Package 'ambient'

August 30, 2018

Type Package

Title A Generator of Multidimensional Noise

Version 0.1.0

Date 2018-08-24

Maintainer Thomas Lin Pedersen <thomasp85@gmail.com>

Description Generation of natural looking noise has many application within simulation, procedural generation, and art, to name a few. The 'ambient' package provides an interface to the 'FastNoise' C++ library and allows for efficient generation of perlin, simplex, worley, cubic, value, and white noise with optional pertubation in either 2, 3, or 4 (in case of simplex and white noise) dimensions.

License MIT + file LICENSE

Encoding UTF-8

SystemRequirements C++11

LazyData true

Imports Rcpp (>= 0.12.18)

LinkingTo Rcpp

RoxygenNote 6.1.0

URL https://github.com/thomasp85/ambient

BugReports https://github.com/thomasp85/ambient/issues

NeedsCompilation yes

Author Thomas Lin Pedersen [cre, aut],
Jordan Peck [aut] (Developer of FastNoise)

Repository CRAN

Date/Publication 2018-08-30 13:40:03 UTC

R topics documented:

ambient-package 2
noise_cubic 3
noise_perlin 4
noise_simplex 5
noise_value 6
noise_white 7
noise_worley 8

Index **10**

`ambient-package` *ambient: A Generator of Multidimensional Noise*

Description

Generation of natural looking noise has many application within simulation, procedural generation, and art, to name a few. The 'ambient' package provides an interface to the 'FastNoise' C++ library and allows for efficient generation of perlin, simplex, worley, cubic, value, and white noise with optional pertubation in either 2, 3, or 4 (in case of simplex and white noise) dimensions.

Author(s)

Maintainer: Thomas Lin Pedersen `<thomasp85@gmail.com>`

Authors:

- Jordan Peck (Developer of FastNoise)

References

https://github.com/Auburns/FastNoise

See Also

Useful links:

- https://github.com/thomasp85/ambient
- Report bugs at https://github.com/thomasp85/ambient/issues

noise_cubic *Cubic noise generator*

Description

Cubic noise is a pretty simple alternative to perlin and simplex noise. In essence it takes a low resolution white noise and scales it up using cubic interpolation. This approach means that while cubic noise is smooth, it is much more random than perlin and simplex noise.

Usage

```
noise_cubic(dim, frequency = 0.01, fractal = "fbm", octaves = 3,
  lacunarity = 2, gain = 0.5, pertubation = "none",
  pertubation_amplitude = 1)
```

Arguments

dim	The dimensions (height, width, (and depth)) of the noise to be generated. The length determines the dimensionality of the noise.
frequency	Determines the granularity of the features in the noise.
fractal	The fractal type to use. Either 'none', 'fbm' (default), 'billow', or 'rigid-multi'. It is suggested that you experiment with the different types to get a feel for how they behaves.
octaves	The number of noise layers used to create the fractal noise. Ignored if fractal = 'none'. Defaults to 3.
lacunarity	The frequency multiplier between successive noise layers when building fractal noise. Ignored if fractal = 'none'. Defaults to 2.
gain	The relative strength between successive noise layers when building fractal noise. Ignored if fractal = 'none'. Defaults to 0.5.
pertubation	The pertubation to use. Either 'none' (default), 'normal', or 'fractal'. Defines the displacement (warping) of the noise, with 'normal' giving a smooth warping and 'fractal' giving a more eratic warping.
pertubation_amplitude	The maximal pertubation distance from the origin. Ignored if pertubation = 'none'. Defaults to 1.

Value

If length(dim) == 2 a matrix, if length(dim) == 3 a 3-dimensional array.

Examples

```
# Basic use
noise <- noise_cubic(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

noise_perlin *Perlin noise generator*

Description

This function generates either 2 or 3 dimensional perlin noise, with optional pertubation and fractality. Perlin noise is one of the most well known gradient noise algorithms and have been used extensively as the basis for generating landscapes and textures, as well as within generative art. The algorithm was developed by Ken Perlin in 1983.

Usage

```
noise_perlin(dim, frequency = 0.01, interpolator = "quintic",
  fractal = "fbm", octaves = 3, lacunarity = 2, gain = 0.5,
  pertubation = "none", pertubation_amplitude = 1)
```

Arguments

`dim`	The dimensions (height, width, (and depth)) of the noise to be generated. The length determines the dimensionality of the noise.
`frequency`	Determines the granularity of the features in the noise.
`interpolator`	How should values between sampled points be calculated? Either `'linear'`, `'hermite'`, or `'quintic'` (default), ranging from lowest to highest quality.
`fractal`	The fractal type to use. Either `'none'`, `'fbm'` (default), `'billow'`, or `'rigid-multi'`. It is suggested that you experiment with the different types to get a feel for how they behaves.
`octaves`	The number of noise layers used to create the fractal noise. Ignored if `fractal = 'none'`. Defaults to `3`.
`lacunarity`	The frequency multiplier between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `2`.
`gain`	The relative strength between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `0.5`.
`pertubation`	The pertubation to use. Either `'none'` (default), `'normal'`, or `'fractal'`. Defines the displacement (warping) of the noise, with `'normal'` giving a smooth warping and `'fractal'` giving a more eratic warping.
`pertubation_amplitude`	The maximal pertubation distance from the origin. Ignored if `pertubation = 'none'`. Defaults to `1`.

Value

If `length(dim) == 2` a matrix, if `length(dim) == 3` a 3-dimensional array.

References

Perlin, Ken (1985). *An Image Synthesizer*. SIGGRAPH Comput. Graph. 19 (0097-8930): 287–296. doi:10.1145/325165.325247.

Examples

```
# Basic use
noise <- noise_perlin(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

`noise_simplex` *Simplex noise generator*

Description

Simplex noise has been developed by Ken Perlin, the inventor of perlin noise, in order to address some of the shortcomings he saw in perlin noise. Compared to perlin noise, simplex noise has lower computational complexity, making it feasable for dimensions above 3 and has no directional artifacts.

Usage

```
noise_simplex(dim, frequency = 0.01, interpolator = "quintic",
  fractal = "fbm", octaves = 3, lacunarity = 2, gain = 0.5,
  pertubation = "none", pertubation_amplitude = 1)
```

Arguments

`dim`	The dimensions (height, width, (and depth, (and time))) of the noise to be generated. The length determines the dimensionality of the noise.
`frequency`	Determines the granularity of the features in the noise.
`interpolator`	How should values between sampled points be calculated? Either `'linear'`, `'hermite'`, or `'quintic'` (default), ranging from lowest to highest quality.
`fractal`	The fractal type to use. Either `'none'`, `'fbm'` (default), `'billow'`, or `'rigid-multi'`. It is suggested that you experiment with the different types to get a feel for how they behaves.
`octaves`	The number of noise layers used to create the fractal noise. Ignored if `fractal = 'none'`. Defaults to `3`.
`lacunarity`	The frequency multiplier between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `2`.
`gain`	The relative strength between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `0.5`.

`pertubation` The pertubation to use. Either `'none'` (default), `'normal'`, or `'fractal'`. Defines the displacement (warping) of the noise, with `'normal'` giving a smooth warping and `'fractal'` giving a more eratic warping.

`pertubation_amplitude` The maximal pertubation distance from the origin. Ignored if `pertubation = 'none'`. Defaults to `1`.

Value

If `length(dim) == 2` a matrix, if `length(dim) %in% c(3, 4)` a 3- or 4-dimensional array.

References

Ken Perlin, (2001) *Noise hardware*. In Real-Time Shading SIGGRAPH Course Notes, Olano M., (Ed.)

Examples

```
# Basic use
noise <- noise_simplex(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

---

noise_value *Value noise generator*

Description

Value noise is a simpler version of cubic noise that uses linear interpolation between neighboring grid points. This creates a more distinct smooth checkerboard pattern than cubic noise, where interpolation takes all the surrounding grid points into accout.

Usage

```
noise_value(dim, frequency = 0.01, interpolator = "quintic",
  fractal = "fbm", octaves = 3, lacunarity = 2, gain = 0.5,
  pertubation = "none", pertubation_amplitude = 1)
```

Arguments

`dim` The dimensions (height, width, (and depth)) of the noise to be generated. The length determines the dimensionality of the noise.

`frequency` Determines the granularity of the features in the noise.

`interpolator` How should values between sampled points be calculated? Either `'linear'`, `'hermite'`, or `'quintic'` (default), ranging from lowest to highest quality.

`fractal` The fractal type to use. Either `'none'`, `'fbm'` (default), `'billow'`, or `'rigid-multi'`. It is suggested that you experiment with the different types to get a feel for how they behaves.

`octaves` The number of noise layers used to create the fractal noise. Ignored if `fractal = 'none'`. Defaults to `3`.

`lacunarity` The frequency multiplier between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `2`.

`gain` The relative strength between successive noise layers when building fractal noise. Ignored if `fractal = 'none'`. Defaults to `0.5`.

`pertubation` The pertubation to use. Either `'none'` (default), `'normal'`, or `'fractal'`. Defines the displacement (warping) of the noise, with `'normal'` giving a smooth warping and `'fractal'` giving a more eratic warping.

`pertubation_amplitude` The maximal pertubation distance from the origin. Ignored if `pertubation = 'none'`. Defaults to `1`.

Value

If `length(dim) == 2` a matrix, if `length(dim) == 3` a 3-dimensional array.

Examples

```
# Basic use
noise <- noise_value(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

---

noise_white *White noise generator*

Description

White noise is a random noise with equal intensities at different frequencies. It is most well-known as what appeared on old televisions when no signal was found.

Usage

```
noise_white(dim, frequency = 0.01, pertubation = "none",
  pertubation_amplitude = 1)
```

Arguments

dim	The dimensions (height, width, (and depth, (and time))) of the noise to be generated. The length determines the dimensionality of the noise.
frequency	Determines the granularity of the features in the noise.
pertubation	The pertubation to use. Either 'none' (default), 'normal', or 'fractal'. Defines the displacement (warping) of the noise, with 'normal' giving a smooth warping and 'fractal' giving a more eratic warping.
pertubation_amplitude	The maximal pertubation distance from the origin. Ignored if pertubation = 'none'. Defaults to 1.

Value

If length(dim) == 2 a matrix, if length(dim) %in% c(3, 4) a 3- or 4-dimensional array.

Examples

```
# Basic use
noise <- noise_white(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

---

noise_worley *Worley (cell) noise generator*

Description

Worley noise, sometimes called cell (or cellular) noise, is quite distinct due to it's kinship to voronoi tesselation. It is created by sampling random points in space and then for any point in space measure the distance to the closest point. The noise can be modified further by changing either the distance measure or by combining multiple distances. The noise algorithm was developed by Steven Worley in 1996 and has been used to simulated water and stone textures among other things.

Usage

```
noise_worley(dim, frequency = 0.01, distance = "euclidean",
  value = "cell", distance_ind = c(1, 2), jitter = 0.45,
  pertubation = "none", pertubation_amplitude = 1)
```

Arguments

dim	The dimensions (height, width, (and depth)) of the noise to be generated. The length determines the dimensionality of the noise.
frequency	Determines the granularity of the features in the noise.

`distance` The distance measure to use, either `'euclidean'` (default), `'manhattan'`, or `'natural'` (a mix of the two)

`value` The noise value to return. Either

- `'value'` (default) A random value associated with the closest point
- `'distance'` The distance to the closest point
- `'distance2'` The distance to the nth closest point (n given by `distance_ind[1]`)
- `'distance2add'` Addition of the distance to the nth and mth closest point given in `distance_ind`
- `'distance2sub'` Substraction of the distance to the nth and mth closest point given in `distance_ind`
- `'distance2mul'` Multiplication of the distance to the nth and mth closest point given in `distance_ind`
- `'distance2div'` Division of the distance to the nth and mth closest point given in `distance_ind`

`distance_ind` Reference to the nth and mth closest points that should be used when calculating `value`.

`jitter` The maximum distance a point can move from its start position during sampling of cell points.

`pertubation` The pertubation to use. Either `'none'` (default), `'normal'`, or `'fractal'`. Defines the displacement (warping) of the noise, with `'normal'` giving a smooth warping and `'fractal'` giving a more eratic warping.

`pertubation_amplitude` The maximal pertubation distance from the origin. Ignored if `pertubation = 'none'`. Defaults to `1`.

Value

If `length(dim) == 2` a matrix, if `length(dim) == 3` a 3-dimensional array.

References

Worley, Steven (1996). *A cellular texture basis function*. Proceedings of the 23rd annual conference on computer graphics and interactive techniques. pp. 291–294. ISBN 0-89791-746-4

Examples

```
# Basic use
noise <- noise_worley(c(100, 100))

image(noise, col = grey.colors(256, 0, 1))
```

Index

ambient *(*ambient-package*)*, 2
ambient-package, 2

noise_cubic, 3
noise_perlin, 4
noise_simplex, 5
noise_value, 6
noise_white, 7
noise_worley, 8