Package 'thinr'

July 26, 2026

Title Binary Image Thinning Algorithms

Version 0.3.0

Description Thinning (skeletonization) algorithms for binary raster images. Provides seven algorithms behind a single dispatching function: Zhang-Suen (Zhang and Suen 1984) <doi:10.1145/357994.358023>, Guo-Hall (Guo and Hall 1989) <doi:10.1145/62065.62074>, a 2-D adaptation of Lee (Lee, Kashyap, and Chu 1994) <doi:10.1006/cgip.1994.1042>, K3M (Saeed, Tabedzki, Rybnik, and Adamski 2010) <doi:10.2478/v10006-010-0024-4>, the parallel form commonly attributed to Hilditch (1969, in 'Machine Intelligence 4'), OPTA / SPTA (Naccache and Shinghal 1984), and Holt and colleagues (1987) <doi:10.1145/12527.12531>. Also provides the medial axis transform (Blum 1967) and a distance transform implementation following Felzenszwalb and Huttenlocher (2012) <doi:10.4086/toc.2012.v008a019>. The thin() API selects the algorithm by name, defaulting to Zhang-Suen. Complements the morphology in the 'EBImage' package, which does not provide a thinning operator.

License LGPL-3

Encoding UTF-8

Depends R (>= 4.2)

LinkingTo Rcpp

Imports Rcpp

Suggests bench, covr, knitr, lintr, rmarkdown, styler, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 8.0.0

URL https://github.com/humanpred/thinr, https://humanpred.github.io/thinr/

BugReports https://github.com/humanpred/thinr/issues

NeedsCompilation yes

Author Bill Denney [aut, cre] (ORCID: <https://orcid.org/0000-0002-5759-428X>, affiliation: Human Predictions, LLC)

Maintainer Bill Denney <wdenney@humanpredictions.com>

Repository CRAN

Date/Publication 2026-07-26 21:10:02 UTC

Contents

`distance_transform` *Distance transform of a binary image*

Description

Compute the distance from each foreground pixel to the nearest background pixel, under one of three standard metrics.

Usage

```
distance_transform(image, metric = c("euclidean", "manhattan", "chessboard"))
```

Arguments

`image`	A binary image: a matrix where non-zero values are foreground and zero values are background. Logical, integer, and numeric inputs are accepted. `NA` values are rejected with an error rather than silently coerced.
`metric`	Distance metric. One of: • `"euclidean"` (default) — exact L2 distance, via Felzenszwalb & Huttenlocher (2012) linear-time separable algorithm. • `"manhattan"` — L1 distance via two-pass forward + backward sweep (Rosenfeld & Pfaltz 1968). • `"chessboard"` — L_infinity (Chebyshev) distance via the same two-pass sweep with 8-connected propagation.

Value

A numeric matrix of the same shape as `image`. Background pixels are 0; foreground pixels carry their distance to the nearest background pixel. When the image is entirely foreground (no background pixel exists) every pixel is `Inf`, consistently across all three metrics — the distance to a non-existent nearest background pixel is unbounded, matching `EBImage::distmap()`. Only pixels count as background; the region outside the matrix is not treated as an implicit background border.

References

Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, 8(19), 415-428. doi:10.4086/toc.2012.v008a019

Rosenfeld, A., & Pfaltz, J. L. (1968). Distance functions on digital pictures. *Pattern Recognition*, 1(1), 33-61. doi:10.1016/00313203(68)900137

Examples

```
# A 5x5 image with a single background pixel in the corner.
m <- matrix(1L, nrow = 5, ncol = 5)
m[1, 1] <- 0L
distance_transform(m, metric = "manhattan")
distance_transform(m, metric = "chessboard")
round(distance_transform(m, metric = "euclidean"), 3)
```

`medial_axis` *Medial axis transform*

Description

Return the medial axis of a binary image: the locus of foreground pixels that are local maxima of the (squared) Euclidean distance transform in at least one of the four principal directions (horizontal, vertical, NW-SE diagonal, NE-SW diagonal). Each skeleton pixel carries width information via the distance value at that point.

Usage

```
medial_axis(image, return_distance = FALSE)
```

Arguments

`image`	A binary image: a matrix where non-zero values are foreground and zero values are background. Logical, integer, and numeric inputs are accepted; `NA` values are not.
`return_distance`	Logical. If `FALSE` (default), return only the binary skeleton in the same storage mode as `image`. If `TRUE`, return a list with elements `skeleton` (the binary skeleton) and `distance` (a numeric matrix of Euclidean distances from each foreground pixel to the nearest background pixel, with 0 on background pixels).

Details

This is different from `thin()`: classical thinning algorithms produce a connected, 1-pixel-wide skeleton without width information. The medial axis transform (Blum 1967) produces a skeleton **with** width information, useful for shape analysis where local thickness matters.

Value

Either a matrix (when `return_distance = FALSE`) or a `list(skeleton, distance)` (when `return_distance = TRUE`).

References

Blum, H. (1967). A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form* (pp. 362-380). MIT Press.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, 8(19), 415-428. doi:10.4086/toc.2012.v008a019

Examples

```
# A 7x9 solid rectangle: the medial axis is the middle row.
m <- matrix(0L, nrow = 7, ncol = 9)
m[3:5, 3:7] <- 1L
medial_axis(m)

# Returning width information alongside the skeleton.
result <- medial_axis(m, return_distance = TRUE)
result$skeleton
round(result$distance, 3)
```

`thin` *Thin (skeletonize) a binary image*

Description

Reduce a binary image to its one-pixel-wide skeleton using one of the supported thinning algorithms.

Usage

```
thin(
  image,
  method = c("zhang_suen", "guo_hall", "lee", "k3m", "hilditch", "opta", "holt"),
  max_iter = 1000L
)
```

Arguments

image	A binary image: a matrix or array where non-zero values are foreground and zero values are background. Logical, integer, and numeric inputs are all accepted. NA values are rejected with an error rather than silently coerced (a background pixel would change the skeleton). The image is treated as a 2-D matrix; arrays with more than two dimensions are not yet supported.
method	Algorithm to use. One of "zhang_suen" (default), "guo_hall", "lee" (2-D adaptation of Lee, Kashyap & Chu 1994), "k3m" (Saeed et al. 2010), "hilditch" (Hilditch 1969), "opta" (Naccache & Shinghal 1984), or "holt" (Holt et al. 1987). See vignette("choosing-a-method") for guidance on which to pick, and vignette("correctness-properties") for the edge-handling and connectivity guarantees shared by every method.
max_iter	Maximum number of passes. Default 1000. Real binary images of typical sizes converge well under 50 passes; the limit is a safety bound against pathological inputs.

Value

A matrix of the same shape and storage mode as image, with foreground pixels marking the thinned skeleton and the rest set to background.

Edge handling

Every kernel inspects an 8-neighbourhood, so a naive implementation can never delete a pixel in the outermost row or column and leaves shapes that touch the matrix border two or three pixels thick. thin() therefore surrounds the image with a one-pixel background margin before thinning and crops it back afterwards, so a shape is skeletonised identically whether or not it touches the frame. This applies uniformly to all seven methods.

Connectivity

Thinning only ever deletes foreground pixels, and each method deletes in an order that keeps 8-connected components connected: an object that starts as one connected component thins to one connected component. In particular a two-pixel-wide stroke thins to a connected one-pixel line rather than fragmenting. See vignette("correctness-properties").

Examples

```
# A 3x3 solid square thins to a single foreground pixel.
m <- matrix(c(0, 0, 0, 0, 0,
              0, 1, 1, 1, 0,
              0, 1, 1, 1, 0,
              0, 1, 1, 1, 0,
              0, 0, 0, 0, 0),
            nrow = 5, byrow = TRUE)
thin(m, method = "zhang_suen")
thin(m, method = "guo_hall")
thin(m, method = "hilditch")
```

Index