

MComp Coverage

Generated by Doxygen 1.9.4

1 Namespace Index	1
1.1 Namespace List	1
2 File Index	3
2.1 File List	3
3 Namespace Documentation	5
3.1 coverage_doc Namespace Reference	5
3.1.1 Function Documentation	5
3.1.1.1 bounding_box()	5
3.1.1.2 close_shape()	6
3.1.1.3 contained_x()	6
3.1.1.4 contained_y()	6
3.1.1.5 gradient()	7
3.1.1.6 graph()	7
3.1.1.7 inner_outer()	7
3.1.1.8 inter_points()	8
3.1.1.9 main()	8
3.1.1.10 midpoint()	8
3.1.1.11 quantise()	8
3.1.1.12 remove_inter()	9
3.1.1.13 to_xy()	9
4 File Documentation	11
4.1 coverage_doc.py File Reference	11
Index	13

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

coverage_doc	5
--	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

coverage_doc.py	11
---------------------------------	----

Chapter 3

Namespace Documentation

3.1 coverage_doc Namespace Reference

Functions

- def [midpoint](#) (p1, p2)
- def [inter_points](#) (p1, p2, num)
- def [gradient](#) (p1, p2)
- def [remove_inter](#) (points)
- def [contained_y](#) (bounds, y)
Checks if a point is contained within a bounding box's y axis.
- def [contained_x](#) (bounds, x)
Checks if a point is contained within the bounding box's x axis.
- def [graph](#) (points, height, width, overlap)
Generates the graph required for TSP from the generated from Quantise.
- def [bounding_box](#) (shape)
Creates a rectangular box which wholly contains the given shape.
- def [quantise](#) (shape, height, width, overlap, nogos)
Quantises a shape within a bounding box.
- def [to_xy](#) (perimeter, nogos)
Converts the perimeters and nogo zones from GPS to UTM.
- def [inner_outer](#) (xy_per, xy_nogos, width)
Generate inner boundary for perimeter and outer boundary(s) for nogo zone(s)
- def [close_shape](#) (shape)
- def [main](#) ()

3.1.1 Function Documentation

3.1.1.1 bounding_box()

```
def coverage_doc.bounding_box (  
    shape )
```

Creates a rectangular box which wholly contains the given shape.

Parameters

<i>shape</i>	The shape to produce a bounding box for
--------------	---

3.1.1.2 close_shape()

```
def coverage_doc.close_shape (  
    shape )
```

3.1.1.3 contained_x()

```
def coverage_doc.contained_x (  
    bounds,  
    x )
```

Checks if a point is contained within the bounding box's x axis.

Parameters

<i>bounds</i>	The bounding box
<i>x</i>	The point to check

Returns

True if the point is within and False if not

3.1.1.4 contained_y()

```
def coverage_doc.contained_y (  
    bounds,  
    y )
```

Checks if a point is contained within a bounding box's y axis.

Parameters

<i>bounds</i>	The bounding box
<i>y</i>	The point to check

Returns

True if the point is within and False if not

3.1.1.5 gradient()

```
def coverage_doc.gradient (
    p1,
    p2 )
```

3.1.1.6 graph()

```
def coverage_doc.graph (
    points,
    height,
    width,
    overlap )
```

Generates the graph required for TSP from the generated from Quantise.

Parameters

<i>points</i>	The points from Quantise, will be used as nodes in the
<i>graph</i>	

Returns

An undirected weighted graph to be used in TSP

3.1.1.7 inner_outer()

```
def coverage_doc.inner_outer (
    xy_per,
    xy_nogos,
    width )
```

Generate inner bounday for perimeter and outer boundary(s) for nogo zone(s)

The perimeter uses an inner offset to prevent going beyond the given bounds. The nogo zones uses an outer offset to prevent the mower going out forbidden areas.

Parameters

<i>xy_per</i>	The perimeter in UTM
<i>xy_nogos</i>	A list of nogo zones in UTM
<i>width</i>	The width of the robot

Returns

Returns the new perimeters to be used in TSP

3.1.1.8 inter_points()

```
def coverage_doc.inter_points (
    p1,
    p2,
    num )
```

3.1.1.9 main()

```
def coverage_doc.main ( )
```

3.1.1.10 midpoint()

```
def coverage_doc.midpoint (
    p1,
    p2 )
```

3.1.1.11 quantise()

```
def coverage_doc.quantise (
    shape,
    height,
    width,
    overlap,
    nogos )
```

Quantises a shape within a bounding box.

Each point will be the centroid of a box the size of the robot.

Parameters

<i>shape</i>	The perimeter of the shape to quantise
<i>height</i>	The height of the robot
<i>width</i>	The width of the robot
<i>overlap</i>	The desired overlap of the route

Returns

A list of centroids contained within the given shape

3.1.1.12 remove_inter()

```
def coverage_doc.remove_inter (
    points )
```

3.1.1.13 to_xy()

```
def coverage_doc.to_xy (
    perimeter,
    nogos )
```

Converts the perimeters and nogo zones from GPS to UTM.

UTM is used as the presumed area the mower will work on will result in the Earth's curvature being negligible. UTM allows for quicker calculates of distance and line equations, whilst providing movement in metres and bearing - generally more useful for navigating the mower than GPS.

```
perimeter:
nogos:
```

Returns

Returns the converted shapes along with the UTM zones. Zones should be the same to reduce error, though it is very unlikely any traversal will cross these boundary lines.

Chapter 4

File Documentation

4.1 coverage_doc.py File Reference

Namespaces

- namespace [coverage_doc](#)

Functions

- def [midpoint](#) (p1, p2)
- def [inter_points](#) (p1, p2, num)
- def [gradient](#) (p1, p2)
- def [remove_inter](#) (points)
- def [contained_y](#) (bounds, y)
Checks if a point is contained within a bounding box's y axis.
- def [contained_x](#) (bounds, x)
Checks if a point is contained within the bounding box's x axis.
- def [graph](#) (points, height, width, overlap)
Generates the graph required for TSP from the generated from Quantise.
- def [bounding_box](#) (shape)
Creates a rectangular box which wholly contains the given shape.
- def [quantise](#) (shape, height, width, overlap, nogos)
Quantises a shape within a bounding box.
- def [to_xy](#) (perimeter, nogos)
Converts the perimeters and nogo zones from GPS to UTM.
- def [inner_outer](#) (xy_per, xy_nogos, width)
Generate inner boundary for perimeter and outer boundary(s) for nogo zone(s)
- def [close_shape](#) (shape)
- def [main](#) ()

Index

- bounding_box
 - [coverage_doc](#), 5
- close_shape
 - [coverage_doc](#), 6
- contained_x
 - [coverage_doc](#), 6
- contained_y
 - [coverage_doc](#), 6
- coverage_doc, 5
 - [bounding_box](#), 5
 - [close_shape](#), 6
 - [contained_x](#), 6
 - [contained_y](#), 6
 - [gradient](#), 7
 - [graph](#), 7
 - [inner_outer](#), 7
 - [inter_points](#), 8
 - [main](#), 8
 - [midpoint](#), 8
 - [quantise](#), 8
 - [remove_inter](#), 9
 - [to_xy](#), 9
- [coverage_doc.py](#), 11
- gradient
 - [coverage_doc](#), 7
- graph
 - [coverage_doc](#), 7
- inner_outer
 - [coverage_doc](#), 7
- inter_points
 - [coverage_doc](#), 8
- main
 - [coverage_doc](#), 8
- midpoint
 - [coverage_doc](#), 8
- quantise
 - [coverage_doc](#), 8
- remove_inter
 - [coverage_doc](#), 9
- to_xy
 - [coverage_doc](#), 9