



Class V4L2VideoCaptureOp

Table of contents

Nested Relationships

Inheritance Relationships

Class Documentation

- Defined in [File v4l2_video_capture.hpp](#)

Nested Relationships

Nested Types

- [Struct V4L2VideoCaptureOp::Buffer](#)

Inheritance Relationships

Base Type

- `public holoscan::Operator` ([Class Operator](#))

Class Documentation

class V4L2VideoCaptureOp : public holoscan::Operator

[Operator](#) class to get the video stream from V4L2.

<https://www.kernel.org/doc/html/v4.9/media/uapi/v4l/v4l2.html>

Inputs a video stream from a V4L2 node, including USB cameras and HDMI IN.

- Input stream is on host. If no pixel format is specified in the yaml configuration file, the pixel format will be automatically selected. However, only `AB24` and `YUYV` are then supported. If a pixel format is specified in the yaml file, then this format will be used. However, note that the operator then expects that this format can be encoded as RGBA32. If not, the behavior is undefined.
- Output stream is on host. Always RGBA32 at this time.

Use `holoscan::ops::FormatConverterOp` to move data from the host to a GPU device.

==Named Outputs==

- **signal** : `nvidia::gxf::VideoBuffer`

- A message containing a video buffer on the host with format `GXF_VIDEO_FORMAT_RGBA`.

==Parameters==

- **allocator**: Memory allocator to use for the output.
- **device**: The device to target (e.g. `"/dev/video0"` for device 0). Default value is `"/dev/video0"`.
- **width**: Width of the video stream. Optional (default: `0`).
- **height**: Height of the video stream. Optional (default: `0`).
- **num_buffers**: Number of V4L2 buffers to use. Optional (default: `4`).
- **pixel_format**: Video stream pixel format (little endian four character code (fourcc)). Default value is `"auto"`.
- **exposure_time**: Exposure time of the camera sensor in multiples of 100 μ s (e.g. setting `exposure_time` to 100 is 10 ms). Optional (default: auto exposure, or camera sensor default). Use `v4l2-ctl -d /dev/<your_device> -L` for a range of values supported by your device.
 - When not set by the user, `V4L2_CID_EXPOSURE_AUTO` is set to `V4L2_EXPOSURE_AUTO`, or to `V4L2_EXPOSURE_APERTURE_PRIORITY` if the former is not supported.
 - When set by the user, `V4L2_CID_EXPOSURE_AUTO` is set to `V4L2_EXPOSURE_SHUTTER_PRIORITY`, or to `V4L2_EXPOSURE_MANUAL` if the former is not supported. The provided value is then used to set `V4L2_CID_EXPOSURE_ABSOLUTE`.
- **gain**: Gain of the camera sensor. Optional (default: auto gain, or camera sensor default). Use `v4l2-ctl -d /dev/<your_device> -L` for a range of values supported by your device.
 - When not set by the user, `V4L2_CID_AUTOGAIN` is set to false (if supported).

- When set by the user, V4L2_CID_AUTOGAIN is set to true (if supported). The provided value is then used to set V4L2_CID_GAIN.

Public Functions

HOLOSCAN_OPERATOR_FORWARD_ARGS (V4L2VideoCaptureOp)

V4L2VideoCaptureOp()=default

virtual void setup(OperatorSpec &spec) override

Define the operator specification.

Parameters

spec – The reference to the operator specification.

virtual void start() override

Implement the startup logic of the operator.

This method is called multiple times over the lifecycle of the operator according to the order defined in the lifecycle, and used for heavy initialization tasks such as allocating memory resources.

virtual void initialize() override

Initialize the operator.

This function is called when the fragment is initialized by Executor::initialize_fragment().

virtual void compute(InputContext &op_input, OutputContext &op_output, ExecutionContext &context) override

Implement the compute method.

This method is called by the runtime multiple times. The runtime calls this method until the operator is stopped.

Parameters

- **op_input** – The input context of the operator.

- **op_output** – The output context of the operator.
- **context** – The execution context of the operator.

virtual void stop() override

Implement the shutdown logic of the operator.

This method is called multiple times over the lifecycle of the operator according to the order defined in the lifecycle, and used for heavy deinitialization tasks such as deallocation of all resources previously assigned in start.

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/05/2024