

Package: flexpolyline (via r-universe)

September 10, 2024

Type Package

Title Flexible Polyline Encoding

Version 1.0.0

Description Binding to the C++ implementation of the flexible polyline encoding by [HERE](#) <<https://github.com/heremaps/flexible-polyline>>. The flexible polyline encoding is a lossy compressed representation of a list of coordinate pairs or coordinate triples. The encoding is achieved by: (1) Reducing the decimal digits of each value; (2) encoding only the offset from the previous point; (3) using variable length for each coordinate delta; and (4) using 64 URL-safe characters to display the result.

License GPL-3

URL <https://munterfi.github.io/flexpolyline/>,
<https://github.com/munterfi/flexpolyline/>

BugReports <https://github.com/munterfi/flexpolyline/issues/>

LinkingTo Rcpp

Imports Rcpp, sf (>= 0.9-3)

Suggests testthat (>= 3.0.0), stringr (>= 1.4.0), knitr (>= 1.28),
rmarkdown (>= 2.1), covr (>= 3.5.0)

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://munterfi.r-universe.dev>

RemoteUrl <https://github.com/munterfi/flexpolyline>

RemoteRef HEAD

RemoteSha 8b584a629ac9a4b098797bdc0672893bc90da3e

Contents

decode	2
decode_sf	3
encode	3
encode_sf	4
get_third_dimension	6
Index	7

decode	<i>Decode a flexible polyline encoded string</i>
--------	--

Description

This function calls `hf::polyline_decode` and `hf::get_third_dimension` of the C++ implementation of the flexible polyline encoding by HERE. Depending on the dimensions of the encoded line, a two or three-dimensional line is decoded.

Usage

```
decode(encoded)
```

Arguments

encoded character, encoded flexible polyline string.

Value

A matrix containing the coordinates of the decoded line.

Examples

```
# 2d line
decode("BFoz5xJ67i1B1B7PzIhaxL7Y")

# 3d line
decode("B1Boz5xJ67i1BU1B7PUzIhaUxL7YU")
```

decode_sf	<i>Wrapper function for decoding to simple features</i>
-----------	---

Description

A wrapper function for [decode](#) that converts the input polylines, encoded in the flexible polyline encoding, to simple feature geometries of the sf package.

Usage

```
decode_sf(encoded, crs = sf::NA_crs_)
```

Arguments

encoded	character, encoded flexible polyline string.
crs	integer or character, coordinate reference system to assign to the sf object (default = sf::NA_crs_).

Value

An sf object, containing the geometries of the decoded lines (Geometry type: "LINESTRING").

Note

The function returns an sf object, therefore the input set of encoded polylines must be of consistent dimension (e.g "XY", "XYM" or "XYZ") to meet the requirements of the constructor of sf objects. For mixed dimensions use the [decode](#) function directly.

Examples

```
decode_sf("B1Voz5xJ67i1Bgkh9B")
decode_sf("BFoz5xJ67i1B1B7P1U9yB")
decode_sf("B1Xoz5xJ67i1Bgkh9B1B7Pgkh9BzIhagkh9BqK-pB_ni6D")
```

encode	<i>Encode a line in the flexible polyline encoding format</i>
--------	---

Description

This function calls `hf::polyline_encode` of the C++ implementation of the flexible polyline encoding by [HERE](#). Depending on the dimensions of the input coordinates, a two or three-dimensional line is encoded.

Usage

```
encode(line, precision = 5L, third_dim = 3L, third_dim_precision = 5L)
```

Arguments

line	matrix, coordinates of the line in 2d or 3d (column order: LNG, LAT, DIM3).
precision	integer, precision to use in encoding (between 0 and 15, default=5).
third_dim	integer, type of the third dimension (0: ABSENT, 1: LEVEL, 2: ALTITUDE, 3: ELEVATION, 4, 6: CUSTOM1, 7: CUSTOM2, default=3).
third_dim_precision	integer, precision to use in encoding for the third dimension (between 1 and 15, default=5).

Value

The line as string in the flexible polyline encoding format.

Examples

```
# 2D
line2d <- matrix(
  c(8.69821, 50.10228,
    8.69567, 50.10201,
    8.69150, 50.10063,
    8.68752, 50.09878),
  ncol = 2, byrow = TRUE
)
encode(line2d)

# 3D
line3d <- matrix(
  c(8.69821, 50.10228, 10,
    8.69567, 50.10201, 20,
    8.69150, 50.10063, 30,
    8.68752, 50.09878, 40),
  ncol = 3, byrow = TRUE
)
encode(line3d)
```

encode_sf

Wrapper function for encoding simple features

Description

A wrapper function for [encode](#) that converts simple feature geometries of the sf package to flexible polyline encoded strings.

Usage

```
encode_sf(
  geom,
  precision = 5,
  third_dim = NULL,
  third_dim_precision = precision
)
```

Arguments

geom	simple feature, sf, sfc or sfg object with geometry type "POINT", "LINESTRING" or "POLYGON".
precision	integer, precision to use in encoding (between 0 and 15, default=5).
third_dim	integer, type of the third dimension (0: ABSENT, 1: LEVEL, 2: ALTITUDE, 3: ELEVATION, 4, 6: CUSTOM1, 7: CUSTOM2, default=NULL).
third_dim_precision	integer, precision to use in encoding for the third dimension (between 1 and 15, default=precision).

Value

The line as string in the flexible polyline encoding format.

Examples

```
# 3D point
point3d <- sf::st_point(
  matrix(c(8.69821, 50.10228, 10), ncol = 3, byrow = TRUE),
  dim = "XYZ"
)
encode_sf(point3d)

# 2D linestring
line2d <- sf::st_linestring(
  matrix(c(
    8.69821, 50.10228,
    8.69567, 50.10201,
    8.68752, 50.09878
  ), ncol = 2, byrow = TRUE)
)
encode_sf(line2d)

# 3D polygon
poly3d <- sf::st_polygon(list(
  matrix(c(
    8.69821, 50.10228, 10,
    8.69567, 50.10201, 20,
    8.69150, 50.10063, 30,
    8.69821, 50.10228, 10
  ), ncol = 3, byrow = TRUE)
)
```

```
), dim = "XYM")  
encode_sf(poly3d)
```

get_third_dimension *Get third dimension of a flexible polyline encoded string*

Description

This function calls `hf::get_third_dimension` of the C++ implementation of the flexible polyline encoding by HERE and return the type of the third dimension.

Usage

```
get_third_dimension(encoded)
```

Arguments

encoded character, encoded flexible polyline string.

Value

A string describing the third dimension.

Examples

```
# 2d line  
get_third_dimension("BFoz5xJ67i1B1B7PzIhaxL7Y")  
  
# 3d line  
get_third_dimension("B1Boz5xJ67i1BU1B7PUzIhaUxL7YU")
```

Index

decode, [2](#), [3](#)
decode_sf, [3](#)

encode, [3](#), [4](#)
encode_sf, [4](#)

get_third_dimension, [6](#)