Teledyne DALSA • 880 Rue McCaffrey • St-Laurent, Québec, H4T 2C7 • Canada
https://www.teledynedalsa.com/

G3-AN0010-V1: RLE for Genie Nano GigE Application Note

Using run-length encoding on Nano GigE cameras

Applies to Genie Nano-1GigE and Nano-5GigE Sony sensor monochrome models

Overview

Run-length encoding (RLE) algorithms compress data by storing a pixel value and number of consecutive pixels with that value in a run. New features enable Nano-1GigE and 5GigE Sony monochrome models to use a custom RLE algorithm that compresses high and low pixel values while leaving middle values untouched. These new features are available through firmware.

Prerequisites

- Genie Nano-1GigE or Nano-5GigE Sony sensor monochrome model
- Firmware[1]:
  - Nano-1GigE (coming soon)
  - Nano-5GigE
- Sapera LT 8.60 or later recommended
- Visual Studio® 2013 (or later)

Sapera LT 8.60
Sapera LT is the image acquisition and control software development kit (SDK) for Teledyne DALSA cameras, which includes CamExpert, a utility providing user-friendly access to camera features for configuration and setup. Sapera LT SDK is available for download from the Teledyne DALSA website:

http://teledynedalsa.com/imaging/support/downloads/sdks/

You will need administrator rights to install Sapera LT.

[1] Make sure your browser supports the ftp protocol; it might be disabled by default.

RLE Implementation on Nano GigE cameras

Run-length encoding (RLE) algorithms are most efficient with images containing many consecutive pixels with the same value (runs), like with icons. The RLE implementation on Nano GigE cameras creates such runs by assigning a fixed high or low value to pixels above and below specified thresholds. RLE compression is then applied to these fixed values only. In short, the algorithm:

- Encodes pixel values above a user-specified bright threshold to a 0xFF value.
- Encodes pixel values below a user-specified dark threshold to a 0x00 value.
- Executes the RLE algorithm on runs of 0xFF and 0x00 values; all values in between remain untouched.

The benefits of this implementation include reduced bandwidth for transferring data, simplicity, and robustness.

Encoding

A run is stored in three bytes:

- 1 byte to store the value, either 0 (0x00) or 255 (0xFF).
- 2 bytes to store the number of pixels in the run (little-endian).

Below is an example with dark threshold = 0x0F and bright threshold = 0xF0.

Image:

Encoded:

The first 4 pixels of the image are encoded with a 0 value, the middle three pixels remain untouched, the last three pixels are encoded with a 255 value.

The maximum length of a run is 65,535 pixels. If longer, then another 3 bytes will be allocated for the next pixels of the run, and so on.

Data transfer and decoding

The original image size is transferred in the GigE Vision header, while the data is transferred in the GigE Vision Image Payload using a *Mono8* pixel format. You can see the amount of image data transferred in the Acquisition and Transfer Control category.

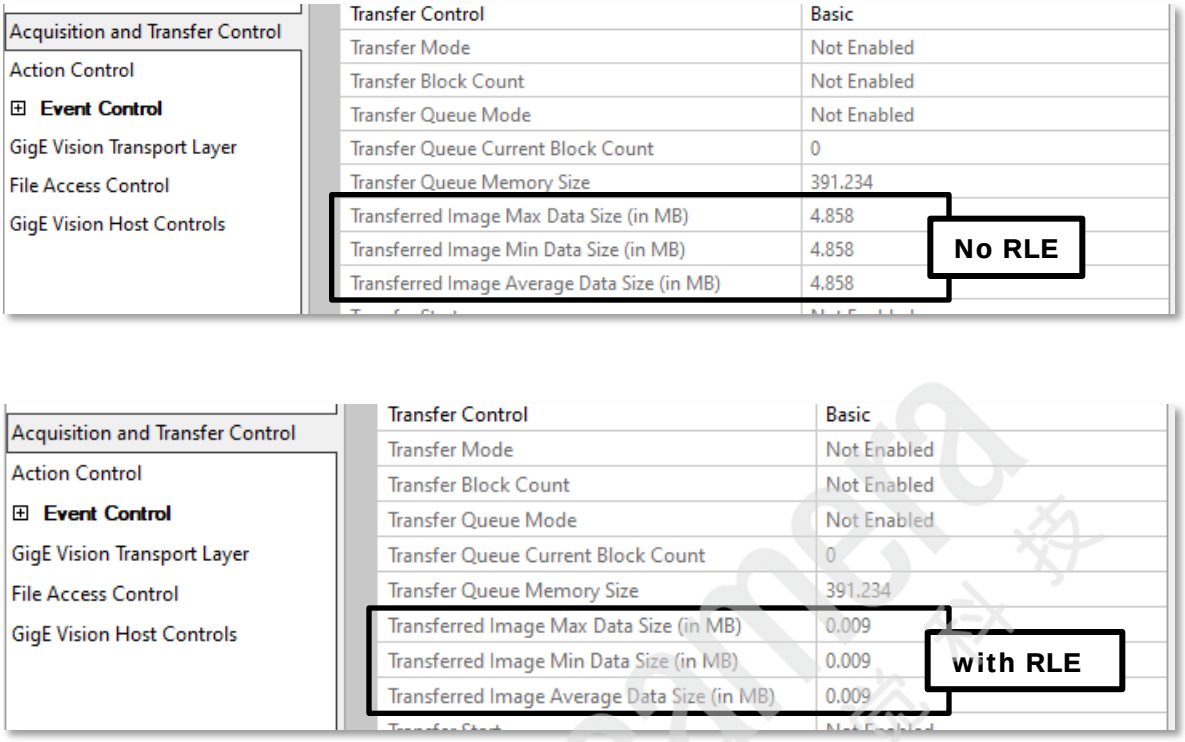


Figure 1. Transferred image data without RLE (above), and with RLE (below).

Decoding is left to the host computer program. The program does not know the size of the compressed image when it receives data, only the original size. Consequently, it needs to decode until the uncompressed data reaches the original size. Sapera buffer parameters CORBUFFER_PRM_IMAGE_WIDTH and CORBUFFER_PRM_IMAGE_HEIGHT can be used to read this information.

Sapera LT example program

The GrabCPP.cpp program file demonstrates how to grab images from a camera into a buffer in the host computer's memory using Sapera++ SapAcquisition and SapBuffer objects, and how to link them using a SapTransfer object. It also shows how the host computer can display the uncompressed images as it receives them using a SapView object. Finally, it contains the RLE decompression code that was implemented for the example.

Before using the program, you must install the firmware and set the RLE features in CamExpert.

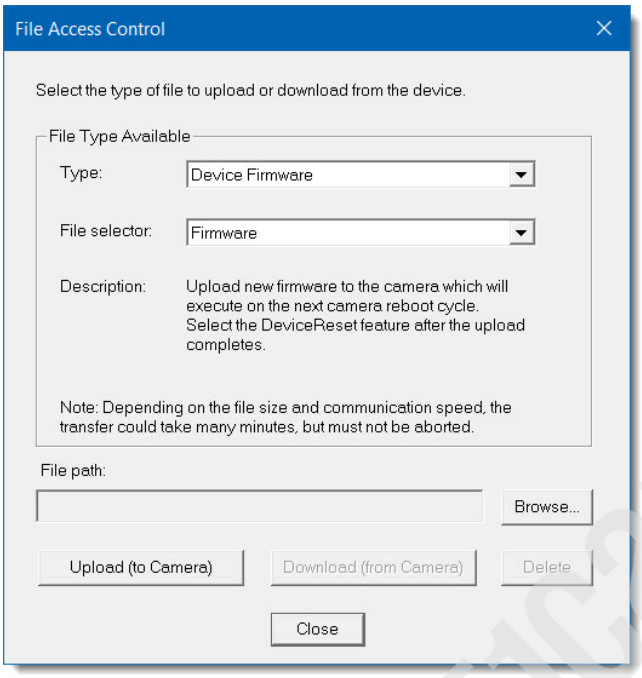
Installing the RLE firmware

Use CamExpert to install the firmware.

To install the new firmware

1. Download the firmware from the specified link.

2. In CamExpert, select your Nano GigE device.
3. In the **File Access Control** category, click **Setting** to open the *File Access Control* dialog.



4. Make sure the **Type** indicates *Device Firmware*, and **File selector** indicates *Firmware*.
5. Click **Browse** to select the firmware file.
6. Click **Upload (to Camera)**.
7. Reset the device.

Setting the RLE features

The firmware adds a new feature category, **rleProcessing**, with the following features:

- **RLE Encoder Mode**. Values: *Active* or *Off*.
- **RLE Encoder Dark Threshold**: Pixel values below the dark threshold will be encoded to 0x00.
- **RLE Encoder Bright Threshold**: Pixel values above the bright threshold will be encoded to 0xFF.

Note that pixels with values equal to either threshold are not modified.

To set RLE features in CamExpert

1. Open the **Advanced Processing** category and select **rleProcessing**.

2. Set **RLE Mode** to *Active*
3. Provide values for the **RLE Bright Threshold** and **RLE Dark Threshold** features.
4. In the **Image Format Controls** category, make sure the **Pixel Format** is *Mono8*.

Running the example program

The GrabCPP.cpp file attached with this application note is a modified version of a program provided with Sapera LT. You will need to replace the existing file and rebuild the project as indicated below.

To compile and run the program

1. Open the C:\Program Files\Teledyne DALSA\Sapera\Examples\Classes\GrabConsole folder.
2. Rename the existing GrabCPP.cpp file to keep it as backup.
3. Paste the GrabCPP.cpp file attached with this application note.
4. In the Classes folder above, open the SapExamples solution for your Visual Studio version.
5. If you are running on a 64-bit platform, open the **Configuration Manager** and, from the **Active solution platform** list, select *x64*.
6. Rebuild the GrabCPP project and run. The console program will open (Figure 2).
7. Select a Nano GigE camera among the connected devices.
8. Select a camera configuration file (.ccf):
   - If the camera is already correctly configured (e.g. through CamExpert), select *No Config File*.
   - Otherwise, select the .ccf file containing the configuration for RLE mode[2].

[2] For convenience, create a .ccf file with the appropriate RLE settings using CamExpert, instead of configuring the camera manually each time. Save it in the default folder.

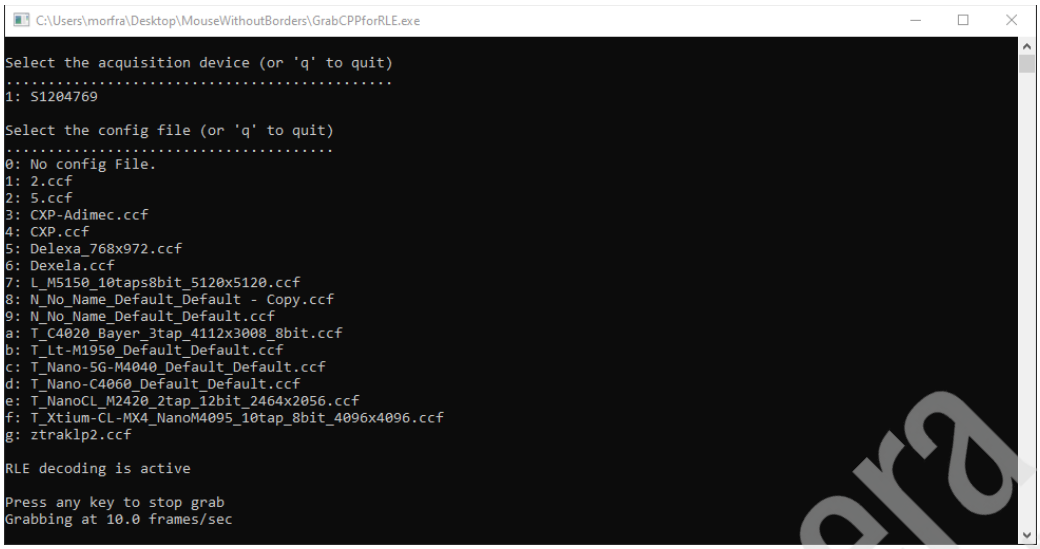


Figure 2. The GrabCPP console program.

Figure 3. (Left) Original image. (Right) RLE Decoded image. Encoding parameters were Dark Threshold = 50, Bright Threshold = 150. The dark runs are prevalent, with a couple of bright runs on the edge of the fan.