



REAL-TIME SPECTRAL IMAGING



Smart control suite for hyperspectral Cameras

Your benefits

- Easy to use user interface
- Record and analyse hyperspectral images
- Full remote control of the camera
- Live view
- Application programming interface
- Source code for Matlab and Python applications

Requirements

Hardware Requirements	
Processor	64 bit capable 64-bit Intel or AMD
Memory	8 GB
Hard Disk	At least 10 GB free
Network	10/100 Ethernet
Data Flow	
Operating System	Windows 10
Permissions	Full Administrator
	SQL
Optional Software	
Operating System	Linux or MacOS
Matlab SDK	Matlab 7.1
Python SDK	Python 2.7
C Link	
Operating System	Windows 10 64-bit
Permissions	Full Administrator

CUBERT UTILS TOUCH



1 Content

1	CONTENT	2
2	GENERAL CONCEPT	4
2.1	LOCATION	4
2.2	AVAILABILITY	4
2.3	DEPENDENCIES AND VERSIONING.....	4
3	GENERAL SYNTAX	5
3.1	USING THE SCHEMA DEFINITION	5
3.2	DOCUMENT LAYOUT.....	5
4	NON-EXPRESSION TAG DETAILS	6
4.1	TAG CONFIGURATION	6
4.2	TAG COMMENT	6
4.3	TAG INPUT	6
4.4	TAG GRAY	7
4.5	TAG R / G / B.....	7
4.6	TAG LABELED	8
4.7	TAG ASSIGN	8
4.8	TAG OP.....	8
4.9	TAG RESOURCE.....	8
5	EXPRESSION-TYPE TAG DETAILS.....	9
5.1	TAG VALUE	9
5.2	TAG VARIABLE.....	9
5.3	TAG RANGEMAX	9
5.4	TAG RANGE	9
5.5	TAG THRESHOLD.....	9
5.6	TAG OPERATOR	10
5.7	TAG CLASSIFIER	10

PREFACE

Thank you for choosing this Cubert GmbH product!

This guide provides details about programming you own User Plug-Ins.

Please keep this document at a safe location.

News and current information on this product can be found on our official webpage:

www.cubert-gmbh.com

2 General Concept

The file location is `<INSTALL DIR>\Bin\user\plugin\<MODE>`

2.1 Location

The User Plug-In is stored in an XML-format. For activating it, it is only required to pass on the content via the SDK as string. The SDK command is `view_setUserPlugin`, as `configuration` argument you simply pass the whole User Plug-In.

The GUI Cubert Utils Touch, however, loads user-plug-ins from disk and offers to transmit them to Cubert Utils Core via the user interface.

The file location is:

`<INSTALL DIR>\Bin\user\plugin\<MODE>`,

where mode is `raw`, `ds`, `ref` and `sprad` for raw, dark-subtract, reflectance and spectral radiance processing mode respectively. If these folders do not exist, you may create them.

2.2 Availability

The User Plug-Ins can be set in any capture mode (processing mode) except sensor raw. However, it is recommended to use user plugins for one mode exclusively as the mathematical expressions may make no or only limited sense for a different processing mode.

2.3 Dependencies and Versioning

The User Plug-Ins are upwards compatible. The version of the engine version can be checked via SDK with the `view_getUserPluginVersion` command. As yet, all plug-ins will operate with all Cubert systems on any supported platform, even if operated via network.

One limitation to this is the utilization of **perClass** classification engine. Here, you must ensure that Cubert Utils Core has a valid RUNTIME license. The creation of the user plugin, as well as sending form either SDK or Cubert Utils Touch does not require any **perClass** license.

3 General Syntax

The schema location is `<INSTALL DIR>\Bin\userplugin.xsd`

3.1 Using the Schema definition

The schema of a User Plug-In is defined by the `userplugin.xsd` file which is located in `<INSTALL DIR>\Bin\userplugin.xsd`

We recommend using a schema-aware XML-Editor when writing a User Plug-In with code highlighting, e.g. the Visual Studio XML editor.

In such editors you need to specify the schema location, where you would point to the `userplugin.xsd` file.

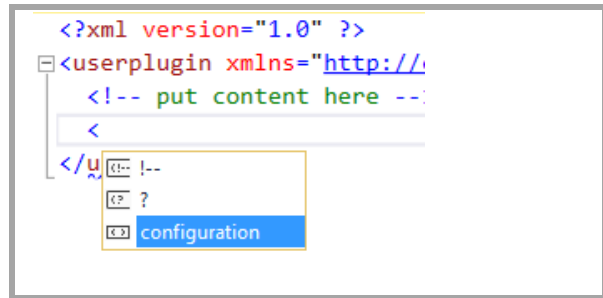
When no proper XML editor is available, a normal text editor will do, as XML files are basically plain text files with some syntax constraints.

3.2 Document layout

In order to get started with the User Plug-In, you generate a text file with the following content:

```
<?xml version="1.0" ?>
<userplugin xmlns="http://cubert-
gmbh.de/user/plugin/userplugin.xsd">
  <!-- put content here -->
</userplugin>
```

Now, with the XML-Editor you can set the XML schema location (if possible). The code highlighting should offer the next valid tags when open the left angle bracket:



The calculation is located in the configuration tag.

The configuration tag contains the actual content. First there is an optional *comment* tag, then any number of (optional) *input* fields and finally the display. The display can either be *gray* or *R/G/B* or *labeled*, e.g.

```
<?xml version="1.0" ?>
<userplugin xmlns="http://cubert-
gmbh.de/user/plugin/userplugin.xsd">
  <!-- put content here -->
  <configuration>
    <comment>some comment</comment>
    <input></input>
    <gray></gray>
  </configuration>
</userplugin>
```

or

```
<?xml version="1.0" ?>
<userplugin xmlns="http://cubert-
gmbh.de/user/plugin/userplugin.xsd">
  <!-- put content here -->
  <configuration>
    <comment>some comment</comment>
    <input></input>
    <R></R>
    <G></G>
    <B></B>
  </configuration>
</userplugin>
```

Please note, that the examples are not valid, because they miss sub-tags or attributes (see next chapter).

4 Non-expression tag details

4.1 Tag configuration

The *configuration* tag has the following attributes:

- *citation*
(optional) Give a citation token (no space symbols allowed)
- *name*
(required) Give a name, only letters and numbers and underscore, between 2 and 20 characters.
- *plugin_author*
(optional) Give an author token (no space symbols allowed, between 4 and 30 tokens.
- *plugin_version*
(required) Either an integer number or 3 integer numbers separated by dots, e.g. 1 or 1.2.3
- *required_engine_version*
(required) The exact version string from the engine, e.g. 2.1.0
- *comment*
This field is a special comment with any text, max 400 signs that will be displayed directly under the plug-in name in the GUI. It is not recommended to use this tag, as it will lead to an overloaded GUI dialogue. Instead use the *comment* tag.

The configuration tag contains in the precise order:

- zero or one *comment* tag

- zero or any number of *input* tags
- either
 - a *gray* tag or
 - first a **R** tag, next a **G** tag and finally a **B** tag or
 - a *labeled* tag

4.2 Tag comment

The *comment* tag is a string of any text (beware the escape syntax for XML-commands) of the maximum length of 400 characters.

4.3 Tag input

The *input* tag is used to store variables. These variables may be visible in the UI and even editable in run-time with some limited displaying and validation options.

The content of the tag is a scalar number (integer for type="wavelength", floating point for type="scalar" and HTML-Style color code for type="color" (e.g. #FFee23))

It has the following attributes:

- *id*
(required) Give each input a unique name by which it can be referenced, only letters and numbers and underscore, between 2 and twelve characters are allowed.
- *type*
(required) This defines how the

calculation will treat this input.
Options are:

- **wavelength**
A wavelength input. The integer number will be interpreted as wavelength. When this input is referenced by a formula, it will be replaced by the slice of the current cube with the wavelength (i.e. a monochromatic image). For each new image, it will be evaluated again.
- **scalar**
A simple scalar value input.
- **color**
Provide a color code in HTML style, e.g. #1253FF, that can be used for the *labeled* display tag.
- **hide**
(optional) Set to "true" to hide the input from the GUI.
- **readOnly**
(optional) "true" if the input shall be read-only.
- **min**
(optional) Give a minimum scalar value. The GUI will not allow inputs below that value. Ignored for type="color"
- **max**
(optional) Give a minimum scalar value. The GUI will not allow inputs below that value. Ignored for type="color"

- **tickFreq**
(optional) If max and min are also defined, the input text box will be replaced by a slider if a positive integer tick frequency is given.

4.4 Tag gray

The gray tag contains one expression-type tag element. It has the following attributes:

- **colormap**
(optional, default: gray) Select a colormap out of the following list: **gray**, **gray_inverted** or **hot**.
- **min**
(optional) Truncates calculation result values below to this value. It will be clipped to the 0-value of the colormap.
- **max**
(optional) Truncates calculation result values above to this value. It will be clipped to the 255-value of the colormap.
- **discretize**
(optional) If the value is set to "true", the values be clipped to the closest value of **min** or **max**. As such, only the first and the last value of the color map are shown.

4.5 Tag R / G / B

The tags **R**, **G** and **B** are separate tags which must follow each other in the exact order. They have no specific attribute and contain one expression-type tag element.

4.6 Tag labeled

The *labeled* tag is a special kind of color map displaying element. It allows to manually assigning a color to numeric input values.

The first child element must be an expression-type node. The calculation result should be integer, if not the results are rounded to integer values.

Followed by this, any number of *assign* tags can be given.

4.7 Tag assign

The *assign* tag can only be used after the expression-type node.

Is has the following attributes:

- *channel*
(required) The numeric value from the calculation that we want to assign a color to
- *color*
(required - or use *ref_color* instead)
The HTML-color code to use for the selected channel.
- *ref_color*
(required - or use *color* instead)
The *id* name of an *input* element with exactly this name. The *input type* must be "color"

4.8 Tag op

The *op* tag can be used within a *classifier* tag (see below). It defines the ROC operation point. The classifier needs to have the ROC selection stored within the pipeline. The content is either a *variable* or a *value* tag element.

- *idx*
(required) The number of ROC curve, starting to count at zero.

4.9 Tag resource

The *resource* tag can be used within a *classifier* tag (see below). The content is the base64-encoded nested data for the classifier engine.

5 Expression-type tag details

5.1 Tag *value*

The tag has no attributes. The content is a static floating-point value.

5.2 Tag *variable*

A variable has no content, but a *ref* attribute. The variable will be replaced by whatever it points to. If it points to a scalar-type input,

5.3 Tag *rangemax*

The *rangemax* tag is like a *variable* tag. Its value will be refreshed depending on the maximum value of the current dynamic range of the cube to be processed, e.g. 255 for a 8 bit cube. The behavior for floating-point values is not defined.

Reflectance's *rangemax* is 65534, 100% is 10000 counts for 16 bit cubes (default).

5.4 Tag *range*

The *range* tag averages the cube between two given spectral positions. If a wavelength boundary is not precisely an existing wavelength of the current cube, the closest wavelength in any direction will be used. As such, even a range completely off the current cube's spectral range will still lead to one channel selected.

The selected channels will be averaged.

The valid attributes are:

- *ref_min*
(required) A reference to valid wavelength-type *input* tag. The wavelength will be selected as lower boundary of the spectral range.
- *ref_max*
(required) A reference to valid wavelength-type *input* tag. The wavelength will be selected as upper boundary of the spectral range.

5.5 Tag *threshold*

The *threshold* element will truncate data above or below a certain value and assign a different value to it. This is useful to avoid divisions by zero or to cut-off noisy data at low counts.

This tag must have exactly one expression-type tag.

The *threshold* tag has the following attributes:

- *threshold*
(required) The threshold value (floating-point)
- *lowpass*
(required) Acts as low-pass, when true, and as high-pass when false.
- *defaultto*
(required) The value cut-off data will be set to.

A *operator* tag division will not avoid divisions by zero.
Use the *threshold* tag, if you cannot guarantee non-zero divisors.

5.6 Tag operator

The *operator* tag contains two expression-type sub-tags, the two operands. The order is crucial for division and subtraction.

The *operator* tag has the following attributes:

- *type*
(required) The operation to be performed: “add”, “subtract”, “divide” or “multiply”

5.7 Tag classifier

The *classifier* tag allows to pass cube data to a separate classifier engine. The engine is PR Sys design's **perClass**.

The content is first zero or more *op* tags followed by a mandatory *resource* tag.

The attribute *engine* must be set to the value “perClass”.

6. Support

In general, the technical support for Cubert hyperspectral cameras is provided by your local distributor.

Europe:

Germany

SphereOptics GmbH
88221 Herrsching, Gewerbestraße 13
Contact partner: Hans Jörg Fischer
E-mail: hfischer@sphereoptics.de
Phone: +49 8152 983 7890

Italy

3D Target S.R.L.
Via Del Mella 76 5131 Brescia (BS)
Contact partner: Paolo Giardini
E-mail: sales@3dtarget.it
Phone: +39 393 935 8869

Netherlands & Turkey

MS MacroSystem
3951 CL Maarn, Tromplaan 1
Contact partner: Mu Elmas
E-mail: m.elmas@planet.nl
Phone: +31 343 442 969

Russia

SOVZOND Company Ltd.
115563 Moscow, Shipilovskaya street
28a
Contact partner: Kristina Uryupina
E-mail: k.uryupina@sovzond.ru
Phone: +7 965 286 51 86

Spain

Bonsai Advanced Technologies100085
28108 Alcobendas Madrid,
Avda Valdelappra 27 Edificio I bajo 3
Contact partner: Fernando Lopez
E-mail:
fernando.lopez@bonsaiadvanced.com
Phone: +34 91 4902334

United Kingdom

Pro-Lite Technology Ltd.
Cranfield Innovation Centre
MK43 OBT Cranfield University Way
Contact partner:
Ian Stansfield (CEO), Nick Barnett
E-mail: ian.stansfield@pro-lite.co.uk,
nick.barnett@pro-lite.co.uk
Phone: +44 0 1234 436110

Asia:

China

Azup International Group Ltd.
100085 Beijing, 2 Shangdi Xinx Road
Contact partner: Anne Xu, Aaron Zhang
E-mail: trading@azup.com.cn,
azupbj@azup.com.cn
Phone: 86 10 6211 1182

Japan

ARGO Corporation
Interplanet Esaka Building 9F,
1-13-48 Esaka-cho Suita Osaka 564-
0063
Contact partner: Toshinori Matsui
E-mail: matsui@argocorp.com
Phone: +816 6339 3366

South Korea

Wonwoo Systems Co., Ltd.
Santevill 106-209
Shindaebang 1 ga-gil 38
07072 Seoul Dongjak-gu
Contact partner: Won Kyun
E-mail: mkkwon@wonwoosystems.co.kr
Phone: 822 533 6720

United States:

USA

Bodkin Design & Engineering
77 Oak Street, Suite 201
MA02481 Wellesley
Contact partner: Andrew Bodkin
E-mail: wab@bodkindesign.com
Phone: +617 795 1968 x115

Australia:

Arborcarbon Pty Ltd.

PO. Box 1065 Central
WA 6156 Willagee
10 Weaver Court Kardinya WA6163
Contact partner: Paul Barber
E-mail: p.barber@arborcarbon.com.au
Phone: +61 8 9467 9876