

Package ‘euclid’

November 22, 2017

Type Package

Title Useful primitives for euclidean geometry

Version 0.1.0

Author J. T. Atria

Maintainer J. T. Atria <jtatria@nomoi.org>

Description Collection of useful primitives for euclidean geometry, developed with a focus on separation of low level computing implementations and high-level algebraic and geometric operations, in order to eventually allow for easy extension to different computing backends.

License GPL v3.0 or greater.

Encoding UTF-8

LazyData true

Imports magrittr, Matrix, polyclip

RoxygenNote 6.0.1

R topics documented:

A	3
bsize	3
C	3
center	3
chart	4
color_alpha	4
color_make	4
compose	4
cosf	5
cotf	5
cscf	5
deg2rad	5
draw_seg	6
draw_vec	6
dwise	6
east	6

interpolate	7
iszero	7
is_ccw	7
is_plist	7
lfilter	8
lineclip	8
lmap	8
north	8
one	9
pclear	9
pintersect	9
position	9
pushout	10
rad2deg	10
random	10
resize	10
rotate	11
rotation	11
secf	11
sinf	11
smul	12
south	12
s_v_op	12
tanf	13
theta	13
to_l	13
to_m	13
translate	14
vadd	14
vaxis	14
vdif	14
vdist	15
vhead	15
vinnerp	15
vnorm	15
vorth	16
vproj	16
vrej	16
vunit	16
vv_s_op	17
vv_v_op	17
vv_w_op	17
vwise	17
v_s_op	18
v_v_op	18
v_w_op	18
west	18
zero	19

A	3
---	---

Index	20
-------	----

A	<i>Unit vector in the a direction</i>
---	---------------------------------------

Description

Unit vector in the a direction

Usage

A(d = 2, a = 1)

bsize	<i>Size of minimal bounding box (2d only)</i>
-------	---

Description

Size of minimal bounding box (2d only)

Usage

bsize(m)

C	<i>Center of m</i>
---	--------------------

Description

Center of m

Usage

C(m)

center	<i>Center object on O()</i>
--------	-----------------------------

Description

Center object on O()

Usage

center(v)

chart	Create a chart for thr given objects
-------	--------------------------------------

Description

Create a chart for thr given objects

Usage

```
chart(..., scale = TRUE, asp = 1, axes = TRUE, ylab = NA, xlab = NA,  
      par = NULL)
```

color_alpha	Add alpha to colors
-------------	---------------------

Description

Add alpha to colors

Usage

```
color_alpha(col = par("fg"), alpha = 0.1)
```

color_make	Make color palette
------------	--------------------

Description

Make color palette

Usage

```
color_make(...)
```

compose	Arranges shapes in m on layout in km according to membs vector
---------	--

Description

Arranges shapes in m on layout in km according to membs vector

Usage

```
compose(m, membs, km, bysize = TRUE, K = 0.05, noverlap = TRUE)
```

cosf	<i>Cosine</i>
------	---------------

Description

Cosine

Usage

```
cosf(v, u = east(dct(v)))
```

cotf	<i>Cotangent</i>
------	------------------

Description

Cotangent

Usage

```
cotf(v, u = east(dct(v)))
```

cscf	<i>Cosecant</i>
------	-----------------

Description

Cosecant

Usage

```
cscf(v, u = east(dct(v)))
```

deg2rad	<i>Degrees to radians</i>
---------	---------------------------

Description

Degrees to radians

Usage

```
deg2rad(d)
```

draw_seg	<i>Draw seg</i>
----------	-----------------

Description

Draw seg

Usage

draw_seg(p0, p1, lty = 2, ...)

draw_vec	<i>Draw vectors</i>
----------	---------------------

Description

Draw vectors

Usage

draw_vec(v, o = 0(dct(v)), ...)

dwise	<i>vapply wrapper for 'column'-wise operations</i>
-------	--

Description

vapply wrapper for 'column'-wise operations

Usage

dwise(m0, f_, ..., m1 = NULL, reduce = FALSE, r = NULL)

east	<i>East axis</i>
------	------------------

Description

East axis

Usage

east(d = 2)

interpolate	<i>Interplate points</i>
-------------	--------------------------

Description

Interplate points

Usage

```
interpolate(P, poly = TRUE, iter = 1)
```

iszero	<i>Is The Zero</i>
--------	--------------------

Description

Is The Zero

Usage

```
iszero(v)
```

is_ccw	<i>Is polygon P stored in counterclockwise or clockwise order? TODO: generalize to arbitrary planes</i>
--------	---

Description

Is polygon P stored in counterclockwise or clockwise order? TODO: generalize to arbitrary planes

Usage

```
is_ccw(P)
```

is_plist	<i>Check if pl is list(x, y)</i>
----------	------------------------------------

Description

Check if pl is list(x, y)

Usage

```
is_plist(pl)
```

lfilter	<i>Filter list</i>
---------	--------------------

Description

Filter list

Usage

```
lfilter(l, pred)
```

lineclip	<i>Line-polygon intersection</i>
----------	----------------------------------

Description

Line-polygon intersection

Usage

```
lineclip(P, p1, p0 = c(0, 0), scalars = FALSE)
```

lmap	<i>Apply linear mapping</i>
------	-----------------------------

Description

Apply linear mapping

Usage

```
lmap(v, map)
```

north	<i>North axis</i>
-------	-------------------

Description

North axis

Usage

```
north(d = 2)
```

one	<i>The One</i>
-----	----------------

Description

The One

Usage

one()

pclear	<i>Push m in the direction of v until $m \cap X = \emptyset$</i>
--------	---

Description

Push m in the direction of v until $m \cap X = \emptyset$

Usage

pclear(P, X, v = C(P), vector = FALSE)

pintersect	<i>Polygon intersection</i>
------------	-----------------------------

Description

Polygon intersection

Usage

pintersect(P0, P1)

position	<i>Position object by translation and rotation</i>
----------	--

Description

Position object by translation and rotation

Usage

position(m, v = vdif(0(dct(m)), C(m)), t = theta(v, axis_major(m)))

pushout	<i>Push P along v until clear of X</i>
---------	--

Description

Push P along v until clear of X

Usage

```
pushout(P, X, v)
```

rad2deg	<i>Radians to degrees</i>
---------	---------------------------

Description

Radians to degrees

Usage

```
rad2deg(r)
```

random	<i>Create n random elements of dimensionality d with the given distribution P</i>
--------	---

Description

Create n random elements of dimensionality d with the given distribution P

Usage

```
random(n = 10, d = 2, P = function(n) runif(n, -1, 1))
```

resize	<i>Resize object</i>
--------	----------------------

Description

Resize object

Usage

```
resize(v, S = NULL, abs = FALSE)
```

rotate	<i>Rotate object</i>
--------	----------------------

Description

Rotate object

Usage

```
rotate(m, theta = zero(), cw = FALSE)
```

rotation	<i>Rotation</i>
----------	-----------------

Description

Rotation

Usage

```
rotation(theta)
```

secf	<i>Secant</i>
------	---------------

Description

Secant

Usage

```
secf(v, u = east(dct(v)))
```

sinf	<i>Sine</i>
------	-------------

Description

Sine

Usage

```
sinf(v, u = east(dct(v)))
```

smul	(v, s) -> v * s
------	-------------------

Description

(v, s) -> v * s

Usage

smul(v, S)

south	West axis
-------	-----------

Description

West axis

Usage

south(d = 2)

s_v_op	<i>Low-level workload functions. The functions in this file are prime candidates for implementation in Rcpp/RcppParallel/CUDA Virtually all operations in this package use this functions. The order of computation is defined in 'storage.R' file. s -> v</i>
--------	---

Description

Low-level workload functions. The functions in this file are prime candidates for implementation in Rcpp/RcppParallel/CUDA Virtually all operations in this package use this functions. The order of computation is defined in 'storage.R' file. s -> v

Usage

s_v_op(s0, f_, d)

tanf	<i>Tangent</i>
------	----------------

Description

Tangent

Usage

```
tanf(v, u = east(dct(v)))
```

theta	<i>Angle between two vectors</i>
-------	----------------------------------

Description

Angle between two vectors

Usage

```
theta(v, u = east(dct(v)), cw = FALSE, deg = FALSE)
```

to_l	<i>Transform object to list(x, y) format</i>
------	--

Description

Transform object to list(x, y) format

Usage

```
to_l(m)
```

to_m	<i>Extract object from list(x, y) or list(list(x, y)) format</i>
------	--

Description

Extract object from list(x, y) or list(list(x, y)) format

Usage

```
to_m(l, combine = FALSE)
```

translate	<i>Translate object</i>
-----------	-------------------------

Description

Translate object

Usage

translate(v, u)

vadd	$(v, u) \rightarrow v + u$
------	----------------------------

Description

$(v, u) \rightarrow v + u$

Usage

vadd(v, u)

vaxis	<i>Alias for A</i>
-------	--------------------

Description

Alias for A

Usage

vaxis(...)

vdif	$(v, u) \rightarrow v - u$
------	----------------------------

Description

$(v, u) \rightarrow v - u$

Usage

vdif(v, u)

<i>vdist</i>	<i>Distance</i>
--------------	-----------------

Description

Distance

Usage

`vdist(v, u = 0(), p = 2)`

<i>vhead</i>	<i>Vector from C(m) to the furthest point in m</i>
--------------	--

Description

Vector from C(m) to the furthest point in m

Usage

`vhead(m, o = C(m), scalar = FALSE)`

<i>vinnerp</i>	<i>Inner product</i>
----------------	----------------------

Description

Inner product

Usage

`vinnerp(v, u)`

<i>vnorm</i>	<i>Norms</i>
--------------	--------------

Description

Norms

Usage

`vnorm(v, p = 2)`

vorth	<i>Vector normal</i>
-------	----------------------

Description

Vector normal

Usage

```
vorth(v, cw = FALSE, a = 1)
```

```
normal(...)
```

vproj	<i>Vector projection</i>
-------	--------------------------

Description

Vector projection

Usage

```
vproj(v, u, scalar = FALSE)
```

vrej	<i>Vector rejection</i>
------	-------------------------

Description

Vector rejection

Usage

```
vrej(v, u, scalar = FALSE)
```

vunit	<i>Unit vector</i>
-------	--------------------

Description

Unit vector

Usage

```
vunit(v)
```

vv_s_op	$(v,u) \rightarrow s$
---------	-----------------------

Description

$(v,u) \rightarrow s$

Usage

vv_s_op(v0, v1, f_)

vv_v_op	$(v,u) \rightarrow v$
---------	-----------------------

Description

$(v,u) \rightarrow v$

Usage

vv_v_op(v0, v1, f_)

vv_w_op	$(v,u) \rightarrow w$
---------	-----------------------

Description

$(v,u) \rightarrow w$

Usage

vv_w_op(v0, v1, f_, d)

vwise	<i>vapply wrapper for 'row'-wise operations</i>
-------	---

Description

vapply wrapper for 'row'-wise operations

Usage

vwise(m0, f_, reduce = FALSE, ..., m1 = NULL, r = NULL)

v_s_op	$v \rightarrow s$
--------	-------------------

Description

$v \rightarrow s$

Usage

v_s_op(v0, f_)

v_v_op	$v \rightarrow v$
--------	-------------------

Description

$v \rightarrow v$

Usage

v_v_op(v0, f_)

v_w_op	$v \rightarrow w$
--------	-------------------

Description

$v \rightarrow w$

Usage

v_w_op(v0, f_, d)

west	<i>South axis</i>
------	-------------------

Description

South axis

Usage

west(d = 2)

zero	<i>The Zero</i>
------	-----------------

Description

The Zero

Usage

zero()

Index

A, [3](#)

bsize, [3](#)

C, [3](#)

center, [3](#)

chart, [4](#)

color_alpha, [4](#)

color_make, [4](#)

compose, [4](#)

cosf, [5](#)

cotf, [5](#)

cscf, [5](#)

deg2rad, [5](#)

draw_seg, [6](#)

draw_vec, [6](#)

dwise, [6](#)

east, [6](#)

interpolate, [7](#)

is_ccw, [7](#)

is_plist, [7](#)

iszero, [7](#)

lfilter, [8](#)

lineclip, [8](#)

lmap, [8](#)

normal(vorth), [16](#)

north, [8](#)

one, [9](#)

pclear, [9](#)

pintersect, [9](#)

position, [9](#)

pushout, [10](#)

rad2deg, [10](#)

random, [10](#)

resize, [10](#)

rotate, [11](#)

rotation, [11](#)

s_v_op, [12](#)

secf, [11](#)

sinf, [11](#)

smul, [12](#)

south, [12](#)

tanf, [13](#)

theta, [13](#)

to_l, [13](#)

to_m, [13](#)

translate, [14](#)

v_s_op, [18](#)

v_v_op, [18](#)

v_w_op, [18](#)

vadd, [14](#)

vaxis, [14](#)

vdif, [14](#)

vdist, [15](#)

vhead, [15](#)

vinnerp, [15](#)

vnorm, [15](#)

vorth, [16](#)

vproj, [16](#)

vrej, [16](#)

vunit, [16](#)

vv_s_op, [17](#)

vv_v_op, [17](#)

vv_w_op, [17](#)

vwise, [17](#)

west, [18](#)

zero, [19](#)