Package 'stopmotion'

March 24, 2026

Type Package

Title Build Stop Motion Animations from Image Sequences

Version 0.1.0

License MIT + file LICENSE

Description A pipeline-friendly toolkit for assembling stop motion animations from sequences of still images. Provides functions to read image directories, restructure frame sequences (duplicate, splice, arrange), apply per-frame pixel transformations (rotate, wiggle, flip, flop, blur, scale, crop, trim, border, background), and export the result as a GIF. All transformation functions accept a 'frames' argument to target any subset of frames, bridging the gap between 'magick' functions that operate on an entire image stack and fine-grained stop motion editing. Image processing is performed via 'ImageMagick Studio LLC' (2024) <https://imagemagick.org>.

Imports magick, checkmate

Suggests testthat (>= 3.0.0), withr, knitr, quarto

VignetteBuilder quarto

Config/testthat/edition 3

Config/testthat/parallel false

Config/testthat/start-first *utils*, *read*, *duplicate*

Language en-GB

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Alban Sagouis [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3827-1063>)

Maintainer Alban Sagouis <sagouis@pm.me>

Repository CRAN

Date/Publication 2026-03-24 10:30:03 UTC

Contents

`arrange` *Reorder frames in a stop-motion film*

Description

Returns a new `magick-image` object with frames placed in the order given by `order`. Use this after `read` when the lexicographic filename sort does not match the intended frame sequence.

Usage

```
arrange(images, order)
```

Arguments

`images`	an object of class `magick-image` to reorder.
`order`	integer vector of frame indices giving the desired order. Must be a permutation of `1:length(images)` (every frame index appearing exactly once).

Value

a `magick-image` object with frames in the requested order.

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
# Swap the first and second frame.
images <- arrange(images, order = c(2L, 1L, seq(3L, 10L)))
```

---

background *Set the background colour of images*

Description

Set the background colour of images

Usage

```
background(images, color = "white", frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`color`	a character string specifying a colour, e.g. `"white"` or `"#FF0000"`.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
background(images = images, color = "white")
background(images = images, color = "white", frames = 1)
```

`blur` *Blur images*

Description

Applies a Gaussian blur to selected frames. Wraps `magick::image_blur`.

Usage

```
blur(images, radius = 1, sigma = 0.5, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`radius`	a non-negative number specifying the blur radius in pixels.
`sigma`	a non-negative number specifying the standard deviation of the Gaussian, controlling blur strength. Use `0` for no blur.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
blur(images = images, radius = 2, sigma = 1)
blur(images = images, radius = 2, sigma = 1, frames = 1:3)
```

`border` *Add a border to images*

Description

Adds a coloured border around selected frames. Wraps `magick::image_border`.

Usage

```
border(images, color = "lightgray", geometry = "10x10", frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`color`	a character string specifying the border colour, e.g. `"black"` or `"#FF0000"`.
`geometry`	a geometry string specifying border width and height, e.g. `"10x10"` for a 10-pixel border on all sides.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
border(images = images, color = "black", geometry = "5x5")
border(images = images, color = "white", geometry = "10x10", frames = 1:3)
```

centre *Align frames to a common set of reference points*

Description

Transforms selected frames so that two user-supplied reference points (e.g. left and right eye positions) map onto the same pixel locations across all frames. The transformation is a full affine warp — rotation, scaling, and translation are applied simultaneously — computed from the two point correspondences via `magick::image_distort`.

Usage

```
centre(images, points, reference = 1L, frames = NULL)

center(images, points, reference = 1L, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`points`	a `data.frame` with columns `frame` (integer frame index), `x` (numeric, pixels from the left edge), and `y` (numeric, pixels from the *bottom* edge, as returned by `locator()` after `plot(as.raster(images[[i]]))`). Exactly two rows per frame are required. Within each frame, the first row is reference point 1 and the second is reference point 2; the pairing must be consistent across frames (e.g. always left-eye first, right-eye second).
`reference`	integer. The frame whose reference points define the target alignment. All other selected frames are warped to match it. The reference frame itself is left unchanged. Defaults to `1L`.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object of the same length as `images`.

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)

# Manually record eye positions for each frame (e.g. using locator())
```

```
points <- data.frame(
  frame = c(1L, 1L, 2L, 2L, 3L, 3L),
  x     = c(210, 390, 215, 388, 208, 392),
  y     = c(180, 182, 176, 179, 183, 181)
)

centre(images = images, points = points, reference = 1L, frames = 1:3)
```

`crop` *Crop images*

Description

Crops selected frames to a given geometry. Wraps `magick::image_crop`.

Usage

```
crop(images, geometry, gravity = NULL, repage = TRUE, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`geometry`	a geometry string specifying the cropped region, e.g. `"100x100+10+10"` (widthx-height+x_offset+y_offset).
`gravity`	anchor point for the crop: one of `"NorthWest"`, `"North"`, `"NorthEast"`, `"West"`, `"Center"`, `"East"`, `"SouthWest"`, `"South"`, `"SouthEast"`. Defaults to `NULL` (top-left).
`repage`	logical. If `TRUE` (default), resets the virtual canvas after cropping.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
crop(images = images, geometry = "200x200+50+50")
crop(images = images, geometry = "200x200", gravity = "Center", frames = 1:3)
```

`duplicate` *Duplicate frames*

Description

Duplicate frames

Usage

```
duplicate(images, style = c("linear", "looped", "shuffle"), frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`style`	one of `"linear"`, `"looped"`, or `"shuffle"`, controlling how duplicates are inserted: `"linear"` inserts one copy immediately before each selected frame, in order. The original frame follows its duplicate. `"looped"` appends one copy of each selected frame (in order) after `max(frames)`, creating a loop-back effect. Requires `frames` to be a consecutive sequence. `"shuffle"` randomly reorders both the originals and their copies within the selected range, replacing those positions. Requires `frames` to be a consecutive sequence.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object with duplicate frames inserted.

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
duplicate(images = images, style = "shuffle", frames = 1:2)
```

flip *Flip images vertically*

Description

Flip images vertically

Usage

```
flip(images, frames = NULL)
```

Arguments

images	an object of class magick-image to modify
frames	integer vector of frame indices to duplicate. Defaults to NULL, which duplicates all frames.

Value

a magick-image object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use stopmotion_verbosity(FALSE) to suppress these messages, or set options(stopmotion.verbose = FALSE) in your script or '.Rprofile'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
flip(images = images, frames = 2:3)
```

flop *Flop images horizontally*

Description

Mirrors selected frames along the vertical axis (left-right reflection). For a vertical flip (top-bottom), see flip. Wraps magick::image_flop.

Usage

```
flop(images, frames = NULL)
```

Arguments

images	an object of class magick-image to modify
frames	integer vector of frame indices to duplicate. Defaults to NULL, which duplicates all frames.

Value

a magick-image object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use stopmotion_verbosity(FALSE) to suppress these messages, or set options(stopmotion.verbose = FALSE) in your script or '.Rprofile'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
flop(images = images)
flop(images = images, frames = 2:3)
```

---

montage *Display frames as a montage*

Description

Arranges selected frames into a single composite image. Wraps magick::image_montage.

Usage

```
montage(
  images,
  geometry = NULL,
  tile = NULL,
  gravity = "Center",
  bg = "white",
  shadow = FALSE,
  frames = NULL
)
```

Arguments

images	an object of class magick-image to modify
geometry	a geometry string controlling the size and spacing of each tile, e.g. "64x64+2+2".
tile	a string specifying the grid layout, e.g. "5x2". Defaults to NULL, which lets ImageMagick choose.
gravity	anchor point for each tile's label and content: one of "Center", "North", "South", etc. Defaults to "Center".
bg	background colour string, e.g. "white".
shadow	logical. Whether to add a drop-shadow under each tile.
frames	integer vector of frame indices to duplicate. Defaults to NULL, which duplicates all frames.

Value

a magick-image object containing a single composite frame.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
montage(images)
montage(images, frames = 1:4, tile = "4x1", geometry = "128x128+4+4")
```

---

preview *Preview an animation*

Description

Converts a stack of frames into an animated GIF for display, with each frame's index and label overlaid as text. In an interactive session the animation opens in the system viewer; in a knitr/Quarto document it is embedded as an inline animated GIF. Wraps magick::image_animate.

Usage

```
preview(images, fps = 10, loop = 0, frames = NULL, label = TRUE)
```

Arguments

images	an object of class magick-image to modify
fps	playback speed in frames per second. Must be a positive integer divisor of 100, because GIF delay is stored in hundredths of a second (delay = 100 / fps). Valid values: 1, 2, 4, 5, 10, 20, 25, 50, 100. Defaults to 10.
loop	a non-negative integer giving the number of times the animation loops. 0 (the default) means loop forever.

frames	integer vector of frame indices to duplicate. Defaults to NULL, which duplicates all frames.
label	logical. Whether to overlay the frame index and label on each frame. Defaults to TRUE.

Value

a magick-image object containing the animated sequence.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
preview(images)
preview(images, fps = 5)
preview(images, label = FALSE)
```

---

read *Read images into a stop-motion film*

Description

Reads all image files from dir (optionally filtered by pattern) and returns them as a magick-image object.

Usage

```
read(dir, pattern = "")
```

Arguments

dir	path to directory containing the images relative to working directory.
pattern	an optional regular expression. Only file names which match the regular expression will be returned.

Value

an object of class magick-image

Frame order

Frames are loaded in the order returned by list.files, which sorts filenames lexicographically. This means the filesystem filename order determines the stop-motion frame order. Name your files accordingly (e.g. frame_001.png, frame_002.png, ...) to guarantee the intended sequence. If you need to reorder frames after loading, use arrange.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
```

`rotate` *Rotate images*

Description

Replaces selected frames with a rotated version in place. For large-angle transformations. For a small-angle hand-held rock effect, see `wiggle`.

Usage

```
rotate(images, degrees, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`degrees`	a number in `[-360, 360]` specifying the rotation angle.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
rotate(images = images, degrees = 90, frames = 2L)
```

scale *Scale images*

Description

Scale images

Usage

```
scale(images, geometry, frames = NULL)
```

Arguments

images	an object of class magick-image to modify
geometry	a character string specifying the target geometry, e.g. "50%" or "800x600".
frames	integer vector of frame indices to duplicate. Defaults to NULL, which duplicates all frames.

Value

a magick-image object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use stopmotion_verbosity(FALSE) to suppress these messages, or set options(stopmotion.verbose = FALSE) in your script or '.Rprofile'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
scale(images = images, geometry = "50%")
scale(images = images, geometry = "50%", frames = 2:3)
```

splice *Splice frames into a film*

Description

Splice frames into a film

Usage

```
splice(images, insert, after)
```

Arguments

images	an object of class magick-image to modify
insert	an object of class magick-image containing 1 or more images to be inserted.
after	integer scalar (or vector of scalars) giving the frame number(s) after which insert will be inserted. When a vector is supplied the insertions are applied left-to-right, each offset by the cumulative growth of the film from prior insertions.

Value

a magick-image object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use stopmotion_verbosity(FALSE) to suppress these messages, or set options(stopmotion.verbose = FALSE) in your script or '.Rprofile'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
splice(images = images, insert = magick::wizard, after = 1)
```

---

stopmotion_verbosity *Control stopmotion verbosity*

Description

Convenience wrapper around options(stopmotion.verbose =) for enabling or disabling the frame-sequence messages printed after each operation. By default messages are shown in interactive sessions and suppressed in non-interactive contexts (e.g. knitr/Quarto rendering).

Usage

```
stopmotion_verbosity(verbose)
```

Arguments

verbose	TRUE to enable messages, FALSE to suppress them.

Value

the previous value of the option, invisibly.

Examples

```
old <- stopmotion_verbosity(FALSE)
on.exit(stopmotion_verbosity(old))
```

`trim` *Trim edges from images*

Description

Removes border pixels from selected frames by detecting the background colour and trimming uniform edges. Wraps `magick::image_trim`.

Usage

```
trim(images, fuzz = 0, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`fuzz`	a number in `[0, 100]` controlling colour tolerance when detecting the background. Higher values trim more aggressively.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
trim(images = images)
trim(images = images, fuzz = 10, frames = 1:3)
```

`wiggle` *Add a wiggle effect to frames*

Description

Inserts two rotated copies after each selected frame — one tilted `+degrees` and one tilted `-degrees` — creating a hand-held stop-motion rock effect. For large-angle permanent rotations, see `rotate`.

Usage

```
wiggle(images, degrees = 3, frames = NULL)
```

Arguments

`images`	an object of class `magick-image` to modify
`degrees`	a positive number specifying the tilt angle in degrees. Both `+degrees` and `-degrees` are applied automatically.
`frames`	integer vector of frame indices to duplicate. Defaults to `NULL`, which duplicates all frames.

Value

a `magick-image` object with 2 extra frames per selected frame.

Verbosity

After each operation a message listing the updated frame sequence is printed in interactive sessions. Use `stopmotion_verbosity(FALSE)` to suppress these messages, or set `options(stopmotion.verbose = FALSE)` in your script or '`.Rprofile`'.

Examples

```
dino_dir <- system.file("extdata", package = "stopmotion")
images <- read(dir = dino_dir)
wiggle(images = images, degrees = 3, frames = 1:2)
```

Index