Package 'vrmlgen'

February 20, 2015

Type Package

Title Generate 3D visualizations for data exploration on the web

Version 1.4.9

Date 2013-12-08

Author Enrico Glaab

Maintainer Enrico Glaab <e.glaab@cs.nott.ac.uk>

Suggests misc3d

Description vrmlgen creates 3D scatter and bar plots, visualizations of 3D meshes, parametric functions and height maps in web-formats like the Virtual Reality Markup Language (VRML, filetype .wrl) and the LiveGraphics3D format.

License GPL (>= 2)

URL http://www.cs.nott.ac.uk/~egg/vrmlgen

LazyLoad yes

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-09 16:17:26

R topics documented:

`vrmlgen-package` *Create 3D data plots, charts and graphs as VRML files*

Description

vrmlgen translates 3D data, i.e. 3D point clouds, meshes or surfaces, into a visual representation in the Virtual Reality Markup Language (VRML, file type .wrl) or alternatively, in the LiveGraphics3D format. Apart from providing access to low-level plotting functions (points, lines, text, etc.) through a unified interface, several higher-level visualization methods are supported, including 3D scatter plots, meshes, bar charts, graphs and density estimation contour surfaces.

Details

Package:	vrmlgen
Type:	Package
Version:	1.4.7
Date:	Date: 2011-02-09
License:	GPL (>= 2)
LazyLoad:	yes
URL:	http://www.cs.nott.ac.uk/~egg/vrmlgen

Author(s)

Enrico Glaab

e.glaab@cs.nott.ac.uk

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# create 33x3 matrix with random 3D input data
mat <- matrix(runif(99, 0, 3), ncol = 3)

# create random class assignment vector with three classes
y <- round(runif(33, 0, 2))
y <- ifelse(y == 0, "class 1", ifelse(y == 1, "class 2", "class 3"))

# create ouput using numbers from 1 to length(y) as metalabels
cloud3d(mat, labels = y, metalabels = 1:length(y), col.axis = "black", col.lab = "blue",
col.bg = "white")

setwd(curdir)
```

`axis3d` *Draw labeled coordinate axes in a 3D-scene*

Description

axis3d draws 3 labeled, orthogonal coordinate axes in a 3D-scene in the VRML- or Livegraphics3D-format. Must be called after vrml.open() or lg3d.open() and before vrml.close() or lg3d.close().

Usage

```
axis3d(lab.axis = c("X-axis", "Y-axis", "Z-axis"),
       filename = NULL, type = "vrml", col.lab = "black",
       col.axis = "black", cex.lab = 1, local_scale = 1,
       global_scale = 1)
```

Arguments

`lab.axis`	a vector of size 3 containing the axis labels
`filename`	filename of the generated output file
`type`	the output type ("vrml" or "lg3d"), this will be set automatically if mesh3d is called after vrml.open() or lg3d.open()
`col.lab`	a vector of colors specifying the axis labels
`col.axis`	color of the axis
`cex.lab`	scaling factor for axis label font size
`local_scale`	a numerical scaling factor to increase/decrease the size of the plotted axes
`global_scale`	a numerical scaling factor applied to all objects in the 3D scene

Details

axis3d adds colored and labeled coordinate axes to an already existing 3D- scene in the VRML- or Livegraphics3D-format. To create a standard scatter plot, bar plot or 3D mesh visualization, which already contains coordinate axes, the higher-level plotting functions cloud3d, bar3d and mesh3d provide more convenient alternatives. axis3d can only be applied within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function and closed using the vrml.close() or lg3d.close() function.

Value

The function is used for its side-effect (writing coordinate axes to a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

text3d, lines3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# start a new VRML environment
vrml.open(file = "axis_example.wrl", scale = 5,
          html.embed = "axis.html")

# draw a coordinate system with dark blue axes
# and dark gray axis labels
axis3d(col.lab = "darkgray", col.axis = "darkblue")

# plot random data points within the coordinate system
# (increase point size by a factor of 2)
mat <- matrix(runif(99, 0, 1), ncol = 3)
points3d(mat, col = "red", scale = 2)

# close the VRML environment and write the output file
vrml.close()
```

```
# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "axis.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "axis.html"), sep = ""))
}

setwd(curdir)
```

---

bar3d *Draw a 3D bar chart*

Description

bar3d creates 3D barplots and height map visualizationsin the VRML- or Livegraphics3D-format.

Usage

```
bar3d(data, row.labels = rownames(data),
      col.labels = colnames(data), metalabels = NULL,
      filename = "out.wrl", type = "vrml", space = 0.5,
      cols = rainbow(length(as.matrix(data))),
      rcols = NULL, ccols = NULL, origin = c(0, 0, 0),
      scalefac = 4, lab.axis = c("X-axis", "Y-axis", "Z-axis"),
      lab.vertical = FALSE, col.axis = "black",
      showaxis = TRUE, autoscale = TRUE,
      ignore_zeros = TRUE, col.lab = "black",
      col.bg = "white", cex.lab = 1, cex.rowlab = 1,
      cex.collab = 1, htmlout = NULL, hwidth = 1200,
      hheight = 800, showlegend = TRUE,
      vrml_navigation = "EXAMINE", vrml_transparency = 0,
      vrml_fov = 0.785, vrml_pos = rep(scalefac + 4, 3),
      vrml_dir = c(0.19, 0.45, 0.87, 2.45),
      lg3d_ambientlight = 0.5)
```

Arguments

data	a numerical matrix with 3 columns and n rows
row.labels	a vector containing the row labels (strings or numbers)
col.labels	a vector containing the column labels (strings or numbers)
metalabels	a vector of strings or numbers containing optional metalabels for the rows which can be accessed by hovering the mouse over a data point in the plot (in VRML)
filename	filename of the generated output file

type	the output type ("vrml" or "lg3d"), this will be set automatically if mesh3d is called after vrml.open() or lg3d.open()
space	space between single bars in a bar plot (as a fraction of the average bar width)
cols	a vector of colors to visualize different classes among the data points. The number of colors should either be one or at least as large as the number of different labels (i.e. length(col) >= length(unique(labels))
rcols	specifies the bar colors per row of the input matrix (the cols- and ccols paramter can remain undefined). The number of colors must correspond to the numbers of rows of the data matrix.
ccols	specifies the bar colors per column of the input matrix (the cols- and rcols-parameter can remain undefined). The number of colors must correspond to the numbers of columns of the data matrix.
origin	a vector of length 3 specifying the coordinates of the plot origin (can be used to translate the plot)
scalefac	a scaling factor for the size of the entire plot
lab.axis	a vector of size 3 containing the axis labels
col.axis	color of the axis
lab.vertical	if TRUE, the data labels will be plotted in vertical instead of horizontal orientation
showaxis	if FALSE, the coordinate axes are hidden in the output
autoscale	if TRUE, the data is automatically scaled to fill the volume between the axes
ignore_zeros	if TRUE, for height values equal to zero no bar will be plotted
col.lab	a vector of colors for the axis labels
col.bg	background color
cex.lab	scaling factor for axis label font size
cex.rowlab	scaling factor for matrix row labels font size
cex.collab	scaling factor for matrix column labels font size
htmlout	a filename for generating an HTML-file to embed the output
hwidth	width of the embedded visualization in the HTML-output
hheight	height of the embedded visualization in the HTML-output
showlegend	if TRUE, a data legend will be added to the 3D plot
vrml_navigation	type of mouse navigation in the VRML file, can be "EXAMINE", "WALK", "SLIDE", "FLY" or "PAN" (VRML only)
vrml_transparency	a number between 0 and 1 specifying the transparency level of plotted objects (VRML only)
vrml_fov	a scalar defining the field of view angle in the VRML file in radians (VRML only)
vrml_pos	a vector of size 3 corresponding to the position of the viewpoint (VRML only)

`vrml_dir` a vector of size 4 specifying the viewing direction (first 3 components) and the rotation of the camera around the direction vector (last component in radians, VRML only)

`lg3d_ambientlight` ambient light gray level (value between 0 and 1, LG3D only)

Details

bar3d generates 3D barplots and height map visualizationsin the VRML- or Livegraphics3D-format. The plots can automatically be embedded in a HTML-file to allow users to inspect the data interactively and from different 3D- perspectives and scalings on a webpage.

To add additional shapes and objects to the final 3D scene, this plotting function can also be called within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function. In this case, cloud3d will inherit all global parameters set in the vrml.open- or lg3d.open-function (e.g. filename, type, htmlout, etc.) and the user does not need to specify these options anymore.

Value

The function is used for its side-effect (creating a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

mesh3d, cloud3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# Height map visualization of geographical data
# with colors corresponding to different height levels
colpalette <- terrain.colors(max(volcano)-min(volcano)+1)
cols <- colpalette[as.vector(t(volcano))-min(volcano)+1]
bar3d(volcano, col.bg = "white", cols = cols,
      type = "vrml", space = 0, showaxis = FALSE,
      filename = "volcano.wrl", htmlout = "volcano.html")
```

```
# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "volcano.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "volcano.html"), sep = ""))
}

# bar plot for a data matrix with numerical metalabels
mat <- matrix(9:1, nrow = 3)
rownames(mat) <- paste('row',1:3)
colnames(mat) <- paste('col',1:3)
bar3d(mat, space = 1, metalabels = 1:9, col.bg = "white",
      col.axis = "blue", col.lab = "black", type = "vrml",
      filename = "barplot.wrl", htmlout = "barplot.html")

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "barplot.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "barplot.html"), sep = ""))
}

# Height map visualization of very rough and highly
# simplified topographic data for the United Kingdom

data(uk_topo)

bar3d(uk_topo, autoscale = FALSE, cols = "blue",
     space = 0, showaxis = FALSE, filename = "uk_topo.wrl",
     htmlout = "uk_topo.html")

setwd(curdir)
```

---

bc_classes *Breast cancer microarray example dataset - class labels*

Description

The class labels for the breast cancer microarray example dataset

Usage

```
data(bc_classes)
```

Details

This vector contains the class labels for the breast cancer microarray example dataset (see bc_dat help page)

Source

The original data set is publicly available at http://www.ebi.ac.uk/microarray-as/ae under the accession number: E-TABM-576

---

bc_dat *Breast cancer microarray example dataset*

Description

Example of a 3D input structure to illustrate VRMLGen's visualization methods

Usage

```
data(bc_dat)
```

Details

This dataset was obtained by applying an Independent Component Analysis to a public breast cancer microarray data set. The resulting three independent component vectors can be visualized in 3D and are used to illustrate some of VRMLGen's features.

Source

The original data set is publicly available at http://www.ebi.ac.uk/microarray-as/ae under the accession number: E-TABM-576

---

c60coords *Atom coordinates of a C60 molecule - example dataset*

Description

Example of a 3D input structure to illustrate VRMLGen's visualization methods

Usage

```
data(c60coords)
```

Details

This dataset contains the 3D-coordinates of a C60 molecule (known as fullerene) used as example input to demonstrate VRMLGen's visualization features.

References

The molecular structure was created with the open-source molecular modeling sofware BALL, see www.ballview.org.

`cloud3d` *Draw a 3D scatter plot*

Description

cloud3d creates dynamic 3D scatter plot visualizations in the VRML- or Livegraphics3D-format.

Usage

```
cloud3d(x, y = NULL, z = NULL, labels = rownames(data),
        filename = "out.wrl", type = "vrml",
        pointstyle = c("s", "b", "c"), metalabels = NULL,
        hyperlinks = NULL, cols = rainbow(length(unique(labels))),
        scalefac = 4, autoscale = "independent",
        lab.axis = c("X-axis", "Y-axis", "Z-axis"),
        col.axis = "black", showaxis = TRUE, col.lab = "black",
        col.bg = "white", cex.lab = 1, htmlout = NULL,
        hwidth = 1200, hheight = 800, showlegend = TRUE,
        vrml_navigation = "EXAMINE", vrml_showdensity = FALSE,
        vrml_fov = 0.785, vrml_pos = rep(scalefac + 4, 3),
        vrml_dir = c(0.19, 0.45, 0.87, 2.45),
        vrml_transparency = 0, lg3d_ambientlight = 0.5)
```

Arguments

`x`	a 3-column numeric matrix of coordinates or a numeric vector of x-coordinates (not used if parameter obj_infile or xfun is specified)
`y`	a numeric vector of y-coordinates (only needed if x is a vector)
`z`	a numeric vector of z-coordinates (only needed if x is a vector)
`labels`	a vector of size n containing optional class labels (strings or numbers) for the data points
`filename`	filename of the generated output file
`type`	the output type ("vrml" or "lg3d"), this will be set automatically if mesh3d is called after vrml.open() or lg3d.open()
`pointstyle`	a vector of point style types. Possible types are "s" for spheres, "b" for boxes and "c" for cone. To draw all points in the same style, use a vector of size 1, otherwise the number of used point styles must be greater or equal to the number of different labels (i.e. length(pointstyles) >= length(unique(labels)))
`metalabels`	a vector of strings or numbers containing optional metalabels for the rows which can be accessed by hovering the mouse over a data point in the plot (in VRML)

hyperlinks	a vector of strings specifying hyperlinks that will be triggered, when the user clicks on the corresponding datapoint
cols	a vector of colors to visualize different classes among the data points. The number of colors should either be one or at least as large as the number of different labels (i.e. length(col) >= length(unique(labels))
scalefac	a numerical scaling factor to increase/decrease the size of the plotted 3D objects
autoscale	the scaling type. "independent" means all axes are auto-scaled independently. "equidist" means all axes are scaled by the same factor and "equicenter" additionally positions the drawn object in the center. "none" means no scaling is applied
lab.axis	a vector of size 3 containing the axis labels
col.axis	color of the axis
showaxis	if FALSE, the coordinate axes are hidden in the output
col.lab	a vector of colors specifying the axis labels
col.bg	background color
cex.lab	scaling factor for axis label font size
htmlout	a filename for generating an HTML-file to embed the VRML-output
hwidth	width of the embedded visualization in the HTML-output
hheight	height of the embedded visualization in the HTML-output
showlegend	if TRUE, a data legend will be drawn based on the group labels
vrml_navigation	type of mouse navigation in the VRML file, can be "EXAMINE", "WALK", "SLIDE", "FLY" or "PAN" (VRML only)
vrml_showdensity	if TRUE, semi-transparent density estimation contour surfaces will be drawn to identify regions of high data density (requires misc3d-package)
vrml_transparency	a number between 0 and 1 specifying the transparency level of plotted objects (VRML only)
vrml_fov	a scalar defining the field of view angle in the VRML file in radians (VRML only)
vrml_pos	a vector of size 3 corresponding to the position of the viewpoint (VRML only)
vrml_dir	a vector of size 4 specifying the viewing direction (first 3 components) and the rotation of the camera around the direction vector (last component in radians, VRML only)
lg3d_ambientlight	ambient light gray level (value between 0 and 1, LG3D only)

Details

cloud3d creates dynamic 3D-scatterplots with optional features like density estimation contour surfaces and hyperlinks for each datapoint in the VRML- or Livegraphics3D-format. The plots can

automatically be embedded in a HTML-file to allow users to inspect the data interactively and from different 3D-perspectives and scalings on a webpage.

To add additional shapes and objects to the final 3D scene, this plotting function can also be called within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function. In this case, cloud3d will inherit all global parameters set in the vrml.open- or lg3d.open-function (e.g. filename, type, htmlout, etc.) and the user does not need to specify these options anymore.

Value

The function is used for its side-effect (creating a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Bowman, A. W. and Azzalini, A. (2007). R package 'sm': *Nonparametric smoothing methods* (version 2.2) URL: http://www.stats.gla.ac.uk/~adrian/sm, http://azzalini.stat.unipd.it/Book_sm

Feng, D. and Tierney, L. (2009). R package 'misc3d": *Miscellaneous 3D plots* (version 0.6-1) URL: http://cran.r-project.org/web/packages/misc3d/index.htm

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

mesh3d, bar3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)


# example 1: visualize random 3D input data
mat <- matrix(runif(99, 0, 3), ncol = 3)

# create random class assignment vector with three classes
y <- round(runif(33, 0, 2))
y <- ifelse(y == 0, "class 1", ifelse(y == 1, "class 2", "class 3"))

# create ouput using numbers from 1 to length(y) as metalabels
cloud3d(mat, labels = y, metalabels = 1:length(y), col.axis = "black",
```

```
        col.lab = "blue", col.bg = "white", type = "vrml",
        filename = "result.wrl", htmlout = "result.html")

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "result.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "result.html"), sep = ""))
}


# example 2: visualization of Edgar Anderson's Iris data
# using density estimation contour surfaces (requires misc3-package!)
irismat <- iris[,1:3]
cloud3d(irismat, labels = iris[,5], vrml_showdensity = TRUE,
        col.axis = "blue", col.lab = "black", lab.axis = colnames(irismat),
        type = "vrml", filename = "result2.wrl", htmlout = "iris.html")

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "iris.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "iris.html"), sep = ""))
}

setwd(curdir)
```

`lg3d.close` *Livegraphics3D output device system*

Description

lg3d.close terminates a Livegraphics3D-environment and writes the corresponding 3D-scene to a Livegraphics3D-file.

Usage

```
lg3d.close()
```

Details

This function closes a Livegraphics3D-environment created with lg3d.open() and writes the Livegraphics3D-plot to the file and directory specified in the lg3d.open- parameters. For more details, see the description of the lg3d.open- function.

Value

The function is used for its side-effect (output of a Livegraphics3D-file in the specified directory) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

vrml.open

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)


# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).

lg3d.open(file = "c60.mat", html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black", scale = 2)

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 2)
  }
}
```

```
lg3d.close()


# show the output in a web-browser
# (Java must be enabled!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

`lg3d.open` *Livegraphics3D output device system*

Description

lg3d.open creates a new Livegraphics3D-environment in which primitive and higher-level plotting functions can be combined together to create a scene in the Livegraphics3D-format.

Usage

```
lg3d.open(filename = "out.m", col = "white", scale = 1,
          html.embed = "out.html", hwidth = 1200,
          hheight = 800, ambientlight = 0.5)
```

Arguments

`filename`	filename of the generated Livegraphics3D output file
`col`	background color
`scale`	a scaling factor for the size of the entire 3D scene
`html.embed`	a filename for generating an HTML-file to embed the VRML-output
`hwidth`	width of the embedded visualization in the HTML-output
`hheight`	height of the embedded visualization in the HTML-output
`ambientlight`	ambient light gray level (value between 0 and 1, LG3D only)

Details

This function creates a new Livegraphics3D-environment to provide access to primitive plotting functions like points3d, lines3d and text3d and to combine multiple plotting functions together (both primitive and higher-level functions). After creating the environment, all primitive and higher-level functions called afterwards will automatically inherit the higher-level settings (background color, navigation-type, etc.) specified in the lg3d.open- parameters. In order to close the environment and create the plot, the lg3d.close() function has to be called.

Value

The function is used for its side-effect (output of a Livegraphics3D-file in the specified directory) and has no return value. The htmlout-parameter can be used to embed the resulting Livegraphics3D-object in a template HTML-file.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

vrml.close

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)


# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).

lg3d.open(file = "c60.mat", html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black", scale = 2)

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 2)
  }
}
```

```
lg3d.close()


# show the output in a web-browser
# (Java must be enabled!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

`lines3d` *Draw lines in a 3D-scene*

Description

lines3d draws lines in a 3D-scene in the VRML- or Livegraphics3D-format. Must be called after vrml.open() or lg3d.open() and before vrml.close() or lg3d.close().

Usage

```
lines3d(x, y = NULL, z = NULL, col = "black", lwd = 1)
```

Arguments

x	a 3-column numeric matrix of coordinates or a numeric vector of x-coordinates
y	a numeric vector of y-coordinates (only needed if x is a vector)
z	a numeric vector of z-coordinates (only needed if x is a vector)
col	the color of the text
lwd	a numerical scaling factor specifying the line width

Details

lines3d adds lines at specified start- and end-points to a 3D-scene in the VRML- or Livegraphics3D-format. In the x,y and z coordinate parameters, the odd-numbered entries represent the line start-points and the subsequent even-numbered entries contain the corresponding end-points. lines3d can only be applied within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function and closed using the vrml.close() or lg3d.close() function.

Value

The function is used for its side-effect (writing lines to a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

text3d, points3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).


vrml.open(file = "c60.wrl", navigation = "EXAMINE",
          html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black")

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 1)
  }
}

vrml.close()

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
```

```
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

---

mesh3d *Draw a 3D-mesh*

Description

mesh3d visualizes 3D meshes and parametric functions in the VRML- or Livegraphics3D-format.

Usage

```
mesh3d(xfun = "sin(v)*cos(u)", yfun = "sin(v)*sin(u)",
       zfun = "cos(v)", param1 = "u", param2 = "v",
       range1 = c(0, 2 * pi), range2 = c(0, pi),
       size1 = 30, size2 = 30, type = "vrml", x = NULL,
       y = NULL, z = NULL, edges = NULL, obj_infile = NULL,
       filename = "out.wrl", write_obj = FALSE, cols = "red",
       scalefac = 4, autoscale = ifelse(is.null(obj_infile),
       "independent", "equicenter"),
       lab.axis = c("X-axis", "Y-axis", "Z-axis"),
       col.axis = "black", showaxis = TRUE, col.lab = "black",
       col.bg = "white", cex.lab = 1, htmlout = NULL,
       hwidth = 1200, hheight = 800,
       vrml_navigation = "EXAMINE", vrml_transparency = 0,
       vrml_fov = 0.785, vrml_pos = rep(scalefac + 4, 3),
       vrml_dir = c(0.19, 0.45, 0.87, 2.45),
       lg3d_ambientlight = 0.5)
```

Arguments

xfun	parametric function to define curves and surfaces (1. dimension)
yfun	parametric function to define curves and surfaces (2. dimension)
zfun	parametric function to define curves and surfaces (3. dimension)
param1	name of the first parameter used in parametric functions
param2	name of the second parameter used in parametric functions
range1	value range for parameter "param1"
range2	value range for parameter "param2"
size1	mesh size for the first parameter used in parametric functions
size2	mesh size for the second parameter used in parametric functions

type	the output type ("vrml" or "lg3d"), this will be set automatically if mesh3d is called after vrml.open() or lg3d.open()
x	a 3-column numeric matrix of coordinates or a numeric vector of x-coordinates (not used if parameter obj_infile or xfun is specified)
y	a numeric vector of y-coordinates (only needed if x is a vector)
z	a numeric vector of z-coordinates (only needed if x is a vector)
edges	a numerical matrix in which each row contains a sequence of vertex identifiers representing the edges of a polygonal face (vertex identifiers are integer numbers, starting with 0)
obj_infile	filename of a 3D mesh input file in obj-format (not required if edges and x,y,z or the parameters for parametric functions are specified)
filename	filename of the generated output file
write_obj	TRUE, if the output is to be created in the obj-format (requires parameters x,y,z and edges to be set)
cols	the color of the output
scalefac	a numerical scaling factor to increase/decrease the size of the plotted 3D objects
autoscale	the scaling type. "independent" means all axes are auto-scaled independently. "equidist" means all axes are scaled by the same factor and "equicenter" additionally positions the drawn object in the center. "none" means no scaling is applied
lab.axis	a vector of size 3 containing the axis labels
col.axis	color of the axis
showaxis	if FALSE, the coordinate axes are hidden in the output
col.lab	a vector of colors specifying the axis labels
col.bg	background color
cex.lab	scaling factor for axis label font size
htmlout	a filename for generating an HTML-file to embed the output
hwidth	width of the embedded visualization in the HTML-output
hheight	height of the embedded visualization in the HTML-output
vrml_navigation	type of mouse navigation in the VRML file, can be "EXAMINE", "WALK", "SLIDE", "FLY" or "PAN" (VRML only)
vrml_transparency	a number between 0 and 1 specifying the transparency level of plotted objects (VRML only)
vrml_fov	a scalar defining the field of view angle in the VRML file in radians (VRML only)
vrml_pos	a vector of size 3 corresponding to the position of the viewpoint (VRML only)
vrml_dir	a vector of size 4 specifying the viewing direction (first 3 components) and the rotation of the camera around the direction vector (last component in radians, VRML only)
lg3d_ambientlight	ambient light gray level (value between 0 and 1, LG3D only)

Details

mesh3d visualizes 3D-curves and surfaces specified as parametric functions as well 3D-meshes defined by vertex-coordinates and edges between them. The output can be generated in the VRML- or Livegraphics3D-format, or in the obj-format, if the x,y,z- and edges-parameters are specified. There are three ways to provide 3D data as input: An obj-file in the current directory can be used as by setting the obj_infile-parameter, a 3D mesh can be specified using the x,y,z and edges parameter (in both cases, no other parameters are required), or a parametric function can be defined using the xfun, yfun and zfun parameters (in this case, param1, param2, range1 and range2 parameters have to be set additionally).

To add additional shapes and objects to the final 3D scene, this plotting function can also be called within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function. In this case, mesh3d will inherit all global parameters set in the vrml.open- or lg3d.open-function (e.g. filename, type, htmlout, etc.) and the user does not need to specify these options anymore.

Value

The function is used for its side-effect (creating a VRML-, Livegraphics3D-, or obj-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

cloud3d, bar3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# Visualization of a parametric function
# defining the surface of 3D spiral

mesh3d(xfun = "s * cos(s) * (4 + cos(t + s))",
       yfun = "s * sin(s) * (4 + cos(t + s))",
       zfun = "s * sin(t + s)", param1 = "s",
       param2 = "t",range1 = c(0, 4 * pi),
       range2 = c(0, 2 * pi), type = "vrml",
       filename = "spiral.wrl",
```

```
       htmlout = "spiral.html")

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "spiral.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "spiral.html"), sep = ""))
}

setwd(curdir)
```

`points3d` *Draw points in a 3D-scene*

Description

points3d plots data points in a 3D-scene in the VRML- or Livegraphics3D-format. Must be called after vrml.open() or lg3d.open() and before vrml.close() or lg3d.close().

Usage

```
points3d(x, y = NULL, z = NULL, col = "black",
         pointstyle = "s", transparency = 0,
         hyperlinks = NULL, scale = 1)
```

Arguments

`x`	a 3-column numeric matrix of coordinates or a numeric vector of x-coordinates
`y`	a numeric vector of y-coordinates (only needed if x is a vector)
`z`	a numeric vector of z-coordinates (only needed if x is a vector)
`col`	the color of the text
`pointstyle`	"s" for sphere, "b" for box, "c" for cone
`transparency`	a number between 0 and 1 specifying the transparency level of plotted objects
`hyperlinks`	a vector of strings specifying hyperlinks that will be triggered, when the user clicks on the corresponding datapoint
`scale`	a numerical scaling factor to increase/decrease the point size

Details

points3d plots data points in different point-styles (sphere, box, cone) in a 3D-scene in the VRML- or Livegraphics3D-format. This function can be used to add single data point representations to an already existing 3D scene, or to flexibly combine different plotting styles like points, lines (see lines3d) and text-strings (see text3d) in a 3D environment. To create a standard scatter plot, bar plot

or 3D mesh visualization, the higher-level plotting functions cloud3d, bar3d and mesh3d provide more convenient alternatives. points3d can only be applied within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function and closed using the vrml.\-close() or lg3d.close() function.

Value

The function is used for its side-effect (writing text in a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

text3d, lines3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).

vrml.open(file = "c60.wrl", navigation = "EXAMINE",
          html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black")

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
```

```
    lines3d(c60coords[c(j,k),], col = "gray", lwd = 1)
  }
}

vrml.close()

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

`rot.comb` *Combine rotation vectors*

Description

rot.comb computes the combined rotation vector for two input rotation vectors.

Usage

```
rot.comb(rotvec1 = c(1, 0, 0, pi/2), rotvec2 = c(0, 0, 1, pi))
```

Arguments

`rotvec1`	a numeric vector of length 4, the first 3 components specify a rotation axis and the last component is the rotation angle in radians
`rotvec2`	a numeric vector of length 4, the first 3 components specify a rotation axis and the last component is the rotation angle in radians

Details

The rot.comb helper function computes the combined rotation vector for two input rotation vectors representing the sequential rotation of a 3D-object around two coordinate axis. In each rotation vector, the first 3 components represent the rotation axis (for the input vectors, this can be c(1,0,0) for the x-axis, c(0,1,0) for y-axis or c(0,0,1) for the z-axis), and the last component represent the rotation angle in radians. The computation is achieved by multypling the quaternion-representations of the input vectors. Please note that this multiplication is not commutative, i.e. the order of the input rotation vectors is important.

Value

Returns a numeric vector of length 4, where the first 3 components represent a rotation axis and the last component is rotation angle in radians.

Author(s)

Enrico Glaab

Examples

```
# compute the combined rotation vector for
# 1) a 90-degree rotation around the y-axis, and
# 2) a 180-degree rotation around the z-axis

rot.comb(c(0,1,0,pi/2), c(0,0,1,pi))
```

---

text3d *Draw text in a 3D-scene*

Description

text3d plots text in a 3D-scene in the VRML- or Livegraphics3D-format. Must be called after vrml.open() or lg3d.open() and before vrml.close() or lg3d.close().

Usage

```
text3d(x, y = NULL, z = NULL, text, col = "black",
       fontweight = "normal", fontfamily = "sans",
       hyperlink = NULL, rot = c(0, 1, 0, 0), scale = 1)
```

Arguments

x	a 3-column numeric matrix of coordinates or a numeric vector of x-coordinates
y	a numeric vector of y-coordinates (only needed if x is a vector)
z	a numeric vector of z-coordinates (only needed if x is a vector)
text	a vector of strings which are to be plotted (size must correspond to the number of coordinates specified in the x-, y-, and z-parameters
col	the color of the text
fontweight	"normal" or "bold"
fontfamily	"sans", "serif" or "typewriter"
hyperlink	a string containing a hyperlink to a website
rot	a numeric vector of length 4, the first 3 components specify a rotation axis and the last component is the rotation angle in radians
scale	a numerical scaling factor to increase/decrease the text size

Details

text3d draws one or multiple text strings in a 3D-scene in the VRML- or Livegraphics3D-format. This function can be used to add additional labels to plots or to attach textual information to a 3D-shape or object. text3d can only be applied within a VRML- or Livegraphics3D-environment created by calling the vrml.open() or lg3d.open() function and closed using the vrml.close() or lg3d.close() function.

Value

The function is used for its side-effect (writing text in a VRML- or Livegraphics3D-file) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

points3d, lines3d

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).

vrml.open(file = "c60.wrl", navigation = "EXAMINE",
          html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black")

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords) - 1))
{
```

```
  for(k in (j + 1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 1)
  }
}

vrml.close()

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

---

theo_edges *Theobromine molecular structure example dataset*

Description

Example of a 3D input structure for the mesh3d function (edges given as 3D-vertex indices).

Usage

```
data(theo_edges)
```

Details

This dataset is an example input for the mesh3d function and contains information on the polygonal faces of a molecular structure (theobromine, a molecule similar to caffeine).

Source

The molecular structure was created with the open-source molecular modeling sofware BALL, see www.ballview.org.

theo_vertices *Theobromine molecular structure example dataset*

Description

Example of a 3D input structure for the mesh3d function (3D-vertex coordinates).

Usage

```
data(theo_vertices)
```

Details

This dataset contains the 3D-vertices of a molecular structure used as example input for the mesh3d function. The molecular structure represents theobromine, a molecule similar to caffeine.

Source

The molecular structure was created with the open-source molecular modeling sofware BALL, see www.ballview.org.

uk_topo *Rough topographic height map for the United Kingdom*

Description

A very rough and highly simplified topographic example data set for the United Kingdom

Usage

```
data(uk_topo)
```

Details

This dataset contains very rough height map data for the islands of the United Kingdom as example input for the bar3d function.

Source

The data was obtained from a gray scale topographic image of the United Kingdom obtained from http://maps.simutrans.com/europe.html after removing all height values below a threshold chosen by visual inspection of the outcome.

vrml.close *VRML output device system*

Description

vrml.close terminates a VRML-environment and writes the corresponding 3D-scene to a VRML file.

Usage

```
vrml.close()
```

Details

This function closes a VRML-environment created with vrml.open() and writes the VRML-plot to the file and directory specified in the vrml.open- parameters. For more details, see the description of the vrml.open function.

Value

The function is used for its side-effect (output of a VRML-file in the specified directory) and has no return value.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

vrml.open

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).
```

```
vrml.open(file = "c60.wrl", navigation = "EXAMINE",
          html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black")

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 1)
  }
}

vrml.close()

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

---

vrml.open *VRML output device system*

Description

vrml.open creates a new VRML-environment in which primitive and higher-level plotting functions can be combined together to create a 3D-VRML scene.

Usage

```
vrml.open(filename = "out.wrl", col = "white", navigation = NULL,
          scale = 1, fov = 0.785, pos = rep(scale + 8, 3),
          dir = c(0.19, 0.45, 0.87, 2.45),
          html.embed = "out.html", hwidth = 1200, hheight = 800)
```

Arguments

filename	filename of the generated VRML output file
col	background color
navigation	type of mouse navigation in the VRML file, can be "EXAMINE", "WALK", "SLIDE", "FLY" or "PAN"
scale	a scaling factor for the size of the entire 3D scene
fov	a scalar defining the field of view angle in the VRML file in radians
pos	a vector of size 3 corresponding to the position of the viewpoint
dir	a vector of size 4 specifying the viewing direction (first 3 components) and the rotation of the camera around the direction vector (last component in radians)
html.embed	a filename for generating an HTML-file to embed the VRML-output
hwidth	width of the embedded visualization in the HTML-output
hheight	height of the embedded visualization in the HTML-output

Details

This function creates a new VRML-environment to provide access to primitive plotting functions like points3d, lines3d and text3d and to combine multiple plotting functions together (both primitive and higher-level functions). After creating the VRML-environment, all primitive and higher-level functions called afterwards will automatically inherit the higher-level settings (background color, navigation-type, etc.) specified in the vrml.open-parameters. In order to close the VRML-environment and create the plot, the vrml.close() function has to be called.

Value

The function is used for its side-effect (output of a VRML-file in the specified directory) and has no return value. The htmlout-parameter can be used to embed the resulting VRML-object in a template HTML-file.

Author(s)

Enrico Glaab

References

Enrico Glaab, Jonathan M. Garibaldi, Natalio Krasnogor (2010). vrmlgen: An R Package for 3D Data Visualization on the Web. *Journal of Statistical Software, 36(8)*, p. 1-18. URL: http://www.jstatsoft.org/v36/i08/

See Also

vrml.close

Examples

```
curdir <- getwd()
outdir <- tempdir()
setwd(outdir)

# This example loads the atom coordinates of a molecule
# (C60, fullerene) and visualizes the molecule in 3D
# using points for the atoms and lines for the atom bonds
# (atom pairs within a given distance threshold).

vrml.open(file = "c60.wrl", navigation = "EXAMINE",
          html.embed = "c60example.html")

# load dataset
data(c60coords)

# plot the atoms as black spheres
points3d(c60coords, col = "black")

# plot the atom bonds as gray lines
# (for all atom pairs with a Euclidean distance < 0.66)
for(j in 1:(nrow(c60coords)-1))
{
  for(k in (j+1):nrow(c60coords))
  {
   if(sqrt(sum((c60coords[j,]-c60coords[k,])^2)) < 0.66)
     lines3d(c60coords[c(j,k),], col = "gray", lwd = 1)
  }
}

vrml.close()

# show the output in a web-browser
# (VRML-plugin must be installed!)
if(file.exists(paste("file://",file.path(outdir,
                "c60example.html"), sep = "")))
{
  browseURL(paste("file://",file.path(outdir,
                  "c60example.html"), sep = ""))
}

setwd(curdir)
```

Index