Package 'gdi'

April 19, 2024

Title Volumetric Analysis using Graphic Double Integration

Version 1.6.0

Date 2024-04-19

Author Darius Nau [aut, cre]

Maintainer Darius Nau <dariusnau@gmx.at>

Description Tools implementing an automated version of the graphic double integration technique (GDI) for volume implementation, and some other related utilities for paleontological image-analysis. GDI was first employed by Jerison (1973) <ISBN:9780323141086> and Hurlburt (1999) <doi:10.1080/02724634.1999.10011145> and is primarily used for volume or mass estimation of (extinct) animals. The package 'gdi' aims to make this technique as convenient and versatile as possible. The core functions of 'gdi' provide utilities for automatically measuring diameters from digital silhouettes provided as image files and calculating volume via graphic double integration with simple elliptical, superelliptical (following Motani 2001 <doi:10.1666/0094-8373(2001)027%3C0735:EBMFST%3E2.0.CO;2>) or complex cross-sectional models. Additionally, the package provides functions for estimating the center of mass position (COM), the moment of inertia (I) for 3D shapes and the second moment of area (Ix, Iy, Iz) of 2D cross-sections, as well as for visualization of results.

Depends jpeg, png

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

R topics documented:

`cscorr` *Measure and analyze cross-sectional geometry supplied as an image.*

Description

Measure and analyze cross-sectional geometry supplied as an image.

Usage

```
cscorr(
  image_file,
  threshold = 0.5,
  channel = 4,
  method = "greater",
  return = "area_corr",
  k = 2,
  scale = 1
)
```

Arguments

`image_file`	Image to be read. Images can be jpeg or png files, or a previously read image saved as an array/matrix-type object in R.
`threshold`	Reference value for color criterium after which pixels that are part of the cross-section are differentiated from the background.
`channel`	Color channel to which to apply the threshold criterium. Default is 4 (alpha channel of rgba image). Channel setting needs to be adjusted depending on the color mode of the image used (e.g. there are two channels to choose from in a greyscale image, and 3 in an rgb image).
`method`	Method for determining which pixels to count. Default "greater" counts pixels with value greater than threshold (e.g. higher opacity, in the case of an alpha channel). "less" counts pixels with a value less than the threshold. "not" counts all pixels not precisely matching threshold. Any other character string results in only pixels exactly matching the value given as threshold being counted.

return	What value to return. Possible values are "area_corr" (Default, returns ratio between measured area and area of ellipse with same horizontal and vertical diameters), "aspect_ratio" (returns aspect ratio), "diameters" (returns diameters), "area" (returns area) and "rotI" (returns correction factors for rotational inertia calculations). Any other value for this parameter will prompt the function to return a vector containing all of these outputs.
k	optional superellipse exponent for the (super)ellipse to which the measurements should be compared (for the "area_corr" setting for the parameter return).
scale	Optional scale of the image (for raw area measurements).

Value

Either a numeric of length 1 (depending on the input of the return parameter), defaulting to the area correction factor (if return=="area_corr"), or (if return is left empty or does not match any of the predefined settings) a numeric vector with 8 elements, containing all the possible outputs (x and y diameters, aspect ratio, area and area correction factor, correction factors representing ratios of rotational inertia in x, y and z planes relative an ellipse of equal diameters).

Examples

```
fdir <- system.file(package="gdi")
correction_factor <- cscorr(file.path(fdir,"exdata","cross_section.png"))
```

---

csI

Calculates the second moment of area (=area moment of inertia, Ix and Iy) and polar moment of inertia (Iz or J) for a cross-section given as an image.

Description

Calculates the second moment of area (=area moment of inertia, Ix and Iy) and polar moment of inertia (Iz or J) for a cross-section given as an image.

Usage

```
csI(
  image_file,
  threshold = 0.5,
  channel = 4,
  method = "greater",
  scale = 1,
  return = "total"
)
```

Arguments

image_file	Image to be read. Images can be jpeg or png files, or a previously read image saved as an object in R.
threshold	Reference value for color criterium after which pixels that are part of the silhouette should be differentiated from the background.
channel	color channel to which to apply the threshold criterium. Default is 4 (alpha channel of rgba image). Channel setting needs to be adjusted depending on the color mode of the image used (e.g. there are two channels to choose from in a greyscale image, and 3 in an rgb image).
method	Method for determining which pixels to count. Default "greater" counts pixels with value greater than threshold (e.g. higher opacity, in the case of an alpha channel). "less" counts pixels with a value less than the threshold. "not" counts all pixels not precisely matching threshold. Any other character string results in only pixels exactly matching the value given as threshold being counted.
scale	Optional scale of the image (number of pixels per linear unit).
return	What to return, defaults to returning both x and y second moments of area and polar moment of inertia for the entire shape (if return=="total"), otherwise returns raw data matrix for all pixels.

Value

A numeric vector containing Ix, Iy and Iz for the shape (default), or a matrix containing area moments and coordinates for each pixel in the image, as well as area moments converted relative to the common centroid of the shape using parallel axis theorem.

Examples

```
fdir <- system.file(package="gdi")
csI(file.path(fdir,"exdata","cross_section.png"))
```

---

fdetect

Tool to help determine which threshold value and method to use with measuresil() or cscorr(). The function analyzes all pixels along the edges of the image to determine the background color, to help with deciding on appropriate settings and avoid errors introduced by inappropriate settings

---

Description

Tool to help determine which threshold value and method to use with measuresil() or cscorr(). The function analyzes all pixels along the edges of the image to determine the background color, to help with deciding on appropriate settings and avoid errors introduced by inappropriate settings

Usage

```
fdetect(image_file, threshold = 0.5, channel = 4, plot = FALSE)
```

Arguments

image_file	Image to be read. Images can be jpeg or png files, or a previously read image saved as an object in R.
threshold	Reference value for color criterium after which pixels that are part of the silhouette should be differentiated from the background.
channel	Color channel to which to apply the threshold criterium. Default is 4 (alpha channel of rgba image). Channel setting needs to be adjusted depending on the color mode of the image used (e.g. there are two channels to choose from in a greyscale image, and 3 in an rgb image).
plot	Whether to plot a histogram with the detected color values (if TRUE) or not (if FALSE, default).

Value

A list()-object containing: $edgetable (a table of the different color values detected and their respective frequencies), $histogram (a histogram-object of the color values), $most_common (the most common color value found), $foreground (a character string, indicating whether the foreground color value is likely "greater" or "less" than the specified threshold), $result (a character string giving a summary of the results)

Examples

```
fdir <- system.file(package="gdi")
fdetect(file.path(fdir,"exdata","lat.png"))
```

---

gdi *Estimate volume using Graphic Double Integration.*

Description

Estimate volume using Graphic Double Integration.

Usage

```
gdi(
  lat,
  dors,
  indices = NULL,
  scale = 10,
  sliceL = 1/scale,
  method = "raw",
  k = 2,
  corr = 1,
  smooth.ends = FALSE,
  return = "total"
)
```

Arguments

lat	Measurements of diameter in lateral view/first of two orthogonal views to be used with the gdi. Can be either a numeric vector, a data.frame (output of measuresil(...,return="all") with a collumn named "diameter", or a text file with diameter measurements to be scanned.
dors	Measurements of diameter in dorsal view/second of two orthogonal views to be used with the gdi. Can be either a numeric vector, a data.frame (output of measuresil(...,return="all") with a collumn named "diameter", or a text file with diameter measurements to be scanned. Must be the same length as lat.
indices	Optional indices specifying a subset of the silhouette measurement vectors to be analyzed. Useful if separate segment calculations are desired.
scale	Scale of the data, given in terms of how many units of the input data (e.g. pixels) are in one side of the desired unit of output volume. Defaults to 10.
sliceL	Length of individual segments to be used in the GDI. Defaults to 1/scale.
method	Method to be used for the GDI. Default "raw" setting calculates each segment as an elliptical cylinder with volume = Area * SliceL. Any other string will result in volume being calculated as an elliptical frustum with base areas based on the measurements of segments i and i+1.
k	Superellipse exponent to be used for the cross-sectional area. Defaults to 2.0 (normal ellipse).
corr	Correction factor for area of cross-sections, calculated as the ratio between the actual cross-sectional area and that of a (super)ellipse (depending on the specified exponent k) with the same diameters. This setting enables the function to account for complex, non-elliptical cross-sections. Default value is 1, i.e. no correction. Can be either a single number, or a numeric vector of the same length as lat and dors (in the case of a changing cross-sectional geometry along the length of the body).
smooth.ends	If method != "raw", specify whether first and last segments should be left raw, or taper to 0 (i.e. be approximated as cones). Only applies if there are no leading or following zeros in the measurement vectors.
return	Determines whether to report the estimated total volume (if default/"total"), or a data.frame() with segment radii, areas and volumes (if left empty of any other character string.

Value

Either a single number representing the total volume estimated (with names indicating the horizontal length of the silhouette in the unit determined by scale), or (if return!="total") a data.frame() containing columns with the radii in both dimensions, the estimated elliptical or superelliptical areas, and the segment volumes.

Examples

```
lateral <- rep(2,4) #generate example data
dorsal <- rep(2,4)
gdi(lat=lateral, dors=dorsal, scale=10, method="raw", k=2.0)
gdi(lat=lateral, dors=lateral/2, scale=10, method="smooth", k=2.3)
```

hCOM	*Finds the horizontal (x axis, i.e. the axis vertical to the cross-sections) position of the center of mass (COM) of the volume. Experimental; only valid for "raw" gdi results with segment volumes approximated as elliptical prisms, or for manually supplied segment COMs. COM is calculated as a weighted mean of all segment COMs, with the segment mass as the weighting factor.*

Description

Finds the horizontal (x axis, i.e. the axis vertical to the cross-sections) position of the center of mass (COM) of the volume. Experimental; only valid for "raw" gdi results with segment volumes approximated as elliptical prisms, or for manually supplied segment COMs. COM is calculated as a weighted mean of all segment COMs, with the segment mass as the weighting factor.

Usage

```
hCOM(
  x,
  volumes = NULL,
  align = "h",
  subtract = NULL,
  densities = NULL,
  scale = 1
)
```

Arguments

x	Either a data frame that is the output of gdi(..., return="all"), or a numeric vector of horizontal segment COM positions.
volumes	An optional separate vector of volumes, required if x is not a data.frame containing volumes.
align	alignment of the silhouette, if "h" (default) the silhouette is assumed to be horizontally aligned, if any other value (e.g. "v") then the silhouette is assumed to be vertically aligned.
subtract	An optional separate vector of volumes, with length equal to the length or nrow() of x, to be subtracted from the volumes for the COM calculation.
densities	An optional vector of segment densities, with length equal to the length or nrow() of x, to be multiplied with the volumes for the COM calculation. If both subtract and densities are supplied, the density is applied only to the "residual" volume that is left after subtraction.
scale	Optional scale value (number of pixels to chosen unit of measurement)

Value

An object of class numeric() containing the x coordinate of the center of mass of the shape, in pixels (or chosen units, if manually calculated)

Examples

```
fdir <- system.file(package="gdi")
measuresil(file.path(fdir,"exdata","lat.png"), return="all")->lat_
measuresil(file.path(fdir,"exdata","dors.png"), return="all")->dors_
gdi(lat_, dors_, return="all")->gdiresults
hCOM(gdiresults)
```

`imghist` *Simple histogram analysis for all color values in an input image. Can be used to help assess whether a chosen threshold value is appropriate for differentiating the silhouette from the background, or for general image analysis purposes.*

Description

Simple histogram analysis for all color values in an input image. Can be used to help assess whether a chosen threshold value is appropriate for differentiating the silhouette from the background, or for general image analysis purposes.

Usage

```
imghist(
  image_file,
  threshold = 0.5,
  channel = 4,
  breaks = seq(0, 1, 0.05),
  plot = TRUE,
  unique = FALSE
)
```

Arguments

`image_file`	Image to be read. Images can be jpeg or png files, or a previously read image saved as an object in R.
`threshold`	Reference value for color criterium after which pixels that are part of the silhouette should be differentiated from the background.
`channel`	color channel to which to apply the threshold criterium. Default is 4 (alpha channel of rgba image). Channel setting needs to be adjusted depending on the color mode of the image used (e.g. there are two channels to choose from in a greyscale image, and 3 in an rgb image).
`breaks`	A vector of breaks for the histogram, defaults to a bin width of 0.05 between color values of 0 and 1.
`plot`	Whether to plot a histogram, defaults to TRUE
`unique`	Whether to return counts for unique color values, defaults to FALSE.

Value

A plotted histogram (unless plot==FALSE), and a matrix containing the counts from the histogram (default) or the counts for unique color values (if unique==TRUE).

Examples

```
fdir <- system.file(package="gdi")
imghist(file.path(fdir,"exdata","lat.png"))
```

`measuresil` *Take pixel-by-pixel measurements of a silhouette in jpeg or png format for use with the gdi function.*

Description

Take pixel-by-pixel measurements of a silhouette in jpeg or png format for use with the gdi function.

Usage

```
measuresil(
  image_file,
  threshold = 0.5,
  channel = 4,
  method = "greater",
  align = "h",
  return = "diameters"
)
```

Arguments

`image_file`	Image to be read. Images can be jpeg or png files, or a previously read image saved as an object in R.
`threshold`	Reference value for color criterium after which pixels that are part of the silhouette are differentiated from the background.
`channel`	Color channel to which to apply the threshold criterium. Default is 4 (alpha channel of rgba image). Channel setting needs to be adjusted depending on the color mode of the image used (e.g. there are two channels to choose from in a greyscale image with transparency, and 3 in an rgb image without transparency, or 4 in a full rgba image).
`method`	Method for determining which pixels to count. Default "greater" counts pixels with value greater than threshold (e.g. higher opacity, in the case of an alpha channel). "less" counts pixels with a value less than the threshold. "not" counts all pixels not precisely matching threshold. Any other character string results in only pixels exactly matching the value given as threshold being counted.
`align`	Indicate whether the silhouette long axis is aligned horizontally (setting "h", default), or vertically (any other parameter setting).

return	Setting for what to return, default setting ("diameters") returns a single vector containing the diameters, any other setting returns a data frame containing centers and diameters.

Value

A numeric vector giving the measurements of the silhouette

Examples

```
fdir <- system.file(package="gdi")
lat <- measuresil(file.path(fdir,"exdata","lat.png"))
```

---

plot_sil *Plots a silhouette read by measuresil()*

Description

Plots a silhouette read by measuresil()

Usage

```
plot_sil(
  sil,
  flip = FALSE,
  add = FALSE,
  xoffset = 0,
  yoffset = 0,
  alpha = 1,
  col = "grey",
  border = "darkgrey",
  scale = 1,
  xlab = "",
  ylab = "",
  ...
)
```

Arguments

sil	A data frame that is the output of measuresil(..., return="all"), containing the center and the diameter of the silhouette at each value for x.
flip	Logical indicating whether to flip axes (needed if measuresil() was performed using align="v", defaults to FALSE.
add	Logical indicating whether to add silhoutte to an existing plot (defaults to FALSE)
xoffset	Optional value by which to shift the silhouette on the x axis
yoffset	Optional value by which to shift the silhouette on the y axis

alpha	Opacity value for fill of polygon (defaults to 1)
col	Fill color of polygon (defaults to "grey")
border	Border color of polygon (defaults to "darkgrey")
scale	Scale to use for plotting (given in pixels/unit). Defaults to 1.
xlab	X axis label to use for plotting (if add=FALSE)
ylab	Y axis label to use for plotting (if add=FALSE)
...	Other parameters to pass on to plot() or lines()

Value

A plotted silhouette

Examples

```
fdir <- system.file(package="gdi")
measuresil(file.path(fdir,"exdata","lat.png"), return="all")->lat_
plot_sil(lat_)
```

`rotI` *Calculates the rotational inertia of a body. Only works with simple circular/elliptical and rectangular cross-sections, thus pixel-precise estimates are recommended.*

Description

Calculates the rotational inertia of a body. Only works with simple circular/elliptical and rectangular cross-sections, thus pixel-precise estimates are recommended.

Usage

```
rotI(
  x,
  y = NULL,
  dors_diam = NULL,
  lat_diam = NULL,
  axis_coord = NULL,
  axis = "yaw",
  volumes = NULL,
  densities = 1,
  corr = 1,
  scale = 1
)
```

Arguments

x	Either a data frame that is structured like output of gdi(..., return="all") containing masses and diameters for pixel-wide segments, or a numeric vector of segment COM positions.
y	An optional vector of vertical (dorsoventral) segment COM positions.
dors_diam	An optional vector of transverse diameters of the silhouette, required if not contained in x.
lat_diam	An optional vector of vertical diameters of the silhouette, required if not contained in x. Needed if inertia for "roll" or "pitch" should be calculated.
axis_coord	An optional coordinate of the axis of rotation, defaults to the center of mass of the entire volume if not set.
axis	Axis of rotation, defaults to "yaw" (i.e. rotation around vertical axis), can also be "pitch" (rotation around transverse axis) or "roll" (rotation around horizontal axis). For yaw rotation, the body is assumed to be bilaterally symmetrical, whereas for pitch rotation, dorsoventral variation in COM of segments is taken into account.
volumes	An optional separate vector of volumes, required if x is not a data.frame containing them.
densities	An optional vector of segment densities, with length equal to the length or nrow() of x, to be multiplied with the volumes to calculate masses used in the inertial calculation.
corr	An optional correction factor for the cross-sectional shape, given as the ratio between the characteristic mass moment of inertia of a plane with the given shape (e.g. determined by cscorr()) and an elliptical plane with the same diameters and assigned mass. Allows the calculation of moments of inertia for bodies with arbitrary cross-sectional shapes.
scale	Scale value, i.e. number of pixels representing 1 m

Value

A numeric vector containing: The total mass (on the basis of gdi volumes and optional densities), the rotational inertia of the shape using a point mass approximation of each segment (yaw/pitch rotation only), rotational inertia using a cylindrical approximation for each segment, rotational inertia using a cuboidal approximation (note that this only changes the mass distribution, while segment masses are still assumed to correspond to gdi results multiplied by optional densities), and rotational inertia using a corrected cylindrical approximation based on value for corr.

Examples

```
fdir <- system.file(package="gdi")
measuresil(file.path(fdir,"exdata","lat.png"), return="all")->lat_
measuresil(file.path(fdir,"exdata","dors.png"), return="all")->dors_
gdi(lat_, dors_, return="all")->gdiresults
rotI(gdiresults)
```

sellipse — *Estimate area of a superellipse. Assistant function for gdi.*

Description

Estimate area of a superellipse. Assistant function for gdi.

Usage

```
sellipse(a, b, k)
```

Arguments

a	First radius of the superellipse.
b	Second radius of the superellipse.
k	superellipse exponent.

Value

A single number giving the area of the superellipse

Examples

```
major_radius<-2
minor_radius<-3
exponent<-2.3
sellipse(major_radius, minor_radius, exponent)
```

sellipse.coo — *calculate coordinates for plotting a superellipse for visualizing body cross-sections*

Description

calculate coordinates for plotting a superellipse for visualizing body cross-sections

Usage

```
sellipse.coo(k, res = 100)
```

Arguments

k	superellipse exponent.
res	the desired resolution

Value

a data frame containing

Examples

```
sellipse.coo(2.0)->df #get coordinates for normal ellipse (exponent k=2)
plot(df$x,df$y,col="black", type="l") #plot normal ellipse
sellipse.coo(2.3)->df2 # get coordinates for superellipse with exponent 2.3
lines(df$x,df$y, col="blue") #plot superellipse
```

`vCOM` *Finds the vertical (y axis, i.e. the axis parallel to the cross-section diameter) position of the center of mass (COM) of the volume. Experimental; only valid for "raw" gdi results with segment volumes approximated as elliptical prisms, or for manually supplied segment COMs. COM is calculated as a weighted mean of all segment COMs, with the segment mass as the weighting factor. Estimates have lower accuracy compared to hCOM, because cross-sectional geometry and variation in density throughout the cross-section is not taken into account.*

Description

Finds the vertical (y axis, i.e. the axis parallel to the cross-section diameter) position of the center of mass (COM) of the volume. Experimental; only valid for "raw" gdi results with segment volumes approximated as elliptical prisms, or for manually supplied segment COMs. COM is calculated as a weighted mean of all segment COMs, with the segment mass as the weighting factor. Estimates have lower accuracy compared to hCOM, because cross-sectional geometry and variation in density throughout the cross-section is not taken into account.

Usage

```
vCOM(
  y,
  volumes = NULL,
  subtract = NULL,
  densities = NULL,
  scale = 1,
  from_top = FALSE
)
```

Arguments

`y`	A data.frame that is the output of gdi(..., return="all"), or a numeric vector containing vertical COM positions for segments
`volumes`	An optional separate vector or data.frame (output of gdi(...,return="all") or vector of volumes.

subtract	An optional separate vector of volumes, with length equal to the length or nrow() of x, to be subtracted from the volumes for the COM calculation.
densities	An optional vector of segment densities, with length equal to the length or nrow() of x, to be multiplied with the volumes for the COM calculation. If both subtract and densities are supplied, the density is applied only to the "residual" volume that is left after subtraction.
scale	Optional scale value (number of pixels to chosen unit of measurement)
from_top	Whether the output coordinate should be measured from the top of the image (standard for image processing software), if TRUE, or from the bottom (standard for plotting in R (if FALSE, default). If TRUE, an attribute to y, containing the vertical dimension relative to which the measurement should be taken is required.

Value

An object of class numeric() containing the y coordinate of the center of mass of the shape, in pixels (or chosen units, if manually calculated)

Examples

```
fdir <- system.file(package="gdi")
measuresil(file.path(fdir,"exdata","lat.png"), return="all")->lat_
measuresil(file.path(fdir,"exdata","dors.png"), return="all")->dors_
gdi(lat_, dors_, return="all")->gdiresults
vCOM(gdiresults)
```

Index