

Package: rectpacker (via r-universe)

December 6, 2024

Type Package

Title Rectangle Packing

Version 1.0.0

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

URL <https://github.com/coolbutuseless/rectpacker>

BugReports <https://github.com/coolbutuseless/rectpacker/issues>

Description Rectangle packing is a packing problem where rectangles are placed into a larger rectangular region (without overlapping) in order to maximise the use space. Rectangles are packed using the skyline heuristic as discussed in Lijun et al (2011) 'A Skyline-Based Heuristic for the 2D Rectangular Strip Packing Problem' <[doi:10.1007/978-3-642-21827-9_29](https://doi.org/10.1007/978-3-642-21827-9_29)>. A function is also included for determining a good small-sized box for containing a given set of rectangles.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Copyright The included 'stb_rect_pack.h' header (v1.01) is Copyright (c) 2017 Sean Barrett and licensed under the MIT license. See COPYRIGHTS file for more details.

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Mike Cheng [aut, cre, cph], Sean Barrett [aut, cph] (Author of included stb_rect_pack.h library)

Repository CRAN

Date/Publication 2024-12-05 19:00:02 UTC

Contents

| | |
|--------------------------|----------|
| calc_small_box | 2 |
| pack_rects | 3 |
| Index | 4 |

| | |
|----------------|---|
| calc_small_box | <i>Find the dimensions of a small box to store all the given rectangles</i> |
|----------------|---|

Description

This is a brute force search with a simple heuristic. Is not guaranteed to find the box with the minimum area, but simply a box that snugly fits the rectangles without too much wasted space.

Usage

```
calc_small_box(
  rect_widths,
  rect_heights,
  aspect_ratios = c(1.61803, 1/1.61803),
  verbosity = 0L
)
```

Arguments

`rect_widths, rect_heights` widths and heights of the rectangles to pack.

`aspect_ratios` Vector of box aspect ratios to be tested. Aspect ratio is defined here as width / height. Default: `c(1.61803, 1/1.61803)` i.e. golden ratio and its inverse.

`verbosity` Level of debugging output. Default: 0 (no output)

Value

List with 2 elements: width and height of a small box which fits all the rectangles.

Examples

```
# Find a minimal box to fit 10 random rectangles.
# Search for boxes with aspect ratios in seq(0.5, 2, length.out = 20)
set.seed(2)
N <- 10
rect_widths <- sample(N)
rect_heights <- sample(N)
box <- calc_small_box(rect_widths, rect_heights,
  aspect_ratios = seq(0.5, 2, length.out = 20))
box
rects <- pack_rects(box$width, box$height, rect_widths, rect_heights)
all(rects$packed)
```

| | |
|------------|---|
| pack_rects | <i>Pack rectangles into a box using the skyline algorithm</i> |
|------------|---|

Description

This implementation accepts only integer valued sizes and coordinates.

Usage

```
pack_rects(box_width, box_height, rect_widths, rect_heights)
```

Arguments

box_width, box_height
dimensions of the box into which the rectangles will be packed. Integer values.

rect_widths, rect_heights
widths and heights of the rectangles to pack.

Value

data.frame of packing information

idx Integer index of rectangle in the input

w, h Integer dimensions of each rectangle

packed Logical: Was this rectangle packed into the box?

x, y Integer coordinates of packing position of bottom-left of rectangle

Examples

```
# Pack 10 rectangles into a 25x25 box
# Note: All rectangles in the results have 'packed=TRUE' which
# means they all fit into the box
set.seed(1)
N <- 10
rect_widths <- sample(N)
rect_heights <- sample(N)
pack_rects(box_width = 25, box_height = 25, rect_widths, rect_heights)
```

Index

`calc_small_box`, [2](#)

`pack_rects`, [3](#)