

# Package ‘FourierDescriptors’

May 13, 2012

**Type** Package

**Title** Generate images using Fourier descriptors.

**Version** 0.1-4

**Date** 2012-05-12

**Author** John Myles White

**Maintainer** John Myles White <jmw@johnmyleswhite.com>

**Description** The FourierDescriptors package provides methods for creating, manipulating and visualizing Fourier descriptors, a representational scheme used to describe closer planar contours. The images most easily described using Fourier descriptors are useful as stimuli for experiments in psychology and neuroscience.

**License** Artistic-2.0

**LazyLoad** yes

**Depends** ggplot2

**Suggests** testthat

**Repository** CRAN

**Date/Publication** 2012-05-13 08:35:24

## R topics documented:

FourierDescriptors-package . . . . .	2
complicate.fourier.descriptor . . . . .	3
create.fourier.descriptor . . . . .	3
cumbend . . . . .	4
cumbend.to.points . . . . .	4
plot.fourier.descriptor . . . . .	5
print.fourier.descriptor . . . . .	6
random.fourier.descriptor . . . . .	6
simplify.fourier.descriptor . . . . .	7

---

FourierDescriptors-package

*Generate closed planar contours using Fourier descriptors.*

---

### Description

Generate closed plane curves using the Fourier descriptor method of Zahn and Roskies 1972.

### Details

Package:	FourierDescriptors
Type:	Package
Version:	0.1
Date:	2010-08-24
License:	Artistic-2.0
LazyLoad:	yes

```
plot(create.fourier.descriptor())
```

### Author(s)

John Myles White

Maintainer: John Myles White <jmw@johnmyleswhite.com>

### References

Zahn and Roskies 1972

### Examples

```
library('FourierDescriptors')  
fd <- random.fourier.descriptor(12, 2)  
print(fd)  
plot(fd)
```

---

```
complicate.fourier.descriptor
```

*Add amplitude weights to the zero-weighted frequencies of an existing Fourier descriptor.*

---

**Description**

Add amplitude weights to the zero-weighted frequencies of an existing Fourier descriptor.

**Usage**

```
complicate.fourier.descriptor(fourier.descriptor, quantity = 1, generating.function = function() {retu
```

**Arguments**

`fourier.descriptor`

A Fourier descriptor object.

`quantity`

How many frequency components should be added? Defaults to 1.

`generating.function`

An anonymous function that returns an amplitude weight. The default function adds weight 1 to each new frequency.

**Examples**

```
library('FourierDescriptors')
```

```
fourier.descriptor <- random.fourier.descriptor(12)
```

```
fd <- complicate.fourier.descriptor(fourier.descriptor)
```

---

```
create.fourier.descriptor
```

*Create a new Fourier descriptor.*

---

**Description**

Create a new Fourier descriptor.

**Usage**

```
create.fourier.descriptor(amplitude = c(0, 1, 0, 0), phase = c(0, 0, 0, 0))
```

**Arguments**

`amplitude`

A numeric vector containing the desired amplitude spectrum. Defaults to `c(0, 1, 0, 0)`

`phase`

A numeric vector containing the desired phase spectrum. Defaults to `c(0, 0, 0, 0)`

**Examples**

```
library('FourierDescriptors')  
  
fourier.descriptor <- create.fourier.descriptor()
```

---

cumbend	<i>Evaluate the angular position of a Fourier descriptor at point t.</i>
---------	--

---

**Description**

Evaluate the angular position of a Fourier descriptor at point t.

**Usage**

```
cumbend(fourier.descriptor, t)
```

**Arguments**

fourier.descriptor	A Fourier descriptor object.
t	At which point should the descriptor be evaluated?

**Examples**

```
library('FourierDescriptors')  
  
fourier.descriptor <- random.fourier.descriptor(12)  
  
theta.start <- cumbend(fourier.descriptor, 0)  
theta.end <- cumbend(fourier.descriptor, 2 * pi)
```

---

cumbend.to.points	<i>Produce a set of spatial coordinates induced by a Fourier descriptor object.</i>
-------------------	---

---

**Description**

Produce a set of spatial coordinates induced by a Fourier descriptor object. Returns a data.frame containing the X, Y coordinates of the curve defined by the Fourier descriptor evaluated at steps evenly spaced points along the complete contour.

**Usage**

```
cumbend.to.points(fourier.descriptor, steps = 720)
```

### Arguments

fourier.descriptor      A Fourier descriptor object.

steps                    At how many points along the curve should the descriptor be evaluated? This amounts to a resolution parameter. More complex curves will require a larger value. Defaults to 720.

### Examples

```
fourier.descriptor <- random.fourier.descriptor(12)

points <- cumbend.to.points(fourier.descriptor)

low.res.points <- cumbend.to.points(fourier.descriptor, steps = 20)
```

---

plot.fourier.descriptor

*Plot a Fourier descriptor object. Requires ggplot2.*

---

### Description

Plot a Fourier descriptor object.

### Usage

```
## S3 method for class 'fourier.descriptor'
plot(x, steps = 720, ...)
```

### Arguments

x                        A Fourier descriptor object.

steps                    At how many points along the curve should the descriptor be evaluated? This amounts to a resolution parameter. More complex curves will require higher values. Defaults to 720.

...                      Other arguments passed to underlying plot call.

### Examples

```
library('FourierDescriptors')

fourier.descriptor <- random.fourier.descriptor(12)

plot(fourier.descriptor)

plot(fourier.descriptor, steps = 20)
```

```
print.fourier.descriptor
```

*Print a Fourier descriptor object.*

---

### Description

Print a Fourier descriptor object.

### Usage

```
## S3 method for class 'fourier.descriptor'  
print(x, ...)
```

### Arguments

x	A Fourier descriptor object.
...	Other arguments passed to underlying plot call.

### Examples

```
library('FourierDescriptors')  
  
fourier.descriptor <- random.fourier.descriptor(12)  
  
print(fourier.descriptor)
```

---

```
random.fourier.descriptor
```

*Create a new Fourier descriptor object with randomly assigned amplitude along its even frequency components.*

---

### Description

Create a new Fourier descriptor object with randomly assigned amplitude along its even frequency components.

### Usage

```
random.fourier.descriptor(total.frequencies, non.zero.frequencies = 1, generating.function = function
```

**Arguments**

- `total.frequencies`  
What is the range of frequencies that the descriptor should employ?
- `non.zero.frequencies`  
How many frequency components should be given non-zero amplitude? Defaults to 1.
- `generating.function`  
An anonymous function that returns an amplitude weight. The default function adds weight 1 to each new frequency.

**Examples**

```
library('FourierDescriptors')  
  
fd <- random.fourier.descriptor(12)
```

---

```
simplify.fourier.descriptor  
    Assign zero amplitude to some of the non-zero frequencies of an existing Fourier descriptor.
```

---

**Description**

Assign zero amplitude to some of the non-zero frequencies of an existing Fourier descriptor.

**Usage**

```
simplify.fourier.descriptor(fourier.descriptor, quantity = 1)
```

**Arguments**

- `fourier.descriptor`  
A Fourier descriptor object.
- `quantity`  
How many frequency components should be removed? Defaults to 1.

**Details**

Returns a new Fourier descriptor.

**Examples**

```
library('FourierDescriptors')  
  
fourier.descriptor <- random.fourier.descriptor(12)  
  
fd <- simplify.fourier.descriptor(fourier.descriptor)
```

# Index

## \*Topic **package**

FourierDescriptors-package, [2](#)

complicate.fourier.descriptor, [3](#)

create.fourier.descriptor, [3](#)

cumbend, [4](#)

cumbend.to.points, [4](#)

FourierDescriptors

(FourierDescriptors-package), [2](#)

FourierDescriptors-package, [2](#)

plot (plot.fourier.descriptor), [5](#)

plot.fourier.descriptor, [5](#)

print (print.fourier.descriptor), [6](#)

print.fourier.descriptor, [6](#)

random.fourier.descriptor, [6](#)

simplify.fourier.descriptor, [7](#)