Package 'image.LineSegmentDetector'

May 8, 2026

Type Package

Title Detect Line Segments in Images

Description An implementation of the Line Segment Detector on digital images described in the paper: ``LSD: A Fast Line Segment Detector with a False Detection Control" by Rafael Grompone von Gioi et al (2012).
The algorithm is explained at <doi:10.5201/ipol.2012.gjmr-lsd>.

Maintainer Jan Wijffels <jwijffels@bnosac.be>

License AGPL-3

Version 0.1.1

URL https://github.com/bnosac/image

Imports Rcpp (>= 0.12.8), sp

LinkingTo Rcpp

Suggests pixmap, magick

RoxygenNote 7.3.2

NeedsCompilation yes

Author Jan Wijffels [aut, cre, cph] (R wrapper),
BNOSAC [cph] (R wrapper),
Rafael Grompone von Gioi [ctb, cph] (src/lsd)

Repository CRAN

Date/Publication 2025-11-27 20:50:02 UTC

Contents

`image.LineSegmentDetector-package`

The image.LineSegmentDetector (LSD) package detects line segments in images

---

Description

LSD is a linear-time Line Segment Detector giving subpixel accurate results. It is designed to work on any digital image without parameter tuning. It controls its own number of false detections: On average, one false alarm is allowed per image. The method is based on Burns, Hanson, and Riseman's method, and uses an a-contrario validation approach according to Desolneux, Moisan, and Morel's theory.

Author(s)

Maintainer: Jan Wijffels <jwijffels@bnosac.be> (R wrapper) [copyright holder]

Other contributors:

- BNOSAC (R wrapper) [copyright holder]
- Rafael Grompone von Gioi <grompone@gmail.com> (src/lsd) [contributor, copyright holder]

References

LSD: A Fast Line Segment Detector with a False Detection Control" by Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 722-732, April, 2010. doi:10.5201/ipol.2012.gjmrlsd

See Also

image_line_segment_detector

`image_line_segment_detector`

Line Segments Detection (LSD) in an image

Description

LSD is a linear-time Line Segment Detector giving subpixel accurate results. It is designed to work on any digital image without parameter tuning. It controls its own number of false detections (NFA): On average, one false alarm is allowed per image. The method is based on Burns, Hanson, and Riseman's method, and uses an a-contrario validation approach according to Desolneux, Moisan, and Morel's theory.

In general, no changes are needed on the arguments. The default arguments provide very good line detections for any digital image without changing the parameters.

Usage

```
image_line_segment_detector(
  x,
  scale = 0.8,
  sigma_scale = 0.6,
  quant = 2,
  ang_th = 22.5,
  log_eps = 0,
  density_th = 0.7,
  n_bins = 1024,
  union = FALSE,
  union_min_length = 5,
  union_max_distance = 5,
  union_ang_th = 7,
  union_use_NFA = FALSE,
  union_log_eps = 0
)
```

Arguments

x	a matrix of image pixel values in the 0-255 range.
scale	Positive numeric value. When different from 1.0, LSD will scale the input image by 'scale' factor by Gaussian filtering, before detecting line segments. Example: if scale=0.8, the input image will be subsampled to 80 percent of its size, before the line segment detector is applied. Suggested value: 0.8
sigma_scale	Positive numeric value. When scale != 1.0, the sigma of the Gaussian filter is: sigma = sigma_scale / scale, if scale < 1.0 and sigma = sigma_scale, if scale >= 1.0. Suggested value: 0.6
quant	Positive numeric value. Bound to the quantization error on the gradient norm. Example: if gray levels are quantized to integer steps, the gradient (computed by finite differences) error due to quantization will be bounded by 2.0, as the worst case is when the error are 1 and -1, that gives an error of 2.0. Suggested value: 2.0
ang_th	Positive numeric value in 0-180 range. Gradient angle tolerance in the region growing algorithm, in degrees. Suggested value: 22.5
log_eps	Detection threshold, accept if -log10(NFA) > log_eps. The larger the value, the more strict the detector is, and will result in less detections. (Note that the 'minus sign' makes that this behavior is opposite to the one of NFA.). Suggested value: 0.0.
density_th	Positive numeric value in 0-1 range. Minimal proportion of 'supporting' points in a rectangle. Suggested value: 0.7.
n_bins	Positive integer value. Number of bins used in the pseudo-ordering of gradient modulus. Suggested value: 1024
union	Logical indicating if you need to post procces image by union close segments. Defaults to FALSE.

`union_min_length`
: Numeric value with minimum length of segment to union

`union_max_distance`
: Numeric value with maximum distance between two line which we would union

`union_ang_th`
: Numeric value with angle threshold in order to union

`union_use_NFA`
: Logical indicating to use NFA to union

`union_log_eps`
: Detection threshold to union

Value

an object of class lsd which is a list with the following elements

- n: The number of found line segments
- lines: A matrix with detected line segments with columns x1, y1, x2, y2, width, p, -log_nfa". Each row corresponds to a line where a line segment is given by the coordinates (x1, y1) to (x2, y2). Column width indicates the width of the line, p is the angle precision in (0, 1) given by angle_tolerance/180 degree, and NFA stands for the Number of False Alarms. The value -log10(NFA) is equivalent but more intuitive than NFA (Number of False Alarms): -1 corresponds to 10 mean false alarms, 0 corresponds to 1 mean false alarm, 1 corresponds to 0.1 mean false alarms, 2 corresponds to 0.01 mean false alarms
- pixels: A matrix where each element correponds to a pixel where each pixel indicates the line segment to which it belongs. Unused pixels have the value '0', while the used ones have the number of the line segment, numbered 1,2,3,..., in the same order as in the lines matrix

References

LSD: A Fast Line Segment Detector with a False Detection Control" by Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 722-732, April, 2010. doi:10.5201/ipol.2012.gjmrlsd

Examples

```
library(pixmap)
imagelocation <- system.file("extdata", "chairs.pgm", package="image.LineSegmentDetector")
image <- read.pnm(file = imagelocation, cellres = 1)
x <- image@grey * 255

linesegments <- image_line_segment_detector(x)
linesegments
plot(image)
plot(linesegments, add = TRUE, col = "red")


imagelocation <- system.file("extdata", "le-piree.pgm", package="image.LineSegmentDetector")
image <- read.pnm(file = imagelocation, cellres = 1)
linesegments <- image_line_segment_detector(image@grey * 255)
plot(image)
plot(linesegments)
```

```
##
## image_line_segment_detector expects a matrix as input
##  if you have a jpg/png/... convert it to pgm first or take the r/g/b channel
library(magick)
x  <- image_read(system.file("extdata", "atomium.jpg", package="image.LineSegmentDetector"))
mat <- image_data(x, channels = "gray")
mat <- as.integer(mat, transpose = TRUE)
mat <- drop(mat)
linesegments <- image_line_segment_detector(mat)
plot(linesegments, lwd = 2)
```

`plot.lsd` *Plot the detected lines from the image_line_segment_detector*

Description

Plot the detected lines from the image_line_segment_detector

Usage

```
## S3 method for class 'lsd'
plot(x, ...)
```

Arguments

`x`	an object of class lsd as returned by `image_line_segment_detector`
`...`	further arguments passed on to plot

Value

invisibly a SpatialLines object with the lines

Examples

```
library(pixmap)
imagelocation <- system.file("extdata", "le-piree.pgm", package="image.LineSegmentDetector")
image <- read.pnm(file = imagelocation, cellres = 1)
linesegments <- image_line_segment_detector(image@grey * 255)
plot(image)
plot(linesegments, add = TRUE, col = "red")
```

Index