FlightView Manual
Last updated: 2023-10-04 EHL

What is FlightView?

FlightView, also known as LiveView, is a program designed at JPL to provide real-time analysis of data from imaging spectrometers arriving at a computer over CameraLink. The computer uses Nvidia's CUDA toolkit to compute the standard deviation of the image on a per-pixel basis, which is useful both for eliminating noise sources and as a high-sensitivity visualization of the data itself. FlightView is an open-source technology available via the NASA-JPL Github site:
https://github.com/nasa-jpl/LiveViewLegacy

Installing FlightView

Part 1: Prerequisites:

OS:

FlightView has been tested on recent versions of Ubuntu Linux and Linux Mint. Most development has taken place using Linux Mint 20.2 with kernel 5.4.0-74-generic.

GPU:

The computer must be using the proprietary Nvidia graphics drivers as well as a matching cuda driver. Additionally, a CUDA development kit must be installed which includes the nvcc nvidia cuda compiler and associated libraries. All these can be installed by using a single download.

Nvidia CUDA linux how-to:
https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html

Be sure to remove any prior version(s) of CUDA and/or the Nvidia drivers prior to installing.

Install your kernel headers and build-essential files. You may need to install the kernel headers again if you update your OS:

```
sudo apt-get install build-essential
sudo apt-get install libc-dev libc6-dev gcc g++
sudo apt-get install linux-headers-$(uname -r)
```

It is strongly recommended to install the ssh server so that if the graphics do not come up, an ssh connection can be made for diagnostics:

```
sudo apt-get install openssh-server
```

The Nvidia CUDA download will include drivers for the nvidia card, cuda support files, and the nvcc.

It is recommended to use the ".run" shell script for the installation, and to not use the RPM or DEB options.

To run the ".run" installer file, download it to a known location that is not in /tmp. Next, log out from the GUI desktop. Press CONTROL-ALT-F2 to get to a text console, or ssh in from a local machine. Issue the following command to stop the desktop process (this is needed in order to run the installer):

```
sudo systemctl stop lightdm # mint
sudo systemctl stop gdm3 # ubuntu
```

Now add execute permission to the runfile, like this (modify for your exact filename):

```
chmod 755 cuda_11.6.0_510.39.01_linux.run
```

Now run the file, as root:

```
sudo ./cuda_11.6.0_510.39.01_linux.run
```

If you run into errors, there are two log files to examine:

```
/var/log/cuda-installer.log
```

and

```
/var/log/nvidia-installer.log
```

Be sure to keep the installer logs, and follow the directions carefully to include the library location with your system. On this particular linux workstation, the file `/etc/ld.so.conf.d/cuda-11-6.conf` was created with the following contents:

```
/usr/local/cuda-11.6/targets/x86_64-linux/lib
```

Do not copy this text directly; adjust it for your system. Be sure to run `sudo ldconfig` after updating the file.

Make sure to blacklist the open source nvidia graphics driver prior to rebooting:

```
sudo bash -c "echo blacklist nouveau > /etc/modprobe.d/blacklist-nvidia-nouveau.conf"
sudo update-initramfs -u
sudo update-grub
```

You may want to take the opportunity to blacklist the wifi or other unneeded hardware while you are editing these files.

Camera Link drivers:

Even when not planning to use camera link, the driver header files are still required for building. Make sure to install the latest EDTpdv drivers:

https://edt.com/file-category/pdv/
(download the .run file and follow the directions. The installer must be run as root, and the default location, /opt/EDTpdv is recommended)

Typically, the installer will attempt to build all the EDTpdv utilities. Some of these may fail, but generally the most important ones will build without too much trouble.

Recent EDT drivers need these additional steps:
```
sudo chmod -R o+r /opt/EDTpdv
sudo mv /opt/EDTpdv/version /opt/EDTpdv/version-orig
```

The drivers include a kernel module, which can be manually re-built with the following commands:
```
cd /opt/EDTpdv/module
sudo make
```
(make includes install)
```
sudo update-initramfs -u
sudo update-grub
```

Rebuilding this module after OS updates is generally required as the kernel will update without bringing in the EDTpdv drivers. Always make sure you have the correct linux-headers package installed (see previous steps for the nvidia driver).

To initialize the camera link port, the `initcam` program is used with a camera link config file. By default, a large set of camera link config files are provided in `/opt/EDTpdv/camera_config`. These plaintext files may be used as a starting point for building a custom config file.

Here is an example for a 640x481 14-bit camera:

```
camera_class:                   "Imaging Spectrometer"
camera_model:                   "Camera Link 14-bit"
camera_info:                    "640x481 HSI"
width:                          640
height:                         481
depth:                          14
extdepth:                       14
CL_CFG_NORM:                    02
continuous:1
MODE_CNTL_NORM: 00
```

Here is an example for a 1280x328 16-bit camera:

```
camera_class:                   "Imaging Spectrometer"
camera_model:                   "FPA"
camera_info:                    "1280x480 16-bit HSI"
width:                          1280
height:                         328
depth:                          16
extdepth:                       16
```

```
rbtfile: aiagcl.bit
continuous: 1
CL_CFG_NORM:                  00
htaps: 4
serial_baud: 19200
```

A Camera Link port can be initialized like this:

```
/opt/EDTpdv/initcam -f /opt/EDTpdv/camera_config/spectrometer.cfg
```

It is recommended to build a shell script for this task and then to call it from a desktop launcher, with the terminal opening to show any error messages.

Example desktop file: init_camera.desktop:

```
[Desktop Entry]
Name=initcam NGIS
Exec=/home/eliggett/bin/init_ngis.sh
Comment=640x481
Terminal=true
Icon=cinnamon-panel-launcher
Type=Application
```

Example shell script file: init_ngis.sh:

```
#!/bin/bash
CFGFILE=/opt/EDTpdv/camera_config/NGIS.cfg
echo "Running $0 for config file $CFGFILE ..."
/opt/EDTpdv/initcam -f $CFGFILE
echo "Done! Closing window in 2 seconds."
sleep 2;
```

To verify the port was initialized, run the `/opt/EDTpdv/take` command, which attempts to acquire one frame from the camera link port. The command will typically return one of the following:

- `EDT /dev/pdv0 open failed. pdv_open(pdv0_0): No such device or address`
    - This means that either the kernel module did not load, or the card has a problem or isn't physically installed. Check the output of `lsmod` and `dmesg` for details.
- `pdv0: Invalid image size. Make sure device has been initialized.`
    - (Run `initcam`)
- `1 images 1 timeouts 0 overruns 0 bytes`
    - The port was initialized, but the camera was not running.
- `1 images 0 timeouts 0 overruns`
    - The port was initialized, and the camera returned a frame without timeout. This means the camera link system is ready for use.

The return value is generally "1" upon error and "0" upon success.

Part 2: Building and Installing FlightView

Libraries needed to build FlightView:

It is required to install the following packages before building FlightView:

```
sudo apt-get install build-essential
sudo apt-get install libc-dev libc6-dev gcc g++
sudo apt-get install linux-headers-$(uname -r)
sudo apt-get install qt5-default qt5-qmake qt5-qmake-bin
sudo apt-get install git
sudo apt-get install libboost-all-dev
sudo apt-get install libqt5svg5 libqt5svg5-dev
sudo apt-get install qtcreator
sudo apt-get install libgsl-dev libgsl23 libgslcblas0 # adjust names
as needed.
```

FlightView also requires libgst and specific libgst plugins.

```
sudo apt-get install libgstreamer1.0-0   libgstreamer1.0-dev
sudo apt-get install gstreamer-1.0-plugins-base  gstreamer-1.0-
plugins-good gstreamer-1.0-plugins-good-dev
```

For work with raw 16-bit grayscale RTP sources, you will need to patch the gstreamer RTP plugin using the following procedure:

First prepare the apt system for source installs:
Enable apt-src repositories, typically by editing `/etc/apt/sources.list` and uncommenting the first few apt-src lines.

Update the apt cache:

```
sudo apt-get update
```

Install apt-src:

```
sudo apt-get install apt-src
```

Install the source for gstreamer1.0-plugins-good:

```
mkdir ~/Downloads/gst
cd ~/Downloads/gst
apt-src install gstreamer1.0-plugins-good
```

Patch the library:

```
cd gst-plugins-good1.0-16.3/gst/rtp
cp gstrtpvrawdepay.c  gstrtpvrawdepay.c-orig
```

For this next step, you will need a copy of the patch file which is within the LiveViewLegacy repository. Download the patch file separately or clone the repo ahead of time:

https://github.com/nasa-jpl/LiveViewLegacy/tree/flight/utils/gst

```
cp ~/Documents/FlightView/LiveViewLegacy/utils/gst/gstrtpvrawdepay.c.patch .
patch gstrtpvrawdepay.c gstrtpvrawdepay.c.patch
```

Now build the plugin:

```
cd ~/Downloads/gst
apt-src build gstreamer1.0-plugins-good
```

Install the modified plugin by installing each of the resulting deb files (adjust names to match):

```
sudo dpkg --install gstreamer1.0-plugins-good-doc_1.14.5-
0ubuntu1~18.04.3_all.deb gstreamer1.0-pulseaudio_1.14.5-
0ubuntu1~18.04.3_amd64.deb gstreamer1.0-qt5_1.14.5-
0ubuntu1~18.04.3_amd64.deb gstreamer1.0-gtk3_1.14.5-
0ubuntu1~18.04.3_amd64.deb gstreamer1.0-plugins-good_1.14.5-
0ubuntu1~18.04.3_amd64.deb gstreamer1.0-plugins-good-dbg_1.14.5-
0ubuntu1~18.04.3_amd64.deb libgstreamer-plugins-good1.0-0_1.14.5-
0ubuntu1~18.04.3_amd64.deb libgstreamer-plugins-good1.0-dev_1.14.5-
0ubuntu1~18.04.3_amd64.deb
```

Download, configure, and build FlightView:

To download, use the following steps:

```
mkdir ~/Documents/FlightView
cd ~/Documents/FlightView
git clone https://github.com/nasa-jpl/LiveViewLegacy.git -b
flightview
```

Now update the submodules (used for the Atlans A7 binary GPS library):

```
cd LiveViewLegacy
git submodule init
git submodule update
```

Compiling the back-end (cuda-take):

If your version of nvcc (`nvcc -V`) is prior to version 11.6, edit the `Makefile (cuda_take/Makefile)`, and uncomment and edit an NVCCFLAGS line which matches your card. You can quyery the card type using the command "`nvidia-smi -L`"

Visit this URL and find the correct specification for your card:
https://arnon.dk/matching-sm-architectures-arch-and-gencode-for-various-nvidia-cards/

Look for the correct “compute” and “sm” codes. If the cuda version is greater than 11.6, these codes may be removed and “arch=native” may be substituted (this is the default NVCCFLAGS line).

```
NVCCFLAGS  = -gencode arch=compute_86,code=sm_86 -lineinfo -Xcompiler
-rdynamic --compiler-options '-Wall -Werror -Wno-unused-function'
```

Verify that the version of `nvcc` on the `$PATH` is the correct version (manual nvidia driver installs typically reside in /usr/local, apt installs in /usr):

```
which nvcc
nvcc -V
nvidia-smi
```

If `nvcc` isn't within the current path, be sure to correct this before proceeding.

To compile the nvidia CUDA back-end, issue the following command from within the `cuda_take` directory:

```
cd cuda_take
```

(current directory is `~/Documents/FlightView/LiveViewLegacy/cuda_take`)

```
make -j
```

If you receive errors about "unknown architecture" then the Makefile will need to be adjusted to specify the correct compute and sm codes. If the error persists, it may mean that your card is newer than the driver supports and you will need to find a beta version of the latest Nvidia cuda drivers.

If you receive this error:

```
/bin/sh: 1: nvcc: not found
```

Then you may need to add the nvidia cuda bin directory to your current path like this (verify this location on your system before proceeding):

```
export PATH=$PATH:/usr/local/cuda/bin
```

(On many systems, the specific cuda folder, such as "cuda-11.4" will be symlinked to "cuda" for convenience.)

Next, create a directory for building FlightView:

```
cd ../..
(location is now ~/Documents/FlightView)
mkdir build
cd build
(location is now ~/Documents/FlightView/build)
```

Run qmake to configure the Makefile:
For a release build (for deployment):

```
qmake ../LiveViewLegacy/liveview.pro
```

For a debug build (for testing):

```
qmake CONFIG+=debug ../LiveViewLegacy/liveview.pro
```

Build FlightView:

```
make -j
```

When FlightView builds, it will be located in a folder within the `build` directory called "`lv_release`", which also contains an icon and an example plaintext desktop launcher (which will likely need to be edited slightly).

It is recommended to install the files as follows:

```
liveview: /usr/local/bin/liveview
liveview_icon.png: /usr/share/pixmaps/liveview_icon.png
liveview.desktop: /usr/share/applications/liveview.desktop
```

Part 3: FlightView Configuration and Startup

FlightView has several command-line startup options as well as a plaintext settings file containing interface preferences.

Command-line arguments:

The command-line options may be displayed by launching FlightView with the "-h" or "--help" arguments. Arguments are not case-sensitive, although specified sub-argument values, such as file paths, are case sensitive. Any sub-arguments are expected to be separated from the primary arguments by a single space (ie, no equals sign may be used).

Argument name	Description	Example
`--datastoragelocation`	Specifies the location for saved raw image data and log files	`--datastoragelocation /mnt/flightdisk/data`
`--flight`	Flight mode. In this mode, the operator is not prompted for file names of recordings, and a GPS connection is required.	`--flight`
`--no-gps`	Disables GPS functionality	`--no-gps`
`--gpsIP`	Specifies the IP address of the Atlans A7 TCP/IP Binary V5 GPS data server.	`--gpsIP 192.168.2.6`
`--debug`	Enable experimental and/or debug options. Not recommended for deployment.	`--debug`
`--no-stddev`	Disable the standard deviation calculation, which uses the GPU	`--no-stddev`
`--rtpcam`	Enable the RTP camera module. Disables xio and cameralink	`--rtpcam`
`--rtpwidth`	Specify the width of the RTP image	`--rtpwidth 640`
`--rtpheight`	Specify the height of the RTP image	`--rtpheight 480`

`--rtpaddress`	Specify the listening address of the RTP stream	`--rtpaddress ::1`
		`--rtpaddress fe02::1`
		`--rtpaddress 224.0.0.0`

`--rtpinterface`	Specify the interface for the RTP stream. Appears to be ignored by the library	`--rtpinterface enps0s1`
`--rtprgb`	The RTP camera module will extract the first 16-bits from an assumed 24-bit RGB format frame. This option is enabled by default.	`--rtprgb`
`--rtpgray`	The RTP camera module will copy all the bits from the RTP stream to the image frame, assuming 16 bits per pixel.	`--rtpgray`
`--rtpgrey`	Same as above	`--rtpgrey`

These arguments can (and should) be included in desktop launcher icons. For example, two launchers can be created, one for flight mode and one for ground testing.

An example for camera link Flight command-line settings:
```
liveview --flight --datastoragelocation /mnt/data/flightdata --gpsIP 10.0.0.6 --no-stddev
```

An example camera link Ground command-line settings (none):
```
liveview --datastoragelocation /tmp
```

Plaintext Settings File:

The plaintext settings file is read from this location:
`~/.config/FlightView/FlightView.conf`

It is recommended to first launch FlightView, press “Preferences”, and then press “Save Settings”. This will create a basic config file which could be edited. Most of the options within the settings file can be edited from the GUI of FlightView, so it is not necessarily required to edit this file. The following is an example configuration file, with comments (do not replicate with the comments, this has not been tested):

```
[Camera]
brightSwap14=false ; causes 0 to be bright and 2^14 to be dark
brightSwap16=false ; same as above but for 16-bit data
nativeScale=true ; Uses the full dynamic range instead of bit masking.
skipFirstRow=true ; skips the first row for standard deviation calculations (skips metadata)
skipLastRow=false ; skips the last row for standard deviation calculations
use2sComp=true ; Converts 2s compliment data into 16-bit unsigned int.

[Flight]
hideFFT=true ; hides the FFT tab
hideHistogramView=false ; hides the standard deviation histogram tab
hideHorizontalCrosshairProfile=false ; hides the horizontal crosshair profile tab
hideHorizontalMeanProfile=false ; hides the horizontal mean profile tab
hidePlayback=true ; hides the playback tab (strongly recommended to hide)
hideStddeviation=false ; hides the standard deviation tab
hideVertCrosshairProfile=false ; hides the vertical crosshair profile tab
hideVertMeanProfile=false ; hides the vertical mean profile tab
hideVerticalOverlay=true ; hides the vertical overlay tab
hideWaterfallTab=true ; hides the monochromatic data waterfall tab (not the same as the
flight waterfall)
```

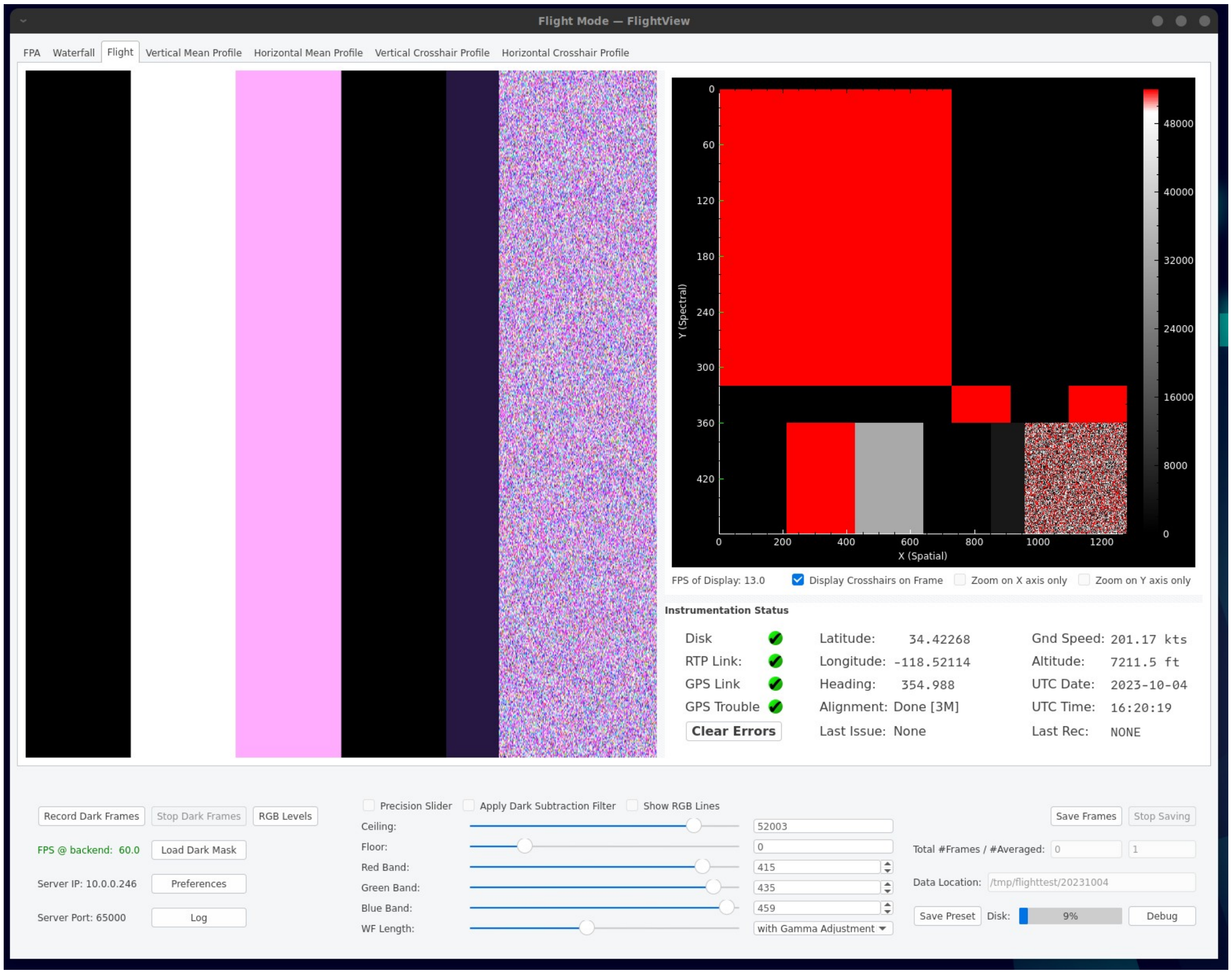
```
percentDiskStop=99 ; sets the absolute maximum disk utilization level. Will not record past
the indicated percent (99% in this case)
percentDiskWarning=85 ; sets the disk warning utilization threshold. 85% in this case.

[Interface]
dsfCeiling=1678 ; dark subtraction ceiling slider value
dsfFloor=-580 ; dark subtraction floor slider value
fftCeiling=1000 ; FFT ceiling value
fftFloor=0 ; FFT floor value
frameColorScheme=0 ; currently selected frame color scheme (“JET”)
frameViewCeiling=6193 ; frame view ceiling value
frameViewFloor=1678 ; frame view floor value
stddevCeiling=799 ; standard deviation ceiling value
stddevFloor=242 ; standard deviation floor value
useDarkTheme=true ; use dark theme for plots
preferredHeight=1200 ; preferred main window height in pixels
preferredWidth=1600 ; preferred main window width in pixels
flightCeiling=37746 ; flight screen ceiling
flightDSFCeiling=10198       ; flight screen ceiling when DSF
flightDSFFloor=-1884         ; flight screen floor when DSF
flightFloor=0                ; flight screen floor
monowfCeiling=10             ; monochromatic waterfall ceiling
monowfDSFCeiling=20000       ; monochromatic waterfall ceiling when DSF
monowfDSFFloor=0             ; monochromatic waterfall floor when DSF
monowfFloor=-2               ; monochromatic waterfall floor
plotPenThickness=4           ; thickness of plot lines
preferredWindowHeight=1024   ; height of main window
preferredWindowWidth=1280    ; width of main window
profileHorizCeiling=65535    ; horizontal plot ceiling
profileHorizDSFCeiling=5     ; horizontal plot ceiling when DSF
profileHorizDSFFloor=-5      ; horizontal plot floor when DSF
profileHorizFloor=-10        ; horizontal plot floor
profileOverlayCeiling=20000     ; overlay ceiling
profileOverlayDSFCeiling=20000  ; overlay ceiling when DSF
profileOverlayDSFFloor=0        ; overlay floor when DSF
profileOverlayFloor=0           ; overlay floor
profileVertCeiling=65535      ; vertical plot ceiling
profileVertDSFCeiling=13423   ; vertical plot ceiling when DSF
profileVertDSFFloor=-2368     ; vertical plot floor when DSF
profileVertFloor=-10          ; vertical plot floor
windowGeometry=@ByteArray()   ; main window geometry (not implemented)
windowState=@ByteArray()      ; main window state (not implemented)

[RGB]
; Flight tab waterfall color selection arrays. Data are often out of order but it seems to
work fine.
bandsRGB\1\bandBlue=300 ; band (horizontal line of a frame) for blue, preset 0
bandsRGB\1\bandGreen=200 ; band (horizontal line of a frame) for green, preset 0
bandsRGB\1\bandName=Best ; Name of preset 0.
bandsRGB\1\bandRed=100 ; band (horizontal line of a frame) for red, preset 0
bandsRGB\1\gainBlue=2.5 ; gain factor for blue
bandsRGB\1\gainGreen=0.86 ; gain factor for green
bandsRGB\1\gainRed=1.42 ; gain factor for red
bandsRGB\1\gamma=1.044 ; gamma factor
bandsRGB\1\gammaEnabled=false ; not implemented, set gamma to zero to disable
```

(continues on next page)

Part 4: FlightView Flight Interface



FPS of Display: 13.0 | [✓ Display Crosshairs on Frame] | Zoom on X axis only | Zoom on Y axis only

Disk	✓	Latitude:	34.42268	Gnd Speed:	201.17 kts
RTP Link:	✓	Longitude:	-118.52114	Altitude:	7211.5 ft
GPS Link	✓	Heading:	354.988	UTC Date:	2023-10-04
GPS Trouble	✓	Alignment:	Done [3M]	UTC Time:	16:20:19
		Last Issue:	None	Last Rec:	NONE

FPS @ backend: 60.0
Server IP: 10.0.0.246
Server Port: 65000

Precision Slider | Apply Dark Subtraction Filter | Show RGB Lines

Ceiling: 52003
Floor: 0
Red Band: 415
Green Band: 435
Blue Band: 459
WF Length: with Gamma Adjustment

Total #Frames / #Averaged: 0 | 1
Data Location: /tmp/flighttest/20231004
Disk: 9%

The flight interface is designed to give the operator a view of the image data, GPS data, disk status, and camera link status. File naming for recordings is automatic.

Interface items:

Tab bar: Select different tabs to see the image data differently. Tabs may be locked out using the settings file.

Left-side image: Waterfall image. The selected red, green, and blue bands are used to determine pixel RGB values (see the RGB sliders). The waterfall may be expanded left-to-right by dragging the divider at the right edge.

Frame view: This area (red and black in this example) shows the raw frame being recorded, optionally with the data being first subtracted from a dark reference. The color scheme (a LUT between brightness values and colors) may be selected in the Preferences window. The scroll wheel can be used to zoom in and out of the image. Panning (once zoomed in) can be done by dragging the image around. Zoom may be restricted to X or Y only with the checkboxes. Double-click to set a "crosshair" from which the crosshair profiles and monochromatic waterfall (hidden) may be drawn from. The crosshair display can be disabled with the checkbox "Display Crosshairs on Frame".

Flight data information area: The space below the line plot contains several status indicators as well as basic GPS data.

Clear Errors: This button clears stored errors. Press this button when an error has been acknowledged by the operator and (potentially) resolved. (See the "Log" for more error information.) Program operation is not altered by this button; it merely clears the error indicators. If an error is persisting, the button may appear not to work, so be sure to check the log for details. When the button is pressed, any GPS-related errors or warnings will be printed out to the Log window.

GPS Link: This "LED" has four states:

- Red X: Error Status. The error may be any of the following:
    - GPS dropped connection
    - Bad decode (such as a bad checksum). One bad decode at initial start is common.
    - Lack of GPS Time telegram (expected once per second)
- Blue circle: Initial status: The GPS is not connected when blue, and has not been rejected or dropped. This is also the status when the --no-gps argument is passed in to flightview
- Green circle: This indicates the GPS connection is healthy, time messages are being received, and the messages are decoding correctly.
- Yellow square with question mark: Indicates a warning status

GPS Trouble: This "LED" has three states:

- Red X: Bad configuration or no reception
- Yellow Square with question mark: Warning, system not fully aligned or otherwise ready.
- Green circle with checkmark: System is ready and fully operataional.
- Blue circle: Initial unknown status state.

CameraLink Status: This "LED" has two states:

- Red X: Frame rate has fallen below 2 FPS. Indicates the camera is likely not producing data.
- Green circle: Frame rate exceeds 2 FPS.

Generally, the EDT driver will time out and produce about 1.5 FPS if there is not any data coming in to the camera link ports.

Disk: This "LED" has three states:

- Green: Disk utilization is below the warning threshold as set by the settings file
- Yellow: Disk utilization is at or above the warning threshold, but below the error threshold
- Red: Disk utilization is at or above the error threshold set by the settings file. The data are not recorded once this happens.

See the Log and also the Disk "progress bar" meter

GPS Latitude and Longitude: Decimal degrees of latitude and longitude. "########" is the initial state prior to receiving data.

GPS Altitude: GPS Altitude, referenced to MSL, in feet.
Heading: GPS Heading in degrees. Readings are from 0 to 360 degrees with decimals.

UTC Date: System date from the computer
UTC Time: Time message received once per second from the computer

Last Rec: Indicates the time of the last recording in hours and minutes (UTC). When bold, the system is currently recording. "None" indicates no recording has taken place yet.

Record Dark Frames: Begins recording a reference frame into a memory buffer. The frames are averaged together in a 64-bit floating point reference frame.

Stop Dark Frames: Press this button to end recording dark frames. Typically 100 to 1000 dark frames are recorded for a reference.

FPS @ backend: This is the framerate received into the cuda back end.

Load Dark Mask: Allows the operator to load a pre-recorded dark mask.

Server IP: This is the ip address that the internal flightview server is using. The TCP/IP server operates on the listed Server Port and accepts connections with the following commands:

- Save to file
- Stop saving
- Query FPS

The intended use of the Server is to automate long data collection sequences. Sample client code, both for python and c++, is provided in the nasa-jpl github under "utils"

Precision slider: Limits the range of the floor and ceiling sliders to -2000 to 2000, useful for when the data are dark subtracted.

Apply Dark Subtraction Filter: Causes the displayed data to be first subtracted from the dark reference data.

Ceiling and Floor sliders: The RGB conversion is clamped to the ceiling and floor selected here and scaled accordingly. Use these sliders to "zoom in" on the most useful range of the instrument's data.

Red, Green, and Blue band sliders: These sliders select the band (horizontal column) of frame data to be used for the RGB waterfall display.

WF Length: Adjusts the apparent "length" of the waterfall display. Use this to adjust the aparent aspect ratio of the resulting image and/or to slow or speed up the waterfall display.

RGB Levels: Opens the RGB Levels toolbox, where the waterfall image gain for each of red, green, and blue colors may be adjusted. Also allows overall image gamma to be set.

Preset Drop-down combo box: Use this drop-down to select a preset number. The presets contain red, green, and blue band values and a name. Select "Rename…" to give the preset a meaningful name, for example "Calibration lamp" or "Water Vapor".

Save Preset: Press this button to save the presets to the settings file.

Save Frames: Press this button to start saving frames. In Flight mode, the frames are named automatically as follows:

Raw data: `YYYY-MM-DD_hhmmss-scenedata.raw`
Data header: `YYYY-MM-DD_hhmmss-scenedata.hdr`
Binary GPS data: `YYYY-MM-DD_hhmmss-scenegps.bin`

Additionally, the following two files are created once per FlightView session:
FlightView event log: `YYYY-MM-DD_hhmmss-FlightView.log` (same as Log dialog box)
Primary GPS data: `YYYY-MM-DD_hhmmss-gpsPrimary.bin` (contains a copy of all GPS data received during the session)

Note: GPS data may be "decoded" using the gpsGUI utility and libraries, available here:
https://github.com/nasa-jpl/gpsGUI
This utility can also connect to an Atlans A7 and display the data in real-time. Important: The Atlans A7 only supports *one* TCP/IP client connection at a time, so do not run the gpsGUI at the same time as FlightView).

Save Location: Only in ground (non-flight) mode. This button brings up a standard save dialog box where the user can select a location for saving files.

Stop Saving: Stops saving raw data.

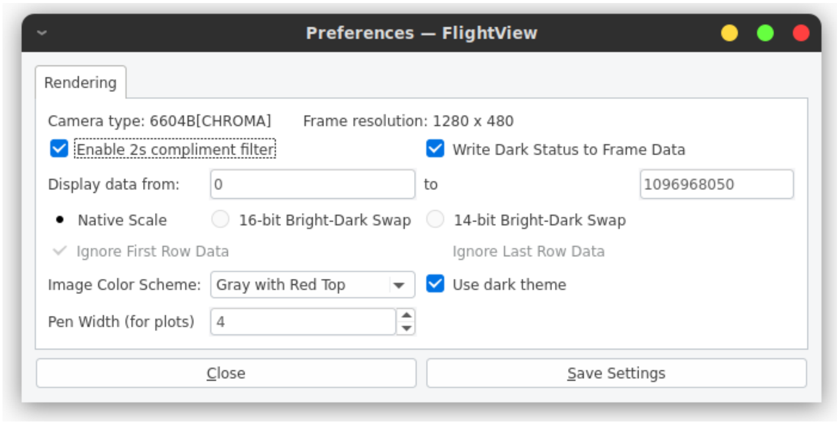
Total #Frames / # Averaged: These two boxes contain the total number of frames recorded for a given raw data collection, and, if used, the number of frames averaged together. In non-flight mode, the operator may enable averaging, for example, averaging every 10 frames together. This is useful in long data collections where the scene is going to be averaged later anyway, such as ground calibration data. The format for average data collections is 32-bit float.

Data Location: The location of the saved data as specified with the `--datastoragelocation` argument. In non-flight mode, the operator may edit this text.

Disk progress bar: Shows the percent disk utilization. Placing the mouse on top of the progress bar shows the amount of space available in GB. Note: System startup disks contain an additional 5% margin which cannot be recorded to. The progress bar takes this into account automatically, thus, 100% full may actually be only 95% full on a startup disk.

Debug: This button should not be pressed and is only visible in debug builds. It is used to aid developers in testing functions without having to build UI elements for every test. Debug builds should not be used in deployments and this button should not be pressed.

Preferences: Opens the preferences dialog box, from which the settings may be changed and saved.


Preferences — FlightView

Rendering

Camera type: 6604B[CHROMA]   Frame resolution: 1280 x 480

[✓] Enable 2s compliment filter   [✓] Write Dark Status to Frame Data

Display data from: 0 to 1096968050

● Native Scale   ○ 16-bit Bright-Dark Swap   ○ 14-bit Bright-Dark Swap

✓ Ignore First Row Data   Ignore Last Row Data

Image Color Scheme: Gray with Red Top   [✓] Use dark theme

Pen Width (for plots) 4

Camera Type: Indicates the type of camera detected. This is depreciated and will not be maintained.

Frame resolution: Frame geometry, not including any metadata rows

Enable 2s compliment filter: Converts 2s compliment data into 16-bit unsigned integer. Applies to all pixels.

Write Dark Status to Frame Data: Writes a value of 2 to pixel 159 when darks are being averaged, otherwise writes a value of 3 to pixel 159.

Display data from: Sets the limits on data which would be applied if the 14-bit bright-dark swap is used.

Native Scale: Specifies that the data shall be left alone.

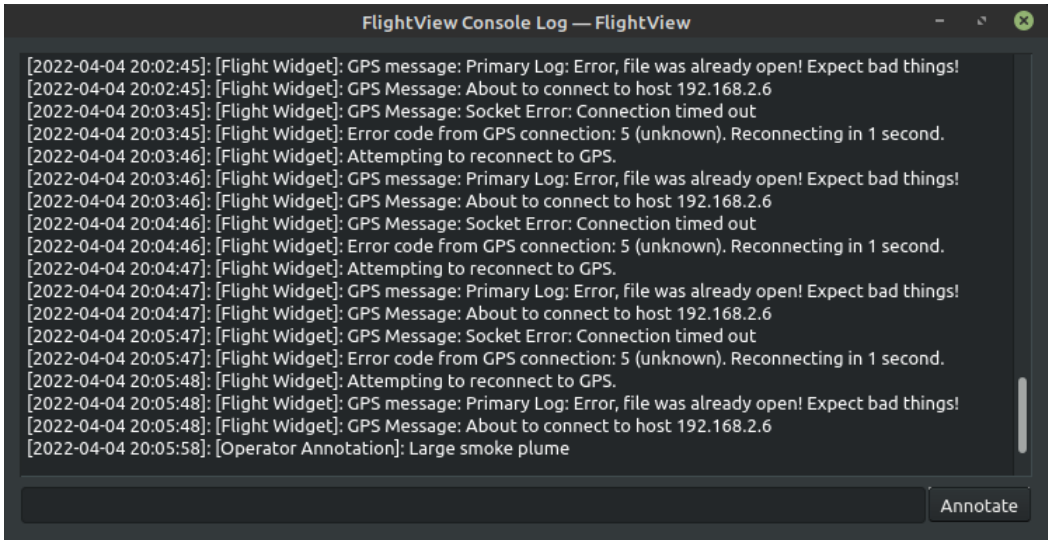
Bright Swap: These buttons cause the data to be mapped to an inverted scale. The 16 bit swap is direct; the 14-bit swap uses the above “display data from” values.

Image Color Scheme: Selects from the built-in color lookup tables.

Use dark theme: Switches the frame and line plots to a dark theme. To make the overall UI of FlightView match the dark theme, select dark theme elements in the Desktop Environment’s theme selector control panel.

Save Settings: Causes the settings, including other UI-related settings such as the RGB presets, to be saved to the settings file. The settings are *not* automatically saved, the user must press this button to save the settings.

Close: Closes this preference dialog box



```
[2022-04-04 20:02:45]: [Flight Widget]: GPS message: Primary Log: Error, file was already open! Expect bad things!
[2022-04-04 20:02:45]: [Flight Widget]: GPS Message: About to connect to host 192.168.2.6
[2022-04-04 20:03:45]: [Flight Widget]: GPS Message: Socket Error: Connection timed out
[2022-04-04 20:03:45]: [Flight Widget]: Error code from GPS connection: 5 (unknown). Reconnecting in 1 second.
[2022-04-04 20:03:46]: [Flight Widget]: Attempting to reconnect to GPS.
[2022-04-04 20:03:46]: [Flight Widget]: GPS message: Primary Log: Error, file was already open! Expect bad things!
[2022-04-04 20:03:46]: [Flight Widget]: GPS Message: About to connect to host 192.168.2.6
[2022-04-04 20:04:46]: [Flight Widget]: GPS Message: Socket Error: Connection timed out
[2022-04-04 20:04:46]: [Flight Widget]: Error code from GPS connection: 5 (unknown). Reconnecting in 1 second.
[2022-04-04 20:04:47]: [Flight Widget]: Attempting to reconnect to GPS.
[2022-04-04 20:04:47]: [Flight Widget]: GPS message: Primary Log: Error, file was already open! Expect bad things!
[2022-04-04 20:04:47]: [Flight Widget]: GPS Message: About to connect to host 192.168.2.6
[2022-04-04 20:05:47]: [Flight Widget]: GPS Message: Socket Error: Connection timed out
[2022-04-04 20:05:47]: [Flight Widget]: Error code from GPS connection: 5 (unknown). Reconnecting in 1 second.
[2022-04-04 20:05:48]: [Flight Widget]: Attempting to reconnect to GPS.
[2022-04-04 20:05:48]: [Flight Widget]: GPS message: Primary Log: Error, file was already open! Expect bad things!
[2022-04-04 20:05:48]: [Flight Widget]: GPS Message: About to connect to host 192.168.2.6
[2022-04-04 20:05:58]: [Operator Annotation]: Large smoke plume
```

Log: Opens the Console Log dialog box, where program events, including errors, are listed. The log is saved to a unique (time-based) name in the location specified in `--datastoragelocation`. The filename has the following format: `YYYY-MM-DD_hhmmss-FlightView.log`. The log dialog box also allows the operator to append notes for later usage, for example, interesting data or malfunctions. The file is ascii text.