Manual for Package: gis
Revision 5M

Karl Kästner

July 5, 2023

Contents

1 @GeoImg

1.1 GeoImg

```
loading, manipulation and writing of geospacial images with world
    file
```

1.2 crop

```
cropt the image and modify the pgw accordingly
```

1.3 read

```
read geospatial image and pgw from disk and
```

1.4 write

```
write geospatial image and pgw to disk
```

2 gis

2.1 GPX

2.2 batavia_zero

3 centreline/@Centreline

3.1 Centreline

3.2 channel_planimetry

3.3 clip

3.4 connect_graph

3.5 curvature

3.6 cut

3.7 determine_width

3.8 distance

3.9 export_cross_section

3.10 export_node

3.11 export_shp

3.12 find_nearest_segment

3.13 from_polygon

3.14 from_shp

3.15 get

3.16 init

```
obj.seg_S(id(end)) = NaN;
```

3.17 init_connect

3.18 init_node_D

3.19 link_centreline

3.20 plot

3.21 plot_connection

3.22 prune

3.23 prune_leaves

3.24 prune_manually

3.25 reachable

3.26 remove_duplicate_points

3.27 resample

3.28 routing

3.29 routing2

3.30 shp_resample_simple

3.31 snmesh

3.32 squeeze

3.33 trim_ends

3.34 weighed_connection_matrix

3.35 xy2sn

4 centreline/@Segment

4.1 Segment

4.2 build_inverse_index

4.3 connectivity_matrix

4.4 init_seg_id

5 centreline

5.1 sn2xy_quadratic

5.2 thalweg

5.3 xy2sn_quadratic

6 gis

6.1 gpx_export_csv

6.2 hgt_plot

6.3 hgt_read

```
% [ floor(mednan(z(kk))) meannan(z(kk)) min(z(kk)) max(z(kk)) ]
```

6.4 hgt_read_all

6.5 hgt_resample

6.6 nmeatime

6.7 read_xyz

7 shapefile/@Shp

7.1 Shp

```
shape file processing
```

7.2 area

```
area of polygon shapes
```

7.3 buffer

```
buffer or shrink a polygon by a fixed distance
```

7.4 centroid

7.5 clip

```
crop input shape file to specified polygon
```

7.6 clip_rect

```
rectrangular crop of the shapefile
```

7.7 close_polygon

```
close polygon, i.e. make the first point identical to the last
```

7.8 concat

```
concatenate two shapefiles
```

7.9 connect_network

```
 TODO make unique
 attach segments to
XY = [cvec(shp.X),shp.;
 knnsearch for nearest n neighbours
 for each segment
```

7.10 contour

7.11 copy_attribute

```
 copy attributes from one shapefile to the other
```

7.12 cp

```
 copy a shapefile on disk
```

7.13 create

```
 create a new shapefile with given geometry
```

7.14 curvature

```
 curvature of line segments
```

7.15 cut

7.16 diameter

```
 determine diameter of polygon of every element
```

7.17 edges

```
edges of polygons line loops loops
```

7.18 export_geo

```
export geometry file undestood by SLIM
```

7.19 export_gpx

```
export data into a gpx file
```

7.20 export_gpx_track

```
export a data into a gpx track file
```

7.21 export_ldb

```
export Delft3D-4 land-boundary
```

7.22 export_poly

```
export poly-file understood by SLIM
```

7.23 export_sdf

7.24 export_spline

```
export splines (for D3D?)
```

7.25 extract_coastline

7.26 first_point

extract first point of all shapefile features

7.27 flat

7.28 generate_four_colour_index

unique colour-indices fpr poligons

7.29 generate_rectangle

generate rectangular polygon

7.30 import_geo

7.31 import_poly

import poly file

7.32 inpolygon

test if point is in any of the polygons

7.33 join_lines

join line segments

7.34 last_point

```
return last point of features
```

7.35 latlon2utm

```
convert latitude and longitude to utm
```

7.36 length

```
number of points of each feature
```

7.37 length2

```
length of line segments
```

7.38 line2point

```
convert lines to points
```

7.39 link_lines

```
link lines with same endpoints
```

7.40 make_clockwise

```
make polygons clockwise
```

7.41 merge

7.42 merge2

7.43 nan2zero

```
replace not a number values with zeros, for writing to disk
NAN values are not allowed according to the spec
```

7.44 padd_nan

```
padd NaN at end of features
```

7.45 plot

```
display the shapefile
```

7.46 points

```
returns points of the features
```

7.47 polygon_boundary

7.48 read

```
read shapefile from file
```

7.49 readZ

```
read shapefile with z-data from file
this is a workaround, as matlab cannot read files with z-data
```

7.50 reassign_id

```
assign a unique identifier to each feature
```

7.51 remove_duplicate_points

```
remove dubplicate points from features
```

7.52 remove_leaves

7.53 remove_nan

```
remove NaN points from features
```

7.54 remove_polygon_closure

```
remove last points of polygon if they are identical to the first
```

7.55 remove_short_elements

```
remove features with few points
```

7.56 renumber

```
generate a new index
```

7.57 resample

```
resample coordinates
```

7.58 resample_2

```
resample coordinates
```

7.59 resample_min

```
resample coordinates
```

7.60 resample_quick

```
resample coordinates
```

7.61 scale

7.62 segment

```
separate disjoint sections of polygons and lines
```

7.63 select_for_refinement

```
select elements for refinement
```

7.64 set_geometry

```
set feature geometry
```

7.65 set_resolution

```
set resolution for mesh generation
```

7.66 singlepart_to_multipart

concatenate line segments (parts) of shp data files into one
same as single part to multipart in qgis
returns also indices into the original file

7.67 skip

quick resampling of features by leaving out points

7.68 smooth

smooth the features

7.69 split_jump

split features where distance between points exceeds a threshold

7.70 split_line

split line features into single sements

7.71 split_nan

splits shp line and polygons at NaN into two different groups

7.72 swap_hemisphere

swap northern and southern hemisphere for UTM coordiantes

7.73 translate

translate coordinates

7.74 write

```
write the shapefile to disk
```

8 shapefile

8.1 astar_multi

8.2 astar_recursive

```
astar path finding algorithm
```

8.3 edge_chain

8.4 edge_from_bnd

8.5 preload_shp

8.6 read_gpx

8.7 shapewrite__

```
Copyright (C) 2014,2015 Philip Nienhuis

This program is free software; you can redistribute it and/or
    modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or
```

```
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses
    />.
-*- texinfo -*-
@deftypefn {Function File} {@var{status} =} shapewrite (@var{shpstr
    }, @var{fname})
Write contents of map- or geostruct to a GIS shape file.

@var{shpstr} must be a valid mapstruct or geostruct, a struct array
     with an
entry for each shape feature, with fields Geometry, BoundingBox,
    and X and Y
(mapstruct) or Lat and Lon (geostruct). For geostructs, Lat and Lon
     field
data will be written as X and Y data. Field Geometry can have data
    values
of 'Point', 'MultiPoint', 'Line', or 'Polygon', all case-
    insensitive. For
each shape feature, field BoundingBox should contain the minimum
    and maximum
(X,Y) coordinates in a 2x2 array [minX, minY; maxX, maxY]. The X
    and Y
fields should contain X (or Latitude) and Y (or Longitude)
    coordinates for
each point or vertex as row vectors; for polylines and polygons
    vertices of
each subfeature (if present) should be separated by NaN entries.

@var{fname} should be a valid shape file name, optionally with a '.
    shp'
suffix.

shapewrite produces 2 or 3 files, i.e. a .shp file (the actual
    shape file),
a .shx file (index file), and if @var{shpstr} contained additional
    fields,
a .dbf file (dBase type 3) with the contents of those additional
    fields.

@var{status} is 1 if the shape file set was written successfully, 0
otherwise.
```

```
@seealso{shaperead, shapeinfo}
@end deftypefn
Author: Philip Nienhuis <prnienhuis@users.sf.net>
Created: 2014-12-30
Input validation
Assess shape variable type (oct or ml/geo ml/map)
Yep. Find out what type
Assume it is an Octave-style struct read by shaperead
Assume it is a Matlab-style mapstruct
Assume it is a Matlab-style geostruct
Not a supported struct type
Check file name
Later on bname.shx and bname.dbf will be read
Prepare a few things
Change Lat/Lon fields into X/Y
Only now (after input checks) open .shp and .shx files & rewind just to be sure
Write headers in .shp & .shx (identical). First magic number 9994 + 5 zeros
In between here = filelength in 16-bit words (single). For .shx it's known
Next, shp file version
Shape feature type
Bounding box. Can be run later for ML type shape structs. Fill with zeros
Skip to start of first record position
Write shape features one by one
Write record start pos to .shx file
Write record contents
Point
Record index number
Record length (fixed)
Shape type
Simply write XY cordinates
MultiPoint
Record index number
Record length
Shape type
Bounding box
Nr of points
Polyline/-gon
Record index number
Prepare multipart polygons
Augment idx for later on, & this trick eliminates trailing NaN rows
Record length
Shape type
Bounding box
Number of parts, number of points, part pointers
Write file length into .shp header
```

```
Close files
Check for dbfwrite function
Write rest of attributes
Attributes + shp data in mapstruct
Attributes + shp data in geostruct
```

8.8 shapewrite_man

8.9 shp2geo

8.10 shp2kml

8.11 shp_plot_attribute

8.12 split_section

8.13 write_polygon

9 gis

9.1 shp2csv

9.2 write_xyz