

20/20 Robot Vision

How to setup cameras in ROS 1 & ROS 2 using camera_aravis

Peter Mortimer

Institute for Autonomous Systems Technology (TAS)
Department of Aerospace Engineering
Bundeswehr University Munich

ROSCon 2022

October 21, 2022



What are Industrial Cameras?

The term industrial camera refers to cameras typically used in machine vision applications. They are also commonly used in robotics research and give the user a lot of control over the image acquisition process.



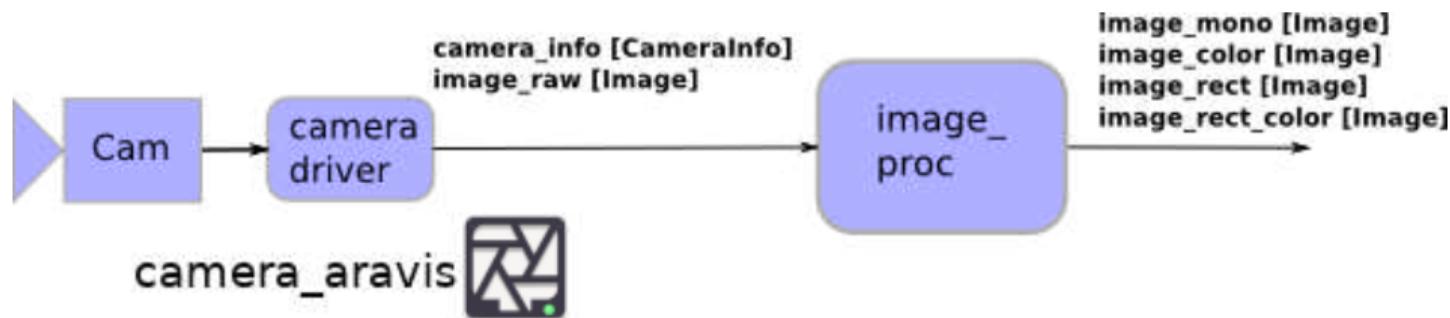
The industrial cameras often only consist of the image sensor.



JAI FS 3200D (■) is a prism camera holding two image sensors.



What does a camera driver do?



- read out the raw camera image (`.../image_raw`)
- adjust settings in the Image Signal Processing (ISP) pipeline



Overview of Camera Driver Packages in ROS 1

Many manufacturers have their custom ROS driver:

- [pylon-ros-camera](#) for cameras from Basler
- [prosilica_driver](#) for cameras from Allied Vision Prosilica
- [ximea_camera](#) for cameras from XIMEA
- [pointgrey_camera_driver](#) for FLIR Point Grey Cameras

...

⇒ Many of these packages are irregularly maintained and don't support industrial cameras from other manufacturers.



camera_aravis

This ROS package is based on the open-source library [aravis](#) that implements the protocols for USB3 and gigabit ethernet cameras. The library works with any GenICam camera. [camera_aravis](#) includes the following features:

- access any ethernet camera over the network by its GUID
- any custom camera feature name can be passed as ROS parameter
- **new:** support for multi-stream cameras
- **new:** support for PTP



Launch File Design

```
<launch>
  <node pkg="nodelet" type="nodelet" name="nodelet_manager" args="manager"/>
  ...
  <node pkg="nodelet" type="nodelet" name="driver" args="standalone camera_aravis/CameraAravisNodelet">
    <param name="guid" value="Basler-acA2440-20gc-23501327"/>
    <param name="frame_id" value="sensor/camera/roof/left"/>
    <param name="camera_info_url" value="file://$(find io_camera)/calibrations/basler_roof_left.yaml"/>
    ...
  </node>
</launch>
```

The camera_aravis driver is implemented as a nodelet. The guid of a given camera can be determined with the [arv-tool](#) command line tool.

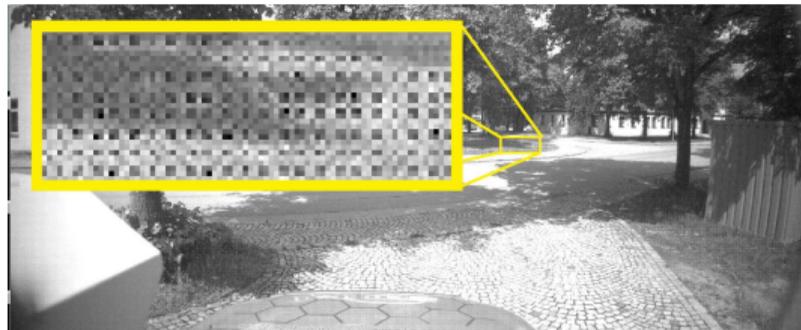


Pixel Formats

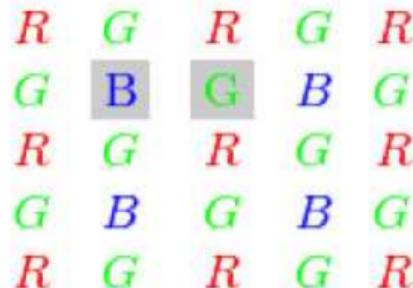
Most cameras the output image to take different formats, but in most cases your camera will use **Mono8** for grayscale images (ROS: `../image_mono`) and **BayerXY** (ROS: `../image_raw`) for mosaic color images. `camera_aravis` uses the OpenCV naming convention for the Bayer pattern.

Color cameras do not record a full color image for each pixel, but reconstruct the color image from an incomplete color sample. The samples typically come from a color filter array with a so-called Bayer pattern of alternating **red (R)**, **green (G)** and **blue (B)** color filters.

The process of reconstructing the color image is called **Debayering** or **Demosaicing**. This is accomplished by the `image_proc/debayer` nodelet in the ROS 1 `image_proc` package.



Raw color images have this grid pattern from the different color measurements after the Color Filter Array (CFA).



In the example above, the bayer pattern of the image sensor has the pixel format **BayerBG8** (image taken from the [OpenCV documentation](#)).



Special Pixel Formats

Debayered Color Pixel Format RGB8

Some cameras (e.g. [FS-3200D-10GE](#)) provide in-camera debayering. You can then skip the debayering step otherwise necessary on the host computer.

Using RGB8 or similar pixel patterns is seen as a tradeoff between an increase in image quality (optimized debayering algorithm for the camera), but a reduction in frame rate and a $3\times$ increase in bandwidth.

10-bit and 12-bit images

Pixel formats that take up more than 1 byte per pixel intensity value (Mono12, BayerBG10) are supported by camera_aravis, but other parts of the image processing pipeline only work with 8-bit images (e.g. `rqt_image_view`).



Exposure Modes

For outdoor environments with dynamic lighting conditions it is best to set an exposure time range for the auto exposure of the camera to operate in. You have to set [ExposureAuto](#) on (supported by most cameras). The camera feature name for the min/max exposure range does not have a genicam-wide standard yet.

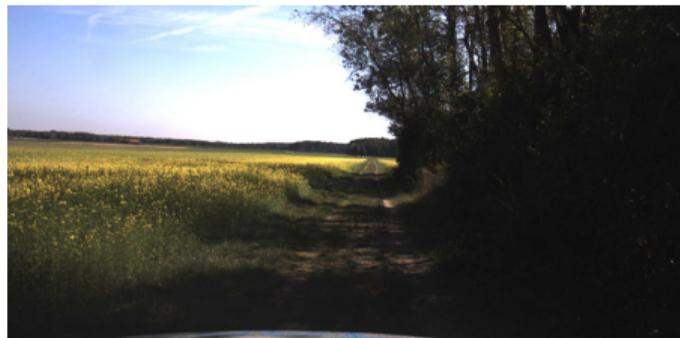
Basler and FLIR cameras typically use the feature names [AutoExposureTimeLowerLimit](#) and [AutoExposureTimeUpperLimit](#).

JAI cameras use the feature names [ExposureAutoControlMin](#) and [ExposureAutoControlMax](#).

The exposure times are generally specified in μs



An image well light in the near distance, but containing overexposed areas in the far distance (see left).



Under these challenging lighting conditions, the camera produces an image underexposed along the path (see right).



Setting an Area of Exposure

Many cameras also support specifying an area of exposure, where the internal auto exposure calculation should be applied on. For JAI cameras this is specified as a percentage value in [ALCReference](#). For Basler cameras the feature is called [AutoTargetValue](#) and is specified as the target average gray value within the area of exposure.

You typically choose the area of interest for your application as the area that should be well exposed.

Basler cameras allow you to specify an exact pixel position, while JAI cameras allow you to enable certain areas of the uncropped image plane ([MidLowLeft](#), [LowLeft](#),...).



The same scene with different areas of the image used for the internal auto exposure control.

High Left	High Mid-left	High Mid-right	High Right
Mid-High Left	Mid-High Mid-left	Mid-High Mid-right	Mid-High Right
Mid-Low Left	Mid-Low Mid-left	Mid-Low Mid-right	Mid-Low Right
Low Left	Low Mid-left	Low Mid-right	Low Right

Camera regions that can be enabled and disabled for the automatic level control (from the [JAI Fusion 3200D Documentation](#))



Gain

Gain adjusts how sensitive the image sensor converts incoming electrons to brightness in the image. Raising the gain value is another way to bright up a low-light environment at the risk of adding random noise. In most cases during daylight it is recommended to maximize the exposure time as much as possible to illuminate an image and to keep the gain value as low as possible.



The same scene for different Gain and a fixed exposure time of $2000\mu\text{s}$.



Tuning Exposure Time and Gain

Most industrial cameras can internally balance their exposure time and gain. For some cameras you can specify a profile that the camera should follow.

Set the standardized genicam camera features [GainAuto](#) and [ExposureAuto](#) on. For Basler cameras there is the [AutoFunctionProfile](#) feature, where you can set [MinimizeGain](#).

We do not recommend using the [MinimizeExposure](#) profile unless your application does require it (e.g. a not well illuminated environment). This profile tends to produce a slight noise pattern across the image (see the top figure).



Camera with [MinimizeExposure](#) set.



Camera with [MinimizeGain](#) set.

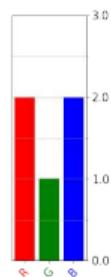
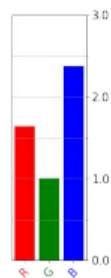


White Balance

Depending on the light source in the scene and its emitting color temperature, the final camera image can have a color tint. The weight of three color channels **red (R)**, **green (G)** and **blue (B)** are adjusted during the white balancing process.

The **BalanceWhiteAuto** feature can automatically adjust the color temperature in the image. If you are operating in a similar environment, then a single **BalanceWhiteReset** at the beginning of the new representative environment is sufficient.

JAI cameras only allow the adjustment of the **DigitalRed** and **DigitalBlue**, assuming a constant 1.0 weight for the green channel.



Same scene shot with different white balance ratios.



camera_aravis in ROS 2

In development with our project partners at the Fraunhofer IOSB.
Some of the roadblocks we face with the current release of ROS 2:

- the driver and other elements of image_proc relied on using the [SubscriberStatusCallback](#) to know whether the node needs to be executing (lazy publishers/subscribers)
- heavy reliance on the ROS parameter server to store all camera feature names



Summary

- camera_aravis gives you control of parts of the ISP.
- If you are using a genicam camera without good ROS driver support, consider using camera_aravis.
- Set min/max exposure time limits.
- Rather increase the exposure time than the gain.
- Reset the white balance values before operating in a new outdoor environment.



Thank you for
your attention!



DALL-E 2 hallucinating to the prompt: *Japanese album cover of an anthropomorphic turtle walking hand-in-hand with a robot in a nice scenery.*

Find us on GitHub: https://github.com/FraunhoferIOSB/camera_aravis



Learning more about the ISP

If you want to learn more about the Image Signal Processing pipeline, then the slides of Prof. Michael S. Brown from his ICCV '19 workshop are a good place to start.

Title: Understanding color & the in-camera image processing pipeline for computer vision

URL: https://www.eecs.yorku.ca/~mbrown/ICCV19_Tutorial_MSBBrown.pdf



Example XML with ISP Settings

```
<launch>
  <node pkg="nodelet" type="nodelet" name="nodelet_manager" args="manager"/>
  ...
  <node pkg="nodelet" type="nodelet" name="driver" args="standalone camera_aravis/CameraAravisNodelet">
    <param name="guid" value="Basler-acA2440-20gc-23501327"/>
    <param name="frame_id" value="sensor/camera/roof/left"/>
    <param name="camera_info_url" value="file://$(find io_camera)/calibrations/basler_roof_left.yaml"/>
    ...
    <param name="pixel_format" value="BayerRG8"/>
    ...
    <param name="GainAuto" value="Continuous"/>
    <param name="ExposureAuto" value="Continuous"/>
    <param name="AutoExposureTimeLowerLimit" value="100"/>
    <param name="AutoExposureTimeUpperLimit" value="30000"/>
    <param name="BalanceWhiteAuto" value="Once"/>
  </node>
  ...
</launch>
```

