

Raytrix GStreamer plugin

29.06.2017

Overview

The Raytrix GStreamer plugin contains the following elements:

element name	input	output	description
rxautoadapt	video/x-bayer	video/x-bayer	Analyzes image content to estimate Gain/Exposure and White balance parameter
rxcudaupload	video/x-bayer	CRxDeviceImage2D (x-bayer)	Uploads the image content to CUDA capable memory, provides GST_V4L2_IO_USERPTR memory for videosources
rxcudadebayer	CRxDeviceImage2D (x-bayer)	CRxDeviceImage2D (rgba)	Performs CFA (color filter array / bayer pattern) demosaicing
rxcudasink	CRxDeviceImage2D (x-bayer) CRxDeviceImage2D (rgba)		Display CUDA device images in a OpenGL window.
rxcudajpegsink	CRxDeviceImage2D (rgba)		Uses Tx1 Jpeg Hardware encoder to encode and write jpeg images to a specified location (currently hardcoded to 3840x2160 pixel images)
rxcudamix	CRxDeviceImage2D (x-bayer)	CRxDeviceImage2D (x-bayer)	Mixes several input streams into a single videowall
rxcudascale	CRxDeviceImage2D (rgba)	CRxDeviceImage2D (rgba)	Scales the input image to a specific size (currently only with hardcoded output size)
rxcudadownload	CRxDeviceImage2D (rgba)	video/-raw	Downloads the image from CUDA memory to host memory

Element details

rxautoadapt

General

This element will analyze the image content to estimate gain/exposure values and white balance parameters.

The element analyzes every nth frame in a specified region of interest (ROI). The update interval is set with the parameter `update` (e.g. `update=5` will analyze every 5th frame). The ROI is given in a normalized [0,1] range.

e.g. your image has a size of 3840x2160 pixel, the ROI definition `roix1=0.2 roix2=0.8 roiy1=0.1 roiy2=0.6` will define a rectangle starting at pixel (768, 216) and ending at pixel (3072, 1296) the ROIs width/height will be (2304, 1080).

Because analyzing every pixel in the image would be too expensive a sub sampling scheme is used. The amount of samples taken in each direction (horizontal and vertical) can be set with the parameter `samples`. Therefore a total of samples^2 pixel are used for Auto Gain/Exp/WB. The boolean parameters `autoexp`, `autogain` and `autowb` control which correction should be applied.

AutoGain / AutoExposure

The average brightness within the ROI is calculated. The gain and exposure values of the camera are now modified to match the given `target` value. To avoid frequent adaptions of exposure/gain, there is also a `tolerance`. Adaption takes place if current average brightness is smaller than `target - tolerance` or larger than `target + tolerance`.

There is also a mechanism to avoid over-exposing the sensor. You can specify at what level a pixel is considered overexposed with the property `oethreshold` and weight those overexposed pixels with a `oepenalty`. Now we have the current average brightness and the target value, we can adapt the gain/exp values. To avoid overshooting

you can specify the rate on how fast we want to reach the target brightness with the parameter `expadaption`. A value of 0.0 will cause no adaption at all, a value of 1.0 will try to reach target brightness as fast as possible. Due to non-linear behavior of the sensor we cannot determine the exact values for exp/gain in a single step. Overshooting is very likely to occur when adaption is done too fast.

 Overexposed Pixel handling is not stable at the moment.

AutoWhiteBalance

Auto white balance is analyzing the image and evaluates if a pixel is considered gray or not. The strictness of this test can be controlled with [wbgraythreshold](#). The higher the value, the more pixel are considered gray. We need a sufficient amount of gray pixel to estimate the color temperature. Once the color temperature is determined from the gray pixel, the Red/Blue balance is shifted to compensate the color-temperature. To avoid overshooting, the parameter [wbadaptation](#) can be used to scale the adaption rate.

rxcudajpegSink

This element uses the Tx1 hardware encoder to compress jpeg images. There are two parameters to control the behavior of this element. The jpeg compression quality can be set with the parameter [quality](#) that ranges from 0 (lowest quality) to 100 (best quality). The second parameter [location](#) decides the path and enumeration of the files to write. The parameter uses standard printf() format conversion to index the files. E.g. location=/path/to/write/image_%05d.jpg will write files to /path/to/write and name the first file: image_00000.jpg, the second image_00001.jpg ..