Package: rbff (via r-universe)

August 27, 2024

Title R Interface to Boundary First Flattening Software (BFF)

Version 0.0.0.9000

Description Flatten 3D meshes into arbitrary 2D shapes using boundary first flattening (<https://github.com/GeometryCollective/boundary-first-flattening>).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

LinkingTo Rcpp

Imports rgl, Rcpp, sf, Rvcg, readobj, colourvalues, magrittr, imager

Suggests rmarkdown, knitr, ragg, sfdct

VignetteBuilder knitr

Depends R (>= 2.10)

SystemRequirements openblas, suitesparse

Repository https://rdinnager.r-universe.dev

RemoteUrl https://github.com/rdinnager/rbff

RemoteRef HEAD

RemoteSha 899e4f32558c82e180ecd79df247f1c8e5e92308

Contents

`beetle` *Beetle 3d Model*

Description

Beetle 3d Model

Usage

```
beetle
```

Format

A `mesh3d` object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

---

`bff_flatten` *Automatically flatten a 3d mesh using boundary first flattening*

Description

Flattening will only work with closed objects if cones are inserted (cuts or seams). Closed objects can instead be mapped to a sphere however. If both `to_disk` and `to_sphere` are `FALSE`, a target boundary shape will be determined automatically that minimized area and angle distortion. If you want to flatten to a particular target boundary shape, use `bff_flatten_to_shape()`

Usage

```
bff_flatten(
  mesh,
  n_cones = 0,
  to_disk = FALSE,
  to_sphere = FALSE,
  normalise = TRUE
)
```

Arguments

mesh	mesh3d object to flatten
n_cones	Number of cone singularities to insert into mesh to reduce distortion. Ignored if to_disk = TRUE
to_disk	Should the mesh be flattened to a disk?
to_sphere	Should the mesh be mapped to a sphere instead (only works with closed objects).
normalise	Should the 2d mapping be normalised between 0 and 1 on the x and y axes?

Value

A bff_flattened object containing the original mesh and its flattened version with corresponding vertices unless to_sphere = TRUE, in which case a bff_sphered object is returned.

Examples

```
data(face)
options(rgl.useNULL = TRUE)
flat_face <- bff_flatten(face)
```

---

bff_flatten_to_shape *Flatten a mesh to a particular boundary shape*

Description

Flatten a mesh to a particular boundary shape

Usage

```
bff_flatten_to_shape(mesh, boundary_shape, normalise = TRUE)
```

Arguments

normalise

`bff_place_image` *Place an image interactively on the flattened mesh and see it mapped to the original 3D mesh.*

Description

Place an image interactively on the flattened mesh and see it mapped to the original 3D mesh.

Usage

```
bff_place_image(
  x,
  expression = NULL,
  filename = NULL,
  tile = TRUE,
  bg = "white",
  alpha_flatten = FALSE,
  ...
)
```

Arguments

`x`	A `bff_flattened` object
`expression`	An R expression that generates an image.
`filename`	Alternative to expression: provide the file name of a png image directly
`tile`	If the image does not cover the whole flattened mesh, should it be tiled (e.g repeated) so that it fills the whole mesh? *Also not implemented yet*
`bg`	Background colour for if `tile = FALSE`.
`alpha_flatten`	Should the alpha channel be flattened? By default, parts of the mesh with completely transparent colours will be invisible. Flattening removes the alpha channel by layering the image over the `bg` colour
`...`	Further arguments passed to the graphic device is `expression` is not `NULL`

Value

A `bff_textured` object containing the original mesh with updated textcoords, its flattened version, and the image for texture mapping

bff_vis_metrics *Visualise the result of flattening a 3d mesh.*

Description

Visualise the result of flattening a 3d mesh.

Usage

```
bff_vis_metrics(
  x,
  metric = c("area distortion", "vertice density", "curvature", "mesh"),
  ...
)
```

Arguments

x	A bff_flattened or bff_sphered object
metric	Which metric to display. One of "area distortion", "vertice density", "curvature", or "mesh"
...	Additional arguments passed on to the drawing functions (shade3d() or wire3d())

Value

None.

Examples

```
data(face)
## don't run the next line if you want to see the visualisation
options(rgl.useNULL = TRUE)
face_flat <- bff_flatten(face)
bff_vis_metrics(face_flat)
```

box *Box 3d Model*

Description

Box 3d Model

Usage

```
box
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

---

bunny *Bunny 3d Model*

Description

Bunny 3d Model

Usage

```
bunny
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

---

cowhead *Cowhead 3d Model*

Description

Cowhead 3d Model

Usage

```
cowhead
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

face *Face 3d Model*

Description

Face 3d Model

Usage

```
face
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

fish *Fish 3d Model*

Description

Fish 3d Model

Usage

```
fish
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

hemisphere *Hemisphere 3d Model*

Description

Hemisphere 3d Model

Usage

```
hemisphere
```

Format

A `mesh3d` object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

kitten *Kitten 3d Model*

Description

Kitten 3d Model

Usage

```
kitten
```

Format

A `mesh3d` object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

maze *Maze 3d Model*

Description

Maze 3d Model

Usage

```
maze
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

punctured_torus *Punctured Torus 3d Model*

Description

Punctured Torus 3d Model

Usage

```
punctured_torus
```

Format

A mesh3d object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

r_logo_png *Get the R logo png file path.*

Description

Get the R logo png file path.

Usage

```
r_logo_png()
```

Value

A file path to the R logo png

Examples

```
plot(imager::load.image(r_logo_png()))
```

spothead *Spothead 3d Model*

Description

Spothead 3d Model

Usage

```
spothead
```

Format

A `mesh3d` object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

`teapot` *Teapot 3d Model*

Description

Teapot 3d Model

Usage

```
teapot
```

Format

A `mesh3d` object.

Source

https://github.com/GeometryCollective/boundary-first-flattening/tree/master/input

Index