

# Package ‘image.CornerDetectionHarris’

May 8, 2026

**Type** Package

**Title** Implementation of the Harris Corner Detection for Images

**Description** An implementation of the Harris Corner Detection as described in the paper ``An Analysis and Implementation of the Harris Corner Detector" by Sánchez J. et al (2018) available at <[doi:10.5201/ipol.2018.229](https://doi.org/10.5201/ipol.2018.229)>. The package allows to detect relevant points in images which are characteristic to the digital image.

**Maintainer** Jan Wijffels <[jwijffels@bnosac.be](mailto:jwijffels@bnosac.be)>

**License** BSD\_2\_clause + file LICENSE

**Version** 0.1.2

**URL** <https://github.com/bnosac/image>

**Imports** Rcpp (>= 0.12.8)

**LinkingTo** Rcpp

**Suggests** magick

**RoxygenNote** 7.1.2

**Encoding** UTF-8

**NeedsCompilation** yes

**Author** Jan Wijffels [aut, cre, cph] (R wrapper),  
BNOSAC [cph] (R wrapper),  
Javier Sánchez Pérez [ctb, cph] (Harris Corner Detector C/C++ code),  
Pascal Getreuer [ctb, cph] (src/gaussian.cpp)

**Repository** CRAN

**Date/Publication** 2024-01-08 19:50:02 UTC

## Contents

image_harris . . . . .	2
<b>Index</b>	<b>4</b>

---

 image\_harris

*Find Corners using Harris Corner Detection*


---

## Description

An implementation of the Harris Corner Detection algorithm explained at doi: [10.5201/ipol.2018.229](https://doi.org/10.5201/ipol.2018.229).

## Usage

```
image_harris(
  x,
  k = 0.06,
  sigma_d = 1,
  sigma_i = 2.5,
  threshold = 130,
  gaussian = c("fast Gaussian", "precise Gaussian", "no Gaussian"),
  gradient = c("central differences", "Sobel operator"),
  strategy = c("all corners", "sort all corners", "N corners", "distributed N corners"),
  Nselect = 1L,
  measure = c("Harris", "Shi-Tomasi", "Harmonic Mean"),
  Nscales = 1L,
  precision = c("quadratic approximation", "quartic interpolation", "no subpixel"),
  cells = 10L,
  verbose = FALSE
)
```

## Arguments

x	an object of class magick-image or a greyscale matrix of image pixel values in the 0-255 range where values start at the top left corner.
k	Harris' K parameter. Defaults to 0.06.
sigma_d	Gaussian standard deviation for derivation. Defaults to 1.
sigma_i	Gaussian standard deviation for integration. Defaults to 2.5.
threshold	threshold for eliminating low values. Defaults to 130.
gaussian	smoothing, either one of 'precise Gaussian', 'fast Gaussian' or 'no Gaussian'. Defaults to 'fast Gaussian'.
gradient	calculation of gradient, either one of 'central differences' or 'Sobel operator'. Defaults to 'central differences'.
strategy	strategy for selecting the output corners, either one of 'all corners', 'sort all corners', 'N corners', 'distributed N corners'. Defaults to 'all corners'.
Nselect	number of output corners. Defaults to 1.
measure	either one of 'Harris', 'Shi-Tomasi' or 'Harmonic Mean'. Defaults to 'Harris'.
Nscales	number of scales for filtering out corners. Defaults to 1.

precision	subpixel accuracy, either one of 'no subpixel', 'quadratic approximation', 'quartic interpolation'. Defaults to 'quadratic approximation'
cells	regions for output corners (1x1, 2x2, ..., NxN). Defaults to 10.
verbose	logical, indicating to show the trace of different substeps

### Value

as list of the relevant points with the x/y locations as well as the strenght. Note y values start at the top left corner of the image.

### Examples

```
library(magick)
path <- system.file(package = "image.CornerDetectionHarris",
                    "extdata", "building.png")
x <- image_read(path)
pts <- image_harris(x)
pts

plt <- image_draw(x)
points(pts$x, pts$y, col = "red", pch = 20)
dev.off()
plt <- image_draw(x)
points(pts$x, pts$y,
       col = "red", pch = 20, cex = 5 * pts$strength / max(pts$strength))
dev.off()

## Or pass on a greyscale matrix starting at top left
mat <- image_data(x, channels = "gray")
mat <- as.integer(mat, transpose = FALSE)
mat <- drop(mat)
pts <- image_harris(mat)
plt <- image_draw(x)
points(pts$x, pts$y, col = "red", pch = 20)
dev.off()
```

# Index

`image_harris`, [2](#)