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/imgproc/Moments.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/imgproc/Subdiv2D.html)
* [No Frames](http://docs.google.com/Subdiv2D.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#4i7ojhp) |
* [Method](#3whwml4)

org.opencv.imgproc

## Class Subdiv2D

* java.lang.Object
  + org.opencv.imgproc.Subdiv2D
* public class Subdiv2D  
  extends java.lang.Object

### Field SummaryFields

| Modifier and Type | Field and Description |
| --- | --- |
| static int | [**NEXT\_AROUND\_DST**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#NEXT_AROUND_DST) |
| static int | [**NEXT\_AROUND\_LEFT**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#NEXT_AROUND_LEFT) |
| static int | [**NEXT\_AROUND\_ORG**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#NEXT_AROUND_ORG) |
| static int | [**NEXT\_AROUND\_RIGHT**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#NEXT_AROUND_RIGHT) |
| static int | [**PREV\_AROUND\_DST**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PREV_AROUND_DST) |
| static int | [**PREV\_AROUND\_LEFT**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PREV_AROUND_LEFT) |
| static int | [**PREV\_AROUND\_ORG**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PREV_AROUND_ORG) |
| static int | [**PREV\_AROUND\_RIGHT**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PREV_AROUND_RIGHT) |
| static int | [**PTLOC\_ERROR**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PTLOC_ERROR) |
| static int | [**PTLOC\_INSIDE**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PTLOC_INSIDE) |
| static int | [**PTLOC\_ON\_EDGE**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PTLOC_ON_EDGE) |
| static int | [**PTLOC\_OUTSIDE\_RECT**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PTLOC_OUTSIDE_RECT) |
| static int | [**PTLOC\_VERTEX**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#PTLOC_VERTEX) |

### Constructor SummaryConstructors

| Constructor and Description |
| --- |
| [**Subdiv2D**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#Subdiv2D())() creates an empty Subdiv2D object. |
| [**Subdiv2D**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#Subdiv2D(org.opencv.core.Rect))([Rect](http://docs.google.com/org/opencv/core/Rect.html) rect) |

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [Subdiv2D](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#__fromPtr__(long))(long addr) |
| int | [**edgeDst**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#edgeDst(int))(int edge) Returns the edge destination. |
| int | [**edgeDst**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#edgeDst(int,%20org.opencv.core.Point))(int edge, [Point](http://docs.google.com/org/opencv/core/Point.html) dstpt) Returns the edge destination. |
| int | [**edgeOrg**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#edgeOrg(int))(int edge) Returns the edge origin. |
| int | [**edgeOrg**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#edgeOrg(int,%20org.opencv.core.Point))(int edge, [Point](http://docs.google.com/org/opencv/core/Point.html) orgpt) Returns the edge origin. |
| int | [**findNearest**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#findNearest(org.opencv.core.Point))([Point](http://docs.google.com/org/opencv/core/Point.html) pt) Finds the subdivision vertex closest to the given point. |
| int | [**findNearest**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#findNearest(org.opencv.core.Point,%20org.opencv.core.Point))([Point](http://docs.google.com/org/opencv/core/Point.html) pt, [Point](http://docs.google.com/org/opencv/core/Point.html) nearestPt) Finds the subdivision vertex closest to the given point. |
| int | [**getEdge**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getEdge(int,%20int))(int edge, int nextEdgeType) Returns one of the edges related to the given edge. |
| void | [**getEdgeList**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getEdgeList(org.opencv.core.MatOfFloat4))([MatOfFloat4](http://docs.google.com/org/opencv/core/MatOfFloat4.html) edgeList) Returns a list of all edges. |
| void | [**getLeadingEdgeList**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getLeadingEdgeList(org.opencv.core.MatOfInt))([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) leadingEdgeList) Returns a list of the leading edge ID connected to each triangle. |
| long | [**getNativeObjAddr**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getNativeObjAddr())() |
| void | [**getTriangleList**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getTriangleList(org.opencv.core.MatOfFloat6))([MatOfFloat6](http://docs.google.com/org/opencv/core/MatOfFloat6.html) triangleList) Returns a list of all triangles. |
| [Point](http://docs.google.com/org/opencv/core/Point.html) | [**getVertex**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getVertex(int))(int vertex) Returns vertex location from vertex ID. |
| [Point](http://docs.google.com/org/opencv/core/Point.html) | [**getVertex**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getVertex(int,%20int%5B%5D))(int vertex, int[] firstEdge) Returns vertex location from vertex ID. |
| void | [**getVoronoiFacetList**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#getVoronoiFacetList(org.opencv.core.MatOfInt,%20java.util.List,%20org.opencv.core.MatOfPoint2f))([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) idx, java.util.List<[MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html)> facetList, [MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html) facetCenters) Returns a list of all Voronoi facets. |
| void | [**initDelaunay**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#initDelaunay(org.opencv.core.Rect))([Rect](http://docs.google.com/org/opencv/core/Rect.html) rect) Creates a new empty Delaunay subdivision |
| void | [**insert**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#insert(org.opencv.core.MatOfPoint2f))([MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html) ptvec) Insert multiple points into a Delaunay triangulation. |
| int | [**insert**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#insert(org.opencv.core.Point))([Point](http://docs.google.com/org/opencv/core/Point.html) pt) Insert a single point into a Delaunay triangulation. |
| int | [**locate**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#locate(org.opencv.core.Point,%20int%5B%5D,%20int%5B%5D))([Point](http://docs.google.com/org/opencv/core/Point.html) pt, int[] edge, int[] vertex) Returns the location of a point within a Delaunay triangulation. |
| int | [**nextEdge**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#nextEdge(int))(int edge) Returns next edge around the edge origin. |
| int | [**rotateEdge**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#rotateEdge(int,%20int))(int edge, int rotate) Returns another edge of the same quad-edge. |
| int | [**symEdge**](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html#symEdge(int))(int edge) |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### NEXT\_AROUND\_DST public static final int NEXT\_AROUND\_DSTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.NEXT_AROUND_DST)

#### NEXT\_AROUND\_LEFT public static final int NEXT\_AROUND\_LEFTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.NEXT_AROUND_LEFT)

#### NEXT\_AROUND\_ORG public static final int NEXT\_AROUND\_ORGSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.NEXT_AROUND_ORG)

#### NEXT\_AROUND\_RIGHT public static final int NEXT\_AROUND\_RIGHTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.NEXT_AROUND_RIGHT)

#### PREV\_AROUND\_DST public static final int PREV\_AROUND\_DSTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PREV_AROUND_DST)

#### PREV\_AROUND\_LEFT public static final int PREV\_AROUND\_LEFTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PREV_AROUND_LEFT)

#### PREV\_AROUND\_ORG public static final int PREV\_AROUND\_ORGSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PREV_AROUND_ORG)

#### PREV\_AROUND\_RIGHT public static final int PREV\_AROUND\_RIGHTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PREV_AROUND_RIGHT)

#### PTLOC\_ERROR public static final int PTLOC\_ERRORSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PTLOC_ERROR)

#### PTLOC\_INSIDE public static final int PTLOC\_INSIDESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PTLOC_INSIDE)

#### PTLOC\_ON\_EDGE public static final int PTLOC\_ON\_EDGESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PTLOC_ON_EDGE)

#### PTLOC\_OUTSIDE\_RECT public static final int PTLOC\_OUTSIDE\_RECTSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PTLOC_OUTSIDE_RECT)

#### PTLOC\_VERTEX public static final int PTLOC\_VERTEXSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.imgproc.Subdiv2D.PTLOC_VERTEX)

### Constructor Detail

#### Subdiv2D public Subdiv2D() creates an empty Subdiv2D object. To create a new empty Delaunay subdivision you need to use the #initDelaunay function.

#### Subdiv2D public Subdiv2D([Rect](http://docs.google.com/org/opencv/core/Rect.html) rect)Parameters:rect - Rectangle that includes all of the 2D points that are to be added to the subdivision. The function creates an empty Delaunay subdivision where 2D points can be added using the function insert() . All of the points to be added must be within the specified rectangle, otherwise a runtime error is raised.

### Method Detail

#### \_\_fromPtr\_\_ public static [Subdiv2D](http://docs.google.com/org/opencv/imgproc/Subdiv2D.html) \_\_fromPtr\_\_(long addr)

#### edgeDst public int edgeDst(int edge) Returns the edge destination.Parameters:edge - Subdivision edge ID. Returns:vertex ID.

#### edgeDst public int edgeDst(int edge, [Point](http://docs.google.com/org/opencv/core/Point.html) dstpt) Returns the edge destination.Parameters:edge - Subdivision edge ID.dstpt - Output vertex location. Returns:vertex ID.

#### edgeOrg public int edgeOrg(int edge) Returns the edge origin.Parameters:edge - Subdivision edge ID. Returns:vertex ID.

#### edgeOrg public int edgeOrg(int edge, [Point](http://docs.google.com/org/opencv/core/Point.html) orgpt) Returns the edge origin.Parameters:edge - Subdivision edge ID.orgpt - Output vertex location. Returns:vertex ID.

#### findNearest public int findNearest([Point](http://docs.google.com/org/opencv/core/Point.html) pt) Finds the subdivision vertex closest to the given point.Parameters:pt - Input point. The function is another function that locates the input point within the subdivision. It finds the subdivision vertex that is the closest to the input point. It is not necessarily one of vertices of the facet containing the input point, though the facet (located using locate() ) is used as a starting point. Returns:vertex ID.

#### findNearest public int findNearest([Point](http://docs.google.com/org/opencv/core/Point.html) pt, [Point](http://docs.google.com/org/opencv/core/Point.html) nearestPt) Finds the subdivision vertex closest to the given point.Parameters:pt - Input point.nearestPt - Output subdivision vertex point. The function is another function that locates the input point within the subdivision. It finds the subdivision vertex that is the closest to the input point. It is not necessarily one of vertices of the facet containing the input point, though the facet (located using locate() ) is used as a starting point. Returns:vertex ID.

#### getEdge public int getEdge(int edge, int nextEdgeType) Returns one of the edges related to the given edge.Parameters:edge - Subdivision edge ID.nextEdgeType - Parameter specifying which of the related edges to return. The following values are possible:

* + - * NEXT\_AROUND\_ORG next around the edge origin ( eOnext on the picture below if e is the input edge)
      * NEXT\_AROUND\_DST next around the edge vertex ( eDnext )
      * PREV\_AROUND\_ORG previous around the edge origin (reversed eRnext )
      * PREV\_AROUND\_DST previous around the edge destination (reversed eLnext )
      * NEXT\_AROUND\_LEFT next around the left facet ( eLnext )
      * NEXT\_AROUND\_RIGHT next around the right facet ( eRnext )
      * PREV\_AROUND\_LEFT previous around the left facet (reversed eOnext )
      * PREV\_AROUND\_RIGHT previous around the right facet (reversed eDnext )

![sample output](pics/quadedge.png) Returns:edge ID related to the input edge.

#### getEdgeList public void getEdgeList([MatOfFloat4](http://docs.google.com/org/opencv/core/MatOfFloat4.html) edgeList) Returns a list of all edges.Parameters:edgeList - Output vector. The function gives each edge as a 4 numbers vector, where each two are one of the edge vertices. i.e. org\_x = v[0], org\_y = v[1], dst\_x = v[2], dst\_y = v[3].

#### getLeadingEdgeList public void getLeadingEdgeList([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) leadingEdgeList) Returns a list of the leading edge ID connected to each triangle.Parameters:leadingEdgeList - Output vector. The function gives one edge ID for each triangle.

#### getNativeObjAddr public long getNativeObjAddr()

#### getTriangleList public void getTriangleList([MatOfFloat6](http://docs.google.com/org/opencv/core/MatOfFloat6.html) triangleList) Returns a list of all triangles.Parameters:triangleList - Output vector. The function gives each triangle as a 6 numbers vector, where each two are one of the triangle vertices. i.e. p1\_x = v[0], p1\_y = v[1], p2\_x = v[2], p2\_y = v[3], p3\_x = v[4], p3\_y = v[5].

#### getVertex public [Point](http://docs.google.com/org/opencv/core/Point.html) getVertex(int vertex) Returns vertex location from vertex ID.Parameters:vertex - vertex ID. Returns:vertex (x,y)

#### getVertex public [Point](http://docs.google.com/org/opencv/core/Point.html) getVertex(int vertex, int[] firstEdge) Returns vertex location from vertex ID.Parameters:vertex - vertex ID.firstEdge - Optional. The first edge ID which is connected to the vertex. Returns:vertex (x,y)

#### getVoronoiFacetList public void getVoronoiFacetList([MatOfInt](http://docs.google.com/org/opencv/core/MatOfInt.html) idx, java.util.List<[MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html)> facetList, [MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html) facetCenters) Returns a list of all Voronoi facets.Parameters:idx - Vector of vertices IDs to consider. For all vertices you can pass empty vector.facetList - Output vector of the Voronoi facets.facetCenters - Output vector of the Voronoi facets center points.

#### initDelaunay public void initDelaunay([Rect](http://docs.google.com/org/opencv/core/Rect.html) rect) Creates a new empty Delaunay subdivisionParameters:rect - Rectangle that includes all of the 2D points that are to be added to the subdivision.

#### insert public void insert([MatOfPoint2f](http://docs.google.com/org/opencv/core/MatOfPoint2f.html) ptvec) Insert multiple points into a Delaunay triangulation.Parameters:ptvec - Points to insert. The function inserts a vector of points into a subdivision and modifies the subdivision topology appropriately.

#### insert public int insert([Point](http://docs.google.com/org/opencv/core/Point.html) pt) Insert a single point into a Delaunay triangulation.Parameters:pt - Point to insert. The function inserts a single point into a subdivision and modifies the subdivision topology appropriately. If a point with the same coordinates exists already, no new point is added. Returns:the ID of the point. **Note:** If the point is outside of the triangulation specified rect a runtime error is raised.

#### locate public int locate([Point](http://docs.google.com/org/opencv/core/Point.html) pt, int[] edge, int[] vertex) Returns the location of a point within a Delaunay triangulation.Parameters:pt - Point to locate.edge - Output edge that the point belongs to or is located to the right of it.vertex - Optional output vertex the input point coincides with. The function locates the input point within the subdivision and gives one of the triangle edges or vertices. Returns:an integer which specify one of the following five cases for point location:

* + - * The point falls into some facet. The function returns #PTLOC\_INSIDE and edge will contain one of edges of the facet.
      * The point falls onto the edge. The function returns #PTLOC\_ON\_EDGE and edge will contain this edge.
      * The point coincides with one of the subdivision vertices. The function returns #PTLOC\_VERTEX and vertex will contain a pointer to the vertex.
      * The point is outside the subdivision reference rectangle. The function returns #PTLOC\_OUTSIDE\_RECT and no pointers are filled.
      * One of input arguments is invalid. A runtime error is raised or, if silent or "parent" error processing mode is selected, #PTLOC\_ERROR is returned.

#### nextEdge public int nextEdge(int edge) Returns next edge around the edge origin.Parameters:edge - Subdivision edge ID. Returns:an integer which is next edge ID around the edge origin: eOnext on the picture above if e is the input edge).

#### rotateEdge public int rotateEdge(int edge, int rotate) Returns another edge of the same quad-edge.Parameters:edge - Subdivision edge ID.rotate - Parameter specifying which of the edges of the same quad-edge as the input one to return. The following values are possible:

* + - * 0 - the input edge ( e on the picture below if e is the input edge)
      * 1 - the rotated edge ( eRot )
      * 2 - the reversed edge (reversed e (in green))
      * 3 - the reversed rotated edge (reversed eRot (in green))

Returns:one of the edges ID of the same quad-edge as the input edge.

#### symEdge public int symEdge(int edge)

* [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/imgproc/Moments.html)
* Next Class
* [Frames](http://docs.google.com/index.html?org/opencv/imgproc/Subdiv2D.html)
* [No Frames](http://docs.google.com/Subdiv2D.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* [Constr](#2et92p0) |
* [Method](#tyjcwt)
* Detail:
* [Field](#1t3h5sf) |
* [Constr](#4i7ojhp) |
* [Method](#3whwml4)

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