Package 'PlotRegionHighlighter'

February 19, 2015

Type Package

Title Creates an envelope that surrounds a set of points plotted in a two dimensional space.

Version 1.0

Date 2013-04-04

Author Elliot Noma

Maintainer Elliot Noma <noma@garrettassetmanagement.com>

Description Creates an envelope around a set of plotted points. The envelope is compact with a boundary that is continuous, smooth and convex. Each point is represented as a circle and the circles and connecting lines are the solution to the multiple pulley problem. This method can be used to highlight regions in a two-dimensional space.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-05 18:25:16

R topics documented:

PlotRegionHighlighter-package *Creates an envelope that surrounds a set of points plotted in two dimensions*

Description

Create an envelope surrounding a set of points in a two-dimensional space. The shape in the union of a polygon and circles surrounding each point. The polygon is determined using an extension of methods to determine the tangent line to two circles and is the solution to the multiple pulley problem. The points can used to highlight a region in a two-dimensional space.

Details

Package:	PlotRegionHighlighter
Type:	Package
Version:	1.0
Date:	2013-04-04
License:	GPL-2

The generateEnvelope function is called with a two-column matrix with each row containing the xy coordinates for each point. Along with vector of radii for the circles surrounding each point, the function generates a list of points defining the envelope surrounding the set of points. The envelope is computed as if it were specifying a pulley that passed around the circles with a minimum perimenter and minimum area for a convex shape containing all the circles. The algorithm uses formulas for calculating the set of lines that is tangent to a pair of circles.

Author(s)

Elliot Noma

Maintainer: Elliot Noma <noma@garrettassetmanagement.com>

References

http://en.wikipedia.org/wiki/Belt_problem

arcAngle *The counterclockwise arc of two points relative to a third*

Description

Compute the beginning and ending angles for an arc in radians relative to a third point

Usage

```
arcAngle(x, y, center)
```

Arguments

x	the xy coordinates of the first point
y	the xy coordinates of the second point that is counterclockwise from the first
center	the xy coordinates of the reference point

Value

a two-item numeric vector containing the angle in radians of the two points relative to the reference point

Author(s)

Elliot Noma

---

centersLineSegmentIntersections

Determine if a line intersects one or more line segments

Description

Determine if a line intersects one or more line segments. The segments are all pairwise combinations of points from a set of points

Usage

```
centersLineSegmentIntersections(tangent, centers)
```

Arguments

tangent	a three-item numeric vector containing the coefficients, a, b and c for a line in two-dimensions defined by ax + by + c = 0
centers	a two-column numeric matrix containing the x and y coordinates for a set of points to be used as the endpoints for the line segments

Value

a Boolean variable. FALSE if the line intersects any of the segments. TRUE otherwise

Author(s)

Elliot Noma

createCircle *A set of points defining a circle or an arc*

Description

A set of xy coordinates defining a circle or an arc on the circle

Usage

```
createCircle(center, r, n = 40, begin = 0, end = 2 * pi)
```

Arguments

center	a two-item vector of the x and y coordinates of the circle center
r	a numeric value defining the radius of the circle
n	a numeric value defining the number of points on the circle
begin	a numeric value for the starting angle of the circle or arc in radians
end	a numeric value for the ending angle of the circle or arc in radians

Value

a two-column matrix of xy coordinates for points on the circle or arc

Author(s)

Elliot Noma

drawArc *Compute the points on an arc starting with the coordinates of the starting and ending points relative to the circle center*

Description

Compute the points on an arc based on the coordinates of the starting and ending points

Usage

```
drawArc(x, y, center, r, ...)
```

Arguments

x	a two-item numeric vector for the xy coordinates of the starting point relative to the center point
y	a two-item numeric vector for the xy coordinates of the ending point relative to the center point
center	a two-item numeric vector for the xy coordinates of the center point for the arc
r	a numeric value for the radius of the circle
...	Optional argment to specify the number of points on the arc

Value

a two-column matrix of xy coordinates for points along the arc

Author(s)

Elliot Noma

---

envelopeArea_and_Perimeter

Calculate the area and perimeter of the envelope

Description

Calculate the area and perimeter of the envelope

Usage

```
envelopeArea_and_Perimeter(segments, centers, r)
```

Arguments

segments	The tangent points output by the generateEnvelope function
centers	a numeric matrix with each row containing the coordinates for a point with the x and y coordinates as the two columns.
r	a numeric vector containing the radii for each point

Details

Calculations are done in the units of the graph

Value

a numeric vector. The area is the first value and the perimeter is the second value

Author(s)

Elliot Noma

References

http://en.wikipedia.org/wiki/Belt_problem

Examples

```
############################################################
# plot

plotCircles <- function(center, r, color="red", ...)
{
a <- createCircle(center, r, ...)
grid.polygon(x = a[,1], y = a[,2], gp=gpar(col=color, lwd=2))

a
}

ncircles <- sample(3:7,1)
centers <- matrix(runif(2*ncircles, min=.2, max=.8), byrow=TRUE, ncol=2)
r <- runif(ncircles,min=.10, max=.20)

envelope <- generateEnvelope(centers, r)
print(envelope$tangent_Points)

require(grid)
grid.newpage()
colors <- rainbow(ncircles * 3 + 3)
for (i in 1:ncircles) circles<- plotCircles(centers[i,], r[i], color=colors[i])
grid.text(1:ncircles, centers[,1], centers[,2])

# plot the envelope containing the circles

envelopeXY <- envelope$envelopeXY
segments <- envelope$tangent_Points

grid.lines(envelopeXY[,1], envelopeXY[,2], gp=gpar(col="orange", lwd=5), default.units="npc")
grid.points(segments[,"x"], segments[,"y"], pch=16, gp=gpar(col="red"), default.units="npc")

# calculate the area and perimeter of the envelope

envelopeStats <- envelopeArea_and_Perimeter(segments, centers, r)

cat("envelope area = ", envelopeStats["area"], " perimeter = ", envelopeStats["perimeter"],"\n")
cat("circle radii = ", r, "\n")
cat("circle area = ", pi * r^2, " = ", sum(pi * r^2), "\ncircle perimenter = ",  2 * pi * r, " = ",  2 * pi * sum(r)
```

findExteriorTangents *Determine the set of tangents to a pair of circles that do not intersect other circles or pass between circles*

Description

Determine the points of tangency for lines tangent to a pair of circles. Keep only tangents that do not intersect other circles or pass between circles

Usage

```
findExteriorTangents(center, r, i, j, rrange = c(-1, 1), krange = c(1, -1))
```

Arguments

center	a two-column numeric matrix of xy coordinates for center points for the set of circles
r	a numeric vector of the radii of the circles
i	a numeric pointer to a row in the matrix of xy coordinates for the centers. This points to the first circle of the pair
j	a numeric pointer to a row in the matrix of xy coordinates for the centers. This points to the second circle of the pair
rrange	a numeric value or two item numeric vector defining the tangents to be computed. Values are -1, 1, or c(1,-1). -1 returns the tangents that cross between the circles. 1 returns those that do not cross between the circles
krange	a numeric value or two item numeric vector defining the tangents to be computed. Values are -1, 1, or c(1,-1).

Value

a matrix with pairs of rows containing the starting and ending tangent points for line segments. Rows are identified by the circle on which they are located and the circle on which the other end point lies.

Author(s)

Elliot Noma

References

http://en.wikipedia.org/wiki/Belt_problem

`generateEnvelope` *Generate an minimum-perimeter envelope that surrounds a set of points in a graph*

Description

The generateEnvelope function is called with the coordinates for points in the envelope along with a vector of radii for the circles surrounding each point. The function generates a list of points defining the envelope surrounding the entire set of points. These coordinates may be displayed using either the line or polygon graphics commands

Usage

```
generateEnvelope(centers, r, ...)
```

Arguments

`centers`	a numeric matrix with each row containing the xy coordinates for each point.
`r`	a numeric vector containing the radius of each point
`...`	additional parameters such as the number of points to return when defining the arcs in the envelope

Value

`envelopeXY`	The x and y coordinates for the points defining the envelope
`tangent_Points`	The tangent points defining the transition between arcs and lines along with the angle relative to the centorid of the input points

Author(s)

Elliot Noma

References

http://en.wikipedia.org/wiki/Belt_problem

Examples

```
############################################################
# plot

plotCircles <- function(center, r, color="red", ...)
{
a <- createCircle(center, r, ...)
grid.polygon(x = a[,1], y = a[,2], gp=gpar(col=color, lwd=2))

a
}
```

```
ncircles <- sample(3:7,1)
centers <- matrix(runif(2*ncircles, min=.2, max=.8), byrow=TRUE, ncol=2)
r <- runif(ncircles,min=.10, max=.20)

envelope <- generateEnvelope(centers, r)
print(envelope$tangent_Points)

require(grid)
grid.newpage()
colors <- rainbow(ncircles * 3 + 3)
for (i in 1:ncircles) circles<- plotCircles(centers[i,], r[i], color=colors[i])
grid.text(1:ncircles, centers[,1], centers[,2])

# plot the envelope containing the circles

envelopeXY <- envelope$envelopeXY
segments <- envelope$tangent_Points

grid.lines(envelopeXY[,1], envelopeXY[,2], gp=gpar(col="orange", lwd=5), default.units="npc")
grid.points(segments[,"x"], segments[,"y"], pch=16, gp=gpar(col="red"), default.units="npc")

# calculate the area and perimeter of the envelope

envelopeStats <- envelopeArea_and_Perimeter(segments, centers, r)

cat("envelope area = ", envelopeStats["area"], " perimeter = ", envelopeStats["perimeter"],"\n")
cat("circle radii = ", r, "\n")
cat("circle area = ", pi * r^2, " = ", sum(pi * r^2), "\ncircle perimenter = ",  2 * pi * r, " = ",  2 * pi * sum(r)

################################################
# plot envelopes around two randomly generated set of points
require(grid)
grid.newpage()
ncircles <- sample(10:25,1)
centers <- matrix(runif(2*ncircles, min=.2, max=.5), byrow=TRUE, ncol=2)
r <- rep(0.1, ncircles)

envelopeXY <- generateEnvelope(centers, r)$envelopeXY
grid.polygon(envelopeXY[,1], envelopeXY[,2], gp=gpar(fill="pink", col="transparent", lwd=5), default.units="npc
grid.points(centers[,1], centers[,2], pch=16, gp=gpar(col="black", cex=1.5), default.units="npc")

ncircles <- sample(10:20,1)
centers <- matrix(runif(2*ncircles, min=.6, max=.8), byrow=TRUE, ncol=2)
r <- rep(0.025, ncircles)

grid.points(centers[,1], centers[,2], pch=16, gp=gpar(col="blue", cex=1.5), default.units="npc")

envelopeXY <- generateEnvelope(centers, r)$envelopeXY
grid.lines(envelopeXY[,1], envelopeXY[,2], gp=gpar(col="blue", lwd=5), default.units="npc")
```

`tangentLine` *Compute the coefficients for the line tangent to two circles*

Description

Compute the coefficients for the line ax + by + c = 0 which is tangent to circles with centers at c1 and c2 with radii r1 and r2. Call this function varying k = -1 and +1 and r1 = r1 and -r1 to calculate the lines. There can be up to four distinct lines that are tangent to both circles.

Usage

```
tangentLine(c1, c2, r1, r2, k = 1)
```

Arguments

`c1`	a 2-item numeric vector containing the x and y coordinates for the first circle
`c2`	a 2-item numeric vector containing the x and y coordinates for the second circle
`r1`	a numeric value for the radius of the first circle
`r2`	a numeric value for the radius of the second circle
`k`	k=1 returns the coefficients for one of the tangent lines and k=-1 returns the coefficients for the second tangent line

Details

There are a maximum of four tangent lines for a pair of circles. These can be obtained by setting argument r1 to either r1 or -r1 and argument k to -1 or 1.

Value

a 3-item vector containing the values for a, b and c in the equation ax + by + c = 0. Returns NULL if there is not tangent line for the input parameters

Author(s)

Elliot Noma

References

http://en.wikipedia.org/wiki/Belt_problem

Examples

```
############################################################
# plot

plotCircles <- function(center, r, color="red", ...)
{
a <- createCircle(center, r, ...)
grid.polygon(x = a[,1], y = a[,2], gp=gpar(col=color, lwd=2))

a
}


require(grid)
grid.newpage()
ncircles <- 2
centers <- matrix(runif(4, min=.2, max=.8), byrow=TRUE, ncol=2)
r <- runif(ncircles,min=.10, max=.20)

colors <- rainbow(ncircles * 3 + 3)
for (i in 1:ncircles) circles<- plotCircles(centers[i,], r[i], color=colors[i])
grid.text(1:ncircles, centers[,1], centers[,2])

ii <- 0
for (r0 in r[1] * c(1,-1))
for (k in c(1,-1)) {
ii <- ii + 1
tangent <- tangentLine(centers[1,], centers[2,], r0, r[2], k=k)  # compute coefficients for the tangent line, if N

if (!is.na(tangent["a"]))
grid.abline(-tangent["c"] / tangent["b"], -tangent["a"] / tangent["b"], gp=gpar(col="blue", lwd=ii), units="npc"
}
```

Index