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/photo/AlignExposures.html)
* [Next Class](http://docs.google.com/org/opencv/photo/CalibrateCRF.html)
* [Frames](http://docs.google.com/index.html?org/opencv/photo/AlignMTB.html)
* [No Frames](http://docs.google.com/AlignMTB.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#3dy6vkm)

org.opencv.photo

## Class AlignMTB

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.photo.AlignExposures](http://docs.google.com/org/opencv/photo/AlignExposures.html)
      * org.opencv.photo.AlignMTB
* public class AlignMTB  
  extends [AlignExposures](http://docs.google.com/org/opencv/photo/AlignExposures.html)  
  This algorithm converts images to median threshold bitmaps (1 for pixels brighter than median luminance and 0 otherwise) and than aligns the resulting bitmaps using bit operations. It is invariant to exposure, so exposure values and camera response are not necessary. In this implementation new image regions are filled with zeros. For more information see CITE: GW03 .

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [AlignMTB](http://docs.google.com/org/opencv/photo/AlignMTB.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/photo/AlignMTB.html#__fromPtr__(long))(long addr) |
| [Point](http://docs.google.com/org/opencv/core/Point.html) | [**calculateShift**](http://docs.google.com/org/opencv/photo/AlignMTB.html#calculateShift(org.opencv.core.Mat,%20org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) img0, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img1) Calculates shift between two images, i. |
| void | [**computeBitmaps**](http://docs.google.com/org/opencv/photo/AlignMTB.html#computeBitmaps(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [Mat](http://docs.google.com/org/opencv/core/Mat.html) tb, [Mat](http://docs.google.com/org/opencv/core/Mat.html) eb) Computes median threshold and exclude bitmaps of given image. |
| boolean | [**getCut**](http://docs.google.com/org/opencv/photo/AlignMTB.html#getCut())() |
| int | [**getExcludeRange**](http://docs.google.com/org/opencv/photo/AlignMTB.html#getExcludeRange())() |
| int | [**getMaxBits**](http://docs.google.com/org/opencv/photo/AlignMTB.html#getMaxBits())() |
| void | [**process**](http://docs.google.com/org/opencv/photo/AlignMTB.html#process(java.util.List,%20java.util.List))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> src, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> dst) Short version of process, that doesn't take extra arguments. |
| void | [**process**](http://docs.google.com/org/opencv/photo/AlignMTB.html#process(java.util.List,%20java.util.List,%20org.opencv.core.Mat,%20org.opencv.core.Mat))(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> src, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> dst, [Mat](http://docs.google.com/org/opencv/core/Mat.html) times, [Mat](http://docs.google.com/org/opencv/core/Mat.html) response) Aligns images |
| void | [**setCut**](http://docs.google.com/org/opencv/photo/AlignMTB.html#setCut(boolean))(boolean value) |
| void | [**setExcludeRange**](http://docs.google.com/org/opencv/photo/AlignMTB.html#setExcludeRange(int))(int exclude\_range) |
| void | [**setMaxBits**](http://docs.google.com/org/opencv/photo/AlignMTB.html#setMaxBits(int))(int max\_bits) |
| void | [**shiftMat**](http://docs.google.com/org/opencv/photo/AlignMTB.html#shiftMat(org.opencv.core.Mat,%20org.opencv.core.Mat,%20org.opencv.core.Point))([Mat](http://docs.google.com/org/opencv/core/Mat.html) src, [Mat](http://docs.google.com/org/opencv/core/Mat.html) dst, [Point](http://docs.google.com/org/opencv/core/Point.html) shift) Helper function, that shift Mat filling new regions with zeros. |

### 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 [AlignMTB](http://docs.google.com/org/opencv/photo/AlignMTB.html) \_\_fromPtr\_\_(long addr)

#### calculateShift public [Point](http://docs.google.com/org/opencv/core/Point.html) calculateShift([Mat](http://docs.google.com/org/opencv/core/Mat.html) img0, [Mat](http://docs.google.com/org/opencv/core/Mat.html) img1) Calculates shift between two images, i. e. how to shift the second image to correspond it with the first.Parameters:img0 - first imageimg1 - second image Returns:automatically generated

#### computeBitmaps public void computeBitmaps([Mat](http://docs.google.com/org/opencv/core/Mat.html) img, [Mat](http://docs.google.com/org/opencv/core/Mat.html) tb, [Mat](http://docs.google.com/org/opencv/core/Mat.html) eb) Computes median threshold and exclude bitmaps of given image.Parameters:img - input imagetb - median threshold bitmapeb - exclude bitmap

#### getCut public boolean getCut()

#### getExcludeRange public int getExcludeRange()

#### getMaxBits public int getMaxBits()

#### process public void process(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> src, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> dst) Short version of process, that doesn't take extra arguments.Parameters:src - vector of input imagesdst - vector of aligned images

#### process public void process(java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> src, java.util.List<[Mat](http://docs.google.com/org/opencv/core/Mat.html)> dst, [Mat](http://docs.google.com/org/opencv/core/Mat.html) times, [Mat](http://docs.google.com/org/opencv/core/Mat.html) response) **Description copied from class:**[**AlignExposures**](http://docs.google.com/org/opencv/photo/AlignExposures.html#process(java.util.List,%20java.util.List,%20org.opencv.core.Mat,%20org.opencv.core.Mat)) Aligns images**Overrides:** [process](http://docs.google.com/org/opencv/photo/AlignExposures.html#process(java.util.List,%20java.util.List,%20org.opencv.core.Mat,%20org.opencv.core.Mat)) in class [AlignExposures](http://docs.google.com/org/opencv/photo/AlignExposures.html) Parameters:src - vector of input imagesdst - vector of aligned imagestimes - vector of exposure time values for each imageresponse - 256x1 matrix with inverse camera response function for each pixel value, it should have the same number of channels as images.

#### setCut public void setCut(boolean value)

#### setExcludeRange public void setExcludeRange(int exclude\_range)

#### setMaxBits public void setMaxBits(int max\_bits)

#### shiftMat public void shiftMat([Mat](http://docs.google.com/org/opencv/core/Mat.html) src, [Mat](http://docs.google.com/org/opencv/core/Mat.html) dst, [Point](http://docs.google.com/org/opencv/core/Point.html) shift) Helper function, that shift Mat filling new regions with zeros.Parameters:src - input imagedst - result imageshift - shift value

* [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/photo/AlignExposures.html)
* [Next Class](http://docs.google.com/org/opencv/photo/CalibrateCRF.html)
* [Frames](http://docs.google.com/index.html?org/opencv/photo/AlignMTB.html)
* [No Frames](http://docs.google.com/AlignMTB.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#3dy6vkm)

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