Application Note

Using Look-Up Tables with Goldeye and Goldeye Pro Cameras

V2.0.0
2025-Nov-14

Scope of this document

This application note explains how to use look-up tables (LUTs) for image processing. Goldeye and Goldeye Pro cameras provide 4 pre-configured and 4 user-configurable LUT files.

Additional documentation

You can find camera user guides and feature references at www.alliedvision.com/en/support/technical-documentation.

Basics about LUTs

A LUT is used to remap pixel counts. In Figure 1, 16383 is mapped to 15612. The other values are mapped correspondingly.

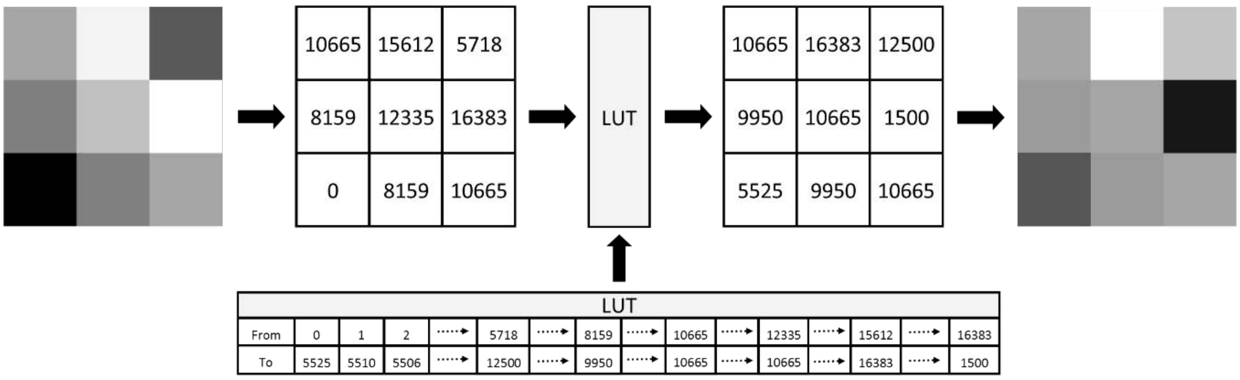

Figure 1: LUT applied to a set of pixels

The LUT consists of a series $\boldsymbol{a_i}$ of 2^{14} (= 16384) count values, therefore the mapping is defined to map each pixel of value $\boldsymbol{i}$ to value $\boldsymbol{a_i}$.

Expressed in pure mathematically terms: May $(a_i)_{i\in N}$ be a series of integers with $(a_i)_{i\in N} \in M$

with $N = [0{:}\,16383] \subset \mathbb{N}_0$ and $N_1, M \subseteq N$.

The LUT is defined by the mapping $\mathcal{L}: N_1 \to M, b \to a_b$, with:

b = Input pixel value
a_b = Output pixel value

Note: The mapping may not necessarily be surjective.

For the example above, the LUT must contain 16383 values to define the mapping for each possible gray value. The values must be integer values in a range from 0 to 16383. However, pixels of different gray input values may be mapped to identical gray output values, for example a LUT may contain the same value more than once (see Figure 1) or not at all.

Usage of LUTs

Observe that the LUT follows after modules the image processing chain, such as the Defective Pixel Correction (DPC), at a bit depth of 14-bit. If this signal is conversed to an 8-bit output pixel format, this occurs at the very end of the image processing chain. See the corresponding camera user guide for details.

Using LUT features

The LUT is controlled by the GenICam features shown in Figure 2. Use the features as described below.

1. Enter the LUT number into `LUTDatasetSelector` to select the corresponding LUT data set.
2. Send a command by `LUTDatasetLoad` to load the data set to be processed.
3. Use `LUTDatasetActive` to display the currently loaded data set.
4. Set `LUTEnable` to ***true*** for applying the LUT.

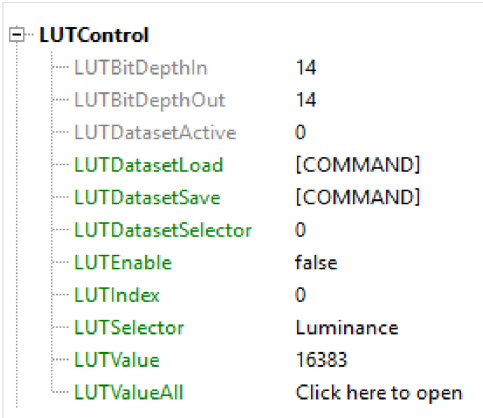

Figure 2: Features for LUT control

5. Check that pixel values have been transferred as desired:

- Either Use `LUTIndex` to display the source of the currently loaded LUT.
- Or use `LUTValue` to display the target of the currently loaded LUT.
- Or use `LUTValueAll`to ouput the corresponding binary data.

LUT factory settings

Table 1 shows LUT values for Goldeye and Goldeye Pro cameras.

The inverting LUT maps each pixel value ***i*** to ***j = 16383 - i***.

The remaining three LUTs apply the listed gamma corrections to the image.

Predefined		**Customizable**	
LUT	**Effect**	**LUT**	**Effect**
0	Inverting	4	(none)
1	Gamma 1.16	5	(none)
2	Gamma 1.18	6	(none)
3	Gamma 1.20	7	(none)

Table 1: LUT factory settings

Pre-configured LUTs: `LUTDatasetSave` is not available because the data sets allow read access only.

User configurable LUTs: You can fill each value with arbitrary numbers, as described below.

Data structure of LUTs

The LUTs of Goldeye and Goldeye Pro cameras are given by binary 14-bit values, for example 2 bytes (= 16 bit) for each entry. Therefore, each LUT has a constant size of 16384 x 2 bytes = 32768 bytes.

The byte order is little endian (MSB): The most significant bytes are stored first, the most significant bytes are stored last. For the LUT table shown in Figure 1 on page 1, only the resulting values are saved successively.

Figure 3 shows the source values given by the position of the 2 bytes in the binary LUT data. .

Example

This example is marked with blue frames in Figure 3.

- The 10th value of LUT 0 is ***16374***, which is the inverted value of ***9***.
- The hexadecimal value of ***16374*** is ***0x3FF6***.
- Because of the little-endianness, the value is stored as ***F6 3F*** in the LUT.

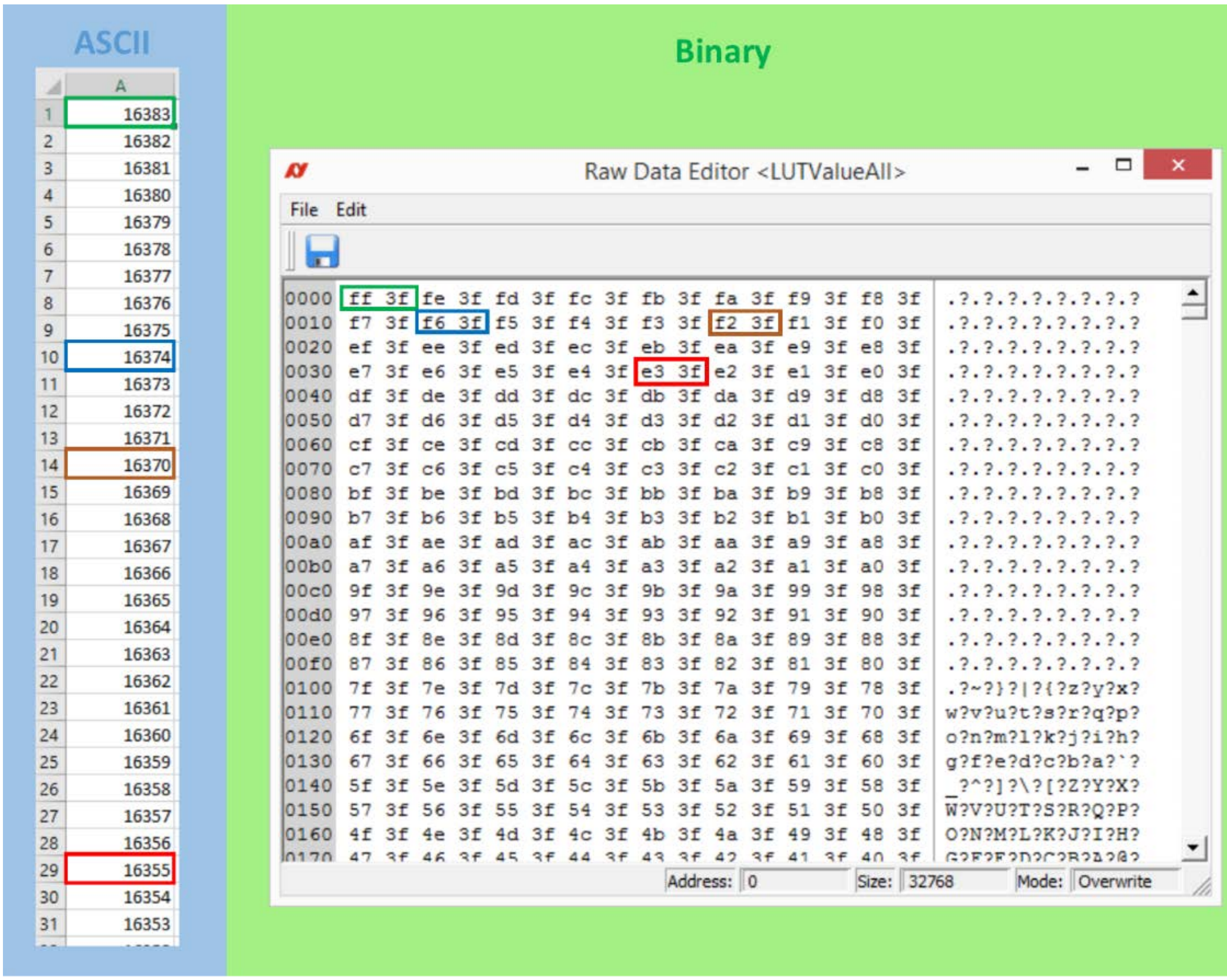

Figure 3: Predefined LUT No. 0 (= inversing LUT) in ASCII and binary representation

Uploading LUTs to the camera

Select from these options:

- Set the ASCII LUT value for each index one by one.
- Upload the binary data at once.
- Upload the binary data by direct file access.

Setting ASCII LUT values for each index

Note: Observe that this approach takes more time to write or read LUT data than Uploading binary LUT data at once.

To set individual ASCII LUT values, follow the steps below:

1. Set `LUTDatasetSelector` to a value between *4* and *7*, to select one of the user configurable LUTs.
2. Execute `LUTDatasetLoad` to load the LUT.
   `LUTDatasetActive` signals which LUT is currently loaded in the camera to be applied or modified.
3. Use `LUTIndex` to define a source value of the lookup table to be modified.
   See Using LUT features on page 2.
4. Use `LUTValue` to set the target value.
5. Repeat steps 1 to 4 for all LUT entries to be modified.
6. Execute `LUTDatasetSave` to save the LUT dataset.

Uploading binary LUT data at once

1. Set `LUTDatasetSelector` to a value between *4* and *7*, to select one of the user configurable LUTs.
2. Execute `LUTDatasetLoad` to load the LUT.
   `LUTDatasetActive` signals which LUT is currently loaded in the camera to be applied or modified.
3. Use `LUTValueAll` to access the LUT's binary data.
   This feature modifies the data by direct memory access.

Using Vimba X SDK

When **Vimba X** is used, the binary LUT data is stored in a variable of type **UcharVector** of size 32768 (2 bytes for 16384 LUT entries) in little endian byte order. Access this feature as shown below.

```
// forsimplicity, error handling has been omitted
// start Vimba
VimbaSystem &sys = VimbaSystem::GetInstance();
sys.Startup();
// get pointers to connected cameras
CameraPtrVector vpCamera;
sys.GetCameras( vpCamera );
```

Code Example 1: (sheet 1 of 2)

```
// open first cam
vpCamera[0]->Open( VmbAccessModeFull );

UcharVector LUTdata( 32768,0 ); // vector containing binary LUT data -> 32768 bytes
// in this case all bytes have been initialized to 0
// fill vector with binary LUT data
// …
FeaturePtr feature; // pointer for feature access
vpCamera[0]->GetFeatureByName( "LUTValueAll", feature ); // get feature
feature->SetValue( LUTdata ); // upload LUT data
vpCamera[0]->GetFeatureByName( "LUTDatasetSave", feature ); // get feature
feature-> RunCommand(); // save LUT data

// The data can also be read from the memory by //
vpCamera[0]->GetFeatureByName("LUTDatasetLoad", feature ); // get feature
feature-> RunCommand(); // load LUT data
vpCamera[0]->GetFeatureByName( "LUTValueAll", feature ); // get feature
feature->GetValue( data ); // download LUT from camera to UcharVector data
```

Code Example 1: (sheet 2 of 2)

When finished, save the LUT data set by executing `LUTDatasetSave`.

Alternatively, you can read the data from the memory using the following command:

```
feature->GetValue( data ); // download LUT from camera to UcharVector data
```

Downloads for tools and programming examples

Get helpful resources:

- A tool to simplify uploading binary data by file access.
- Information on using LUTs with Goldeye and Goldeye Pro cameras, including programming examples for the access modes described above.

Please contact the Allied Vision Support team at www.alliedvision.com/en/about-us/contact-us/technical-support-repair-/-rma.

Contact us

Website, email

General
www.alliedvision.com/en/contact
info@alliedvision.com

Distribution partners
www.alliedvision.com/en/avt-locations/avt-distributors

Support
www.alliedvision.com/en/support
www.alliedvision.com/en/about-us/contact-us/technical-support-repair-/-rma

Offices

Europe, Middle East, and Africa (Headquarters)
Allied Vision Technologies GmbH
Taschenweg 2a
07646 Stadtroda, Germany
T// +49 36428 677-0 (Reception)
T// +49 36428 677-230 (Sales)
F// +49 36428 677-28

North, Central, and South America, Canada
Allied Vision Technologies Canada Inc.
300 – 4621 Canada Way
Burnaby, BC V5G 4X8, Canada
T// +1 604 875 8855

USA
Allied Vision Technologies, Inc.
102 Pickering Way- Suite 502
Exton, PA 19341, USA
Toll-free// +1-877-USA-1394
T// +1 978 225 2030

Asia-Pacific
China
Allied Vision Technologies Shanghai Co Ltd.
B-510, Venture International Business Park
2679 Hechuan Road
Minhang District, Shanghai 201103
People's Republic of China
T// +86 21 64861133

Japan
Allied Vision Technologies
Yokohama Portside Bldg. 10F
8-1 Sakae-cho, Kanagawa-ku
Yokohama-shi, Kanagawa, 221-0052
T// +81 (0) 45 577 9527

Singapore
Allied Vision Technologies Asia Pte. Ltd
82 Playfair Rd, #07-01 D'Lithium
Singapore 368001
T// +65 6634 9027

Liability, trademarks, and copyright

Allied Vision has tested the product under the described conditions. The customer assumes all risk of product damage, application compromise or potential failure, and Sales Warranty loss related to deviation from the described conditions. Allied Vision's acknowledgement of such deviations in the customer's modified product or applications does not constitute advice for use. No Warranty is offered or implied by Allied Vision regarding the customer's assumed risk or legal responsibilities with such modified products or applications.