Package ‘decido’

October 13, 2022

Version  0.3.0
Title   Bindings for 'Mapbox' Ear Cutting Triangulation Library
Description  Provides constrained triangulation of polygons. Ear cutting (or ear
clipping) applies constrained triangulation by successively 'cutting' triangles from
a polygon defined by path/s. Holes are supported by introducing a bridge segment
between polygon paths. This package wraps the 'header-only' library 'earcut.hpp'
<https://github.com/mapbox/earcut.hpp.git> which includes a reference to the
License   MIT + file LICENSE
Encoding  UTF-8
LazyData  true
ByteCompile  true
LinkingTo   Rcpp
Depends   R (>= 3.2.5)
Imports  Rcpp
RoxygenNote  7.1.0
Suggests  covr, testthat, knitr, oz, rmarkdown, spelling
SystemRequirements  C++11
VignetteBuilder  knitr
URL  https://hypertidy.github.io/decido
BugReports  https://github.com/hypertidy/decido/issues
Language  en-US
NeedsCompilation  yes
Author  Michael Sumner [aut, cre], Andrew Smith [ctb] (provided C++ guidance),
Mapbox [cph] (author of header library earcut.hpp), Mark Padgham [ctb] (help with
CRAN issues), David Cooley [ctb] (added header capability for linking from other
packages)
Maintainer  Michael Sumner <mdsumner@gmail.com>
R topics documented:

earcut ................................................................. 2
plot_ears ............................................................. 4

Index 5

earcut  Constrained polygon triangulation

Description

Produce a triangulation index into x,y coordinates of a polygon that may include holes. Holes are specified by input argument holes which marks the starting index of each hole, if any.

Usage

earcut(xy, holes = 0, ...)

## Default S3 method:
earcut(xy, holes = 0L, ...)

Arguments

xy xy-coordinates, either a list, matrix, or data frame
holes index of starting position of each hole in x,y, leave set to 0 if no holes
... unused

Details

Triangles are returned in counter-clockwise orientation, a common convention that ascribes a positive area to the triangle. (Orientation may be collinear or numerically ambiguous and so may be undetermined).

Ear cutting (or ear clipping) applies constrained triangulation by successively 'cutting' triangles from a polygon defined by path/s. Holes are supported, the earcut library works with single-island-with-holes polygons, analogous to the POLYGON type in simple features.

To understand the specification of holes, see the examples with comment starting "1) Notice how the hole begins ..." in relation to the example code.

Value

integer vector of triangle index, in sets of three
earcut

See Also

plot_ears

Examples

## single ring polygon
x <- c(0, 0, 0.75, 1, 0.5, 0.8, 0.69)
y <- c(0, 1, 0.8, 0.7, 0.6, 0)
(ind <- earcut(cbind(x, y)))
plot_ears(cbind(x, y), ind)

## polygon with a hole
x <- c(0, 0, 0.75, 1, 0.5, 0.8, 0.69,
      0.2, 0.5, 0.5, 0.3, 0.2)
y <- c(0, 1, 0.8, 0.7, 0.6, 0,
      0.2, 0.2, 0.4, 0.6, 0.4)
ind <- earcut(cbind(x, y), holes = 8)
plot_ears(cbind(x, y), ind)

## 1) Notice how the hole begins at index 8,
## hence holes = 8 above, and holes = c(8, 13) below
plot_ears(cbind(x, y), ind, col = "grey", border = NA)
text(x, y, labels = seq_along(x), pos = 2)

## add another hole
x <- c(0, 0, 0.75, 1, 0.5, 0.8, 0.69,
      0.2, 0.5, 0.5, 0.3, 0.2, 0.15, 0.23, 0.2)
y <- c(0, 1, 0.8, 0.7, 0.6, 0,
      0.2, 0.2, 0.4, 0.4, 0.6, 0.4,
      0.65, 0.65, 0.81)
ind <- earcut(cbind(x, y), holes = c(8, 13))
plot_ears(cbind(x, y), ind, col = "grey")

## simpler shape with more than one hole
# the two inside holes are open to each other
# (so we can use the same data for one hole or two)
x <- c(0, 0, 1, 1,
      0.4, 0.2, 0.2, 0.4,
      0.6, 0.8, 0.8, 0.6)

y <- c(0, 1, 1, 0,
      0.2, 0.2, 0.4, 0.4,
      0.6, 0.6, 0.4, 0.4)
ind <- decido::earcut(cbind(x, y), holes = c(5, 9))
plot_ears(cbind(x, y), ind, col = "grey")
plot_holes(cbind(x, y), holes = c(5, 9), col = "grey")
ind <- decido::earcut(cbind(x, y), holes = 5)
plot_ears(cbind(x, y), ind, col = "grey")
plot_holes(cbind(x, y), holes = 5, col = "grey")
plot_ears

Plot ears or polygons

Description

Plot the triangles produced by earcut, or plot the polygon paths using the same interface as earcut uses. This allows for easy comparison and checking of what the results should be.

Usage

plot_ears(xy, idx, add = FALSE, ...)
plot_holes(xy, holes = 0, add = FALSE, ...)

Arguments

xy xy-coordinates, either a list, matrix, or data frame
idx index of triangles
add add to current plot, or create a new
... arguments to polypath
holes index of starting position of holes (see earcut)

Details

For both functions the first input is a matrix of x,y coordinates.

For plot_ears the second input is the index output of earcut. The index is treated in sets of 3 values, with individual calls to polypath to draw a polygon for each triangle.

For plot_holes the second input is the holes argument that would be used for earcut. This is used to split the coordinates at these positions, inserting NA values as per the mechanism used by graphics::polypath to break coordinates into separate polygon rings. (There’s no winding rule here plot_rules is hard-coded to always use the evenodd rule, so that winding order may be ignored).

See Also

earcut

Examples

## after ?polypath
x <- cbind(c(.1, .1, .9, .9, .2, .2, .8, .8),
           c(.1, .9, .9, .1, .2, .8, .8, .2))
plot_holes(x, holes = 5, col = "grey")
Index

earcut, 2, 4

graphics::polypath, 4

plot_ears, 4, 4
plot_holes, 4
plot_holes (plot_ears), 4
polypath, 4