from the last frame.

#### getBackgroundRatio public double getBackgroundRatio() Returns the "background ratio" parameter of the algorithm If a foreground pixel keeps semi-constant value for about backgroundRatio\\*history frames, it's considered background and added to the model as a center of a new component. It corresponds to TB parameter in the paper.Returns:automatically generated

#### getComplexityReductionThreshold public double getComplexityReductionThreshold() Returns the complexity reduction threshold This parameter defines the number of samples needed to accept to prove the component exists. CT=0.05 is a default value for all the samples. By setting CT=0 you get an algorithm very similar to the standard Stauffer&Grimson algorithm.Returns:automatically generated

#### getDetectShadows public boolean getDetectShadows() Returns the shadow detection flag If true, the algorithm detects shadows and marks them. See createBackgroundSubtractorMOG2 for details.Returns:automatically generated

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

#### getNMixtures public int getNMixtures() Returns the number of gaussian components 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

#### getVarInit public double getVarInit() Returns the initial variance of each gaussian componentReturns:automatically generated

#### getVarMax public double getVarMax()

#### getVarMin public double getVarMin()

#### getVarThreshold public double getVarThreshold() Returns the variance threshold for the pixel-model match The main threshold on the squared Mahalanobis distance to decide if the sample is well described by the background model or not. Related to Cthr from the paper.Returns:automatically generated

#### getVarThresholdGen public double getVarThresholdGen() Returns the variance threshold for the pixel-model match used for new mixture component generation Threshold for the squared Mahalanobis distance that helps decide when a sample is close to the existing components (corresponds to Tg in the paper). If a pixel is not close to any component, it is considered foreground or added as a new component. 3 sigma => Tg=3\\*3=9 is default. A smaller Tg value generates more components. A higher Tg value may result in a small number of components but they can grow too large.Returns:automatically generated

#### setBackgroundRatio public void setBackgroundRatio(double ratio) Sets the "background ratio" parameter of the algorithmParameters:ratio - automatically generated

#### setComplexityReductionThreshold public void setComplexityReductionThreshold(double ct) Sets the complexity reduction thresholdParameters:ct - automatically generated

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

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

#### setNMixtures public void setNMixtures(int nmixtures) Sets the number of gaussian components in the background model. The model needs to be reinitalized to reserve memory.Parameters:nmixtures - 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

#### setVarInit public void setVarInit(double varInit) Sets the initial variance of each gaussian componentParameters:varInit - automatically generated

#### setVarMax public void setVarMax(double varMax)

#### setVarMin public void setVarMin(double varMin)

#### setVarThreshold public void setVarThreshold(double varThreshold) Sets the variance threshold for the pixel-model matchParameters:varThreshold - automatically generated

#### setVarThresholdGen public void setVarThresholdGen(double varThresholdGen) Sets the variance threshold for the pixel-model match used for new mixture component generationParameters:varThresholdGen - 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/BackgroundSubtractorKNN.html)
* [Next Class](http://docs.google.com/org/opencv/video/DenseOpticalFlow.html)
* [Frames](http://docs.google.com/index.html?org/opencv/video/BackgroundSubtractorMOG2.html)
* [No Frames](http://docs.google.com/BackgroundSubtractorMOG2.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