

Package: cuvis.r (via r-universe)

October 14, 2026

Title R Bindings for the Cubert CUVIS Hyperspectral SDK

Version 0.1.0

Description Provides R bindings to the Cubert CUVIS C SDK for reading, processing, and exporting hyperspectral imaging data from Cubert snapshot cameras. Supports loading proprietary '.cu3s' session files, applying the full calibration pipeline (dark subtraction, white reference normalization, spectral radiance, reflectance), and exporting to standard formats (ENVI, TIFF). Requires the Cubert CUVIS SDK to be installed separately. This package mirrors the API of 'cuvis.python' (<<https://github.com/cubert-hyperspectral/cuvis.python>>) for R users.

License Apache License (>= 2) | file LICENSE

URL <https://github.com/cttir/cuvis.r>, <https://cttir.github.io/cuvis.r/>

BugReports <https://github.com/cttir/cuvis.r/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

SystemRequirements Cubert CUVIS SDK (>= 3.4.0) - download from <<https://cloud.cubert-gmbh.de/s/qpxkyWkycrmBK9m>>

Depends R (>= 4.1.0)

Imports cli, rlang

Suggests testthat (>= 3.0.0), withr, knitr, rmarkdown, covr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Config/Needs/website CTTIR/themakR

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-10-14 10:29:46 UTC

RemoteUrl <https://github.com/CTTIR/cuvis.r>

RemoteRef HEAD
RemoteSha ea199c4866c5ca78a98c743a3faf633a8835f6b1

Contents

cuvis_available	2
cuvis_export_envi	3
cuvis_export_session	3
cuvis_export_tiff	4
cuvis_get_cube	4
cuvis_get_measurement	5
cuvis_get_metadata	6
cuvis_get_wavelengths	6
cuvis_has_reference	7
cuvis_init	7
cuvis_is_initialized	8
cuvis_processing_context	9
cuvis_processing_modes	9
cuvis_reference_types	10
cuvis_reprocess	10
cuvis_sample_data	11
cuvis_sample_session	11
cuvis_session	12
cuvis_set_reference	12
cuvis_shutdown	13
cuvis_version	13

Index **14**

cuvis_available	<i>Check if CUVIS SDK is Available</i>
-----------------	--

Description

Tests whether the Cubert CUVIS SDK library is linked and usable.

Usage

cuvis_available()

Value

Logical. TRUE if the SDK is available, FALSE otherwise.

cuvis_export_envi	<i>Export a Measurement to ENVI Format</i>
-------------------	--

Description

Writes the measurement data as an ENVI header/binary file pair.

Usage

```
cuvis_export_envi(measurement, dir, permissive = TRUE)
```

Arguments

measurement	A cuvis_measurement object.
dir	Character. Output directory.
permissive	Logical. Allow export even if some data is missing. Default TRUE.

Value

Invisible dir.

cuvis_export_session	<i>Export a Measurement to a New Session File</i>
----------------------	---

Description

Saves the measurement as a new .cu3s session file.

Usage

```
cuvis_export_session(measurement, dir, permissive = TRUE)
```

Arguments

measurement	A cuvis_measurement object.
dir	Character. Output directory for the .cu3s file.
permissive	Logical. Default TRUE.

Value

Invisible dir.

cuvis_export_tiff	<i>Export a Measurement to TIFF Format</i>
-------------------	--

Description

Export a Measurement to TIFF Format

Usage

```
cuvis_export_tiff(
  measurement,
  dir,
  format = c("MultiChannel", "Single", "MultiPage"),
  compression = c("None", "LZW"),
  permissive = TRUE
)
```

Arguments

measurement	A cuvis_measurement object.
dir	Character. Output directory.
format	Character. TIFF format: "MultiChannel" (default), "Single", or "MultiPage".
compression	Character. Compression mode: "None" (default) or "LZW".
permissive	Logical. Default TRUE.

Value

Invisible dir.

cuvis_get_cube	<i>Extract Hyperspectral Cube from a Measurement</i>
----------------	--

Description

Returns the 3D data cube as an R array with wavelength metadata.

Usage

```
cuvis_get_cube(measurement)
```

Arguments

measurement	A cuvis_measurement object.
-------------	-----------------------------

Value

A numeric 3D array with dimensions [rows, cols, bands]. The "wavelengths" attribute contains a numeric vector of band center wavelengths in nanometers.

Examples

```
## Not run:
cuvis_init()
session <- cuvis_session("path/to/file.cu3s")
mesu <- cuvis_get_measurement(session, 1)
cube <- cuvis_get_cube(mesu)
dim(cube)                # e.g., c(100, 100, 61)
attr(cube, "wavelengths") # wavelengths in nm
cuvis_shutdown()

## End(Not run)
```

cuvis_get_measurement *Get a Measurement from a Session*

Description

Extracts a single measurement from a session file by index.

Usage

```
cuvis_get_measurement(session, index = 1L, item_type = "no_gaps")
```

Arguments

session	A cuvis_session object.
index	Integer. 1-based measurement index.
item_type	Character. Type of items to enumerate: "no_gaps" (default), "all_frames", or "references".

Value

A cuvis_measurement S3 object, or NULL if no measurement exists at that index.

cuvis_get_metadata	<i>Get Measurement Metadata</i>
--------------------	---------------------------------

Description

Returns metadata fields for a measurement including name, integration time, serial number, processing mode, and more.

Usage

```
cuvis_get_metadata(measurement)
```

Arguments

measurement	A cuvis_measurement object.
-------------	-----------------------------

Value

A named list with fields: name, path, comment, integration_time (ms), averages, distance, serial_number, product_name, assembly, processing_mode (integer), capture_time (ms since epoch), measurement_flags (integer).

cuvis_get_wavelengths	<i>Get Wavelength Vector from a Measurement</i>
-----------------------	---

Description

Convenience function to extract just the wavelength vector from a measurement's cube data.

Usage

```
cuvis_get_wavelengths(measurement)
```

Arguments

measurement	A cuvis_measurement object.
-------------	-----------------------------

Value

Numeric vector of wavelengths in nm.

cuvis_has_reference	<i>Check if a Reference is Set</i>
---------------------	------------------------------------

Description

Check if a Reference is Set

Usage

```
cuvis_has_reference(  
  ctx,  
  type = c("dark", "white", "white_dark", "sprad", "distance")  
)
```

Arguments

ctx	A cuvis_processing_context object.
type	Character. Reference type to check.

Value

Logical.

cuvis_init	<i>Initialize the CUVIS SDK</i>
------------	---------------------------------

Description

Must be called before any other cuvis.r function. Loads the Cubert CUVIS SDK and configures the user settings directory.

Usage

```
cuvis_init(settings_dir = NULL)
```

Arguments

settings_dir	Character. Path to the CUVIS settings directory — the one holding factory/ and user/. If NULL (default), the directory is discovered in this order: the CUVIS_SETTINGS environment variable, then the standard install locations for the platform, and only as a last resort a temporary directory.
--------------	---

Details

The settings directory is not optional in practice. The SDK loads its factory calibration from it, and pointing it at an empty directory makes every subsequent `cuvis_proc_cont_apply()` fail with "Could not process measurement" — which looks like a corrupt recording rather than a configuration problem. Hence the search over real install locations before falling back to a temporary directory, and the warning when it does.

Value

Invisible NULL.

Examples

```
## Not run:  
cuvis_init()  
# ... use SDK ...  
cuvis_shutdown()  
  
## End(Not run)
```

cuvis_is_initialized	<i>Is the CUVIS SDK Currently Initialised?</i>
----------------------	--

Description

Is the CUVIS SDK Currently Initialised?

Usage

```
cuvis_is_initialized()
```

Value

Logical.

Examples

```
cuvis_is_initialized()
```

`cuvis_processing_context`*Create a Processing Context for Calibration*

Description

Creates a processing context from a session file or measurement. This context holds calibration references and processing parameters for reprocessing measurements.

Usage

```
cuvis_processing_context(base, load_references = TRUE)
```

Arguments

<code>base</code>	A <code>cuvis_session</code> or <code>cuvis_measurement</code> object.
<code>load_references</code>	Logical. If TRUE (default), automatically loads references embedded in the session file.

Value

A `cuvis_processing_context` S3 object.

`cuvis_processing_modes`*CUVIS Processing Modes*

Description

Named integer vector of processing modes matching the CUVIS SDK C enum values. Used internally by `cuvis_reprocess()`.

Usage

```
cuvis_processing_modes
```

Details

- Raw (0): Raw sensor data, no processing
- DarkSubtract (1): Dark current subtracted
- Reflectance (2): Normalized reflectance (0-1)
- SpectralRadiance (3): Calibrated to physical radiance (W/m²/sr/nm)
- Preview (5): Low-resolution preview only

cuvis_reference_types *CUVIS Reference Types*

Description

Named integer vector of reference types matching the CUVIS SDK C enum values. Used by [cuvis_set_reference\(\)](#).

Usage

```
cuvis_reference_types
```

Details

- Dark (0): Dark reference frame
- White (1): White reference frame
- WhiteDark (2): Dark reference for the white reference
- SpRad (3): Spectral radiance calibration
- Distance (4): Distance calibration

cuvis_reprocess *Reprocess a Measurement to a New Calibration Level*

Description

Applies processing context settings to a measurement, converting it to the requested calibration level (e.g., reflectance).

Usage

```
cuvis_reprocess(
    ctx,
    measurement,
    mode = c("reflectance", "spectral_radiance", "dark_subtract", "raw", "preview"),
    allow_recalib = FALSE
)
```

Arguments

ctx	A cuvis_processing_context with references set.
measurement	A cuvis_measurement to reprocess.
mode	Character. Target processing mode: "reflectance" (default), "spectral_radiance", "dark_subtract", "raw", or "preview".
allow_recalib	Logical. Allow recalibration if needed. Default FALSE.

Value

The same cuvis_measurement object, reprocessed in place. Extract the cube with `cuvis_get_cube()`.

cuvis_sample_data	<i>Get Path to Sample Data</i>
-------------------	--------------------------------

Description

Returns the path to Cubert sample data. Searches in order:

1. .cuvis/ directory in the working directory or any parent (up to 5 levels)
2. CUVIS_DATA environment variable
3. SDK default install location

Usage

```
cuvis_sample_data(file = NULL)
```

Arguments

file	Character. Specific file name (e.g., "test_mesu.cu3s"). If NULL, returns the sample data root directory.
------	--

Value

Character path, or NULL if not found.

cuvis_sample_session	<i>Load the Default Sample Session</i>
----------------------	--

Description

Convenience function that locates and loads the first .cu3s file found in the sample data directories.

Usage

```
cuvis_sample_session()
```

Value

A cuvis_session object, or NULL with a warning if sample data is not found.

cuvis_session	<i>Load a Cubert Session File</i>
---------------	-----------------------------------

Description

Opens a .cu3s session file containing one or more hyperspectral measurements with their calibration references.

Usage

```
cuvis_session(path)
```

Arguments

path	Character. Path to a .cu3s session file.
------	--

Value

A cuvis_session S3 object (wrapping an external pointer to the SDK session handle). Use [cuvis_get_measurement\(\)](#) to extract individual measurements.

Examples

```
## Not run:
cuvis_init()
session <- cuvis_session("path/to/measurement.cu3s")
print(session)
cuvis_shutdown()

## End(Not run)
```

cuvis_set_reference	<i>Set a Calibration Reference</i>
---------------------	------------------------------------

Description

Assigns a reference measurement (dark, white, etc.) to a processing context.

Usage

```
cuvis_set_reference(
  ctx,
  measurement,
  type = c("dark", "white", "white_dark", "sprad", "distance")
)
```

Arguments

ctx	A cuvis_processing_context object.
measurement	A cuvis_measurement object (the reference frame).
type	Character. One of "dark", "white", "white_dark", "sprad", or "distance".

Value

Invisible ctx (modified in place via external pointer).

cuvis_shutdown	<i>Shut Down the CUVIS SDK</i>
----------------	--------------------------------

Description

Releases all SDK resources. Call when done using cuvis.r functions.

Usage

```
cuvis_shutdown()
```

Value

Invisible NULL.

cuvis_version	<i>Get CUVIS SDK Version</i>
---------------	------------------------------

Description

Note: The SDK must be initialized with `cuvis_init()` before calling this function.

Usage

```
cuvis_version()
```

Value

Character string with the SDK version.

Index

cuvis_available, [2](#)
cuvis_export_envi, [3](#)
cuvis_export_session, [3](#)
cuvis_export_tiff, [4](#)
cuvis_get_cube, [4](#)
cuvis_get_cube(), [11](#)
cuvis_get_measurement, [5](#)
cuvis_get_measurement(), [12](#)
cuvis_get_metadata, [6](#)
cuvis_get_wavelengths, [6](#)
cuvis_has_reference, [7](#)
cuvis_init, [7](#)
cuvis_init(), [13](#)
cuvis_is_initialized, [8](#)
cuvis_processing_context, [9](#)
cuvis_processing_modes, [9](#)
cuvis_reference_types, [10](#)
cuvis_reprocess, [10](#)
cuvis_reprocess(), [9](#)
cuvis_sample_data, [11](#)
cuvis_sample_session, [11](#)
cuvis_session, [12](#)
cuvis_set_reference, [12](#)
cuvis_set_reference(), [10](#)
cuvis_shutdown, [13](#)
cuvis_version, [13](#)