Bradley Davis

This document specifies the file format for images sent by the satellite and the various stages of the image through the transfer chain

Image Format

File Format For Images Sent From The Satellite

Revision: 1.0.0



Table of Contents

[1 Example Heading 2](#_Toc497910174)

[1.1 Example Subheading 2](#_Toc497910175)

[1.1.1 Example Subheading 2](#_Toc497910176)

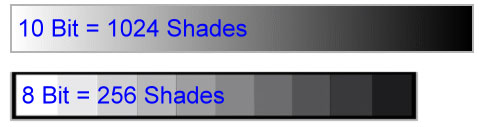
[2 Example Heading 2](#_Toc497910177)

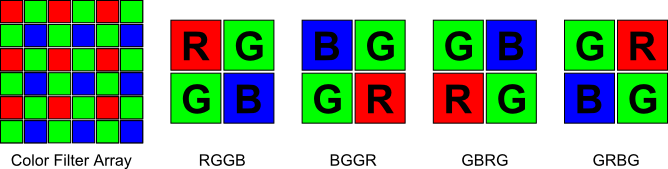
[2.1 Example Subheading 3](#_Toc497910178)

[2.1.1 Example Subheading 3](#_Toc497910179)

# 1 Camera Output Format

The IHU is designed to accept standard parallel data from the camera. The camera outputs either 8 bit or 10 bit pixels.



The camera has a Bayer filter to detect color. The pixel only measures the intensity of its corresponding color filter. 

# 2 Cougs in Space Raw Image (.csri)

This format of image is designed to store raw pixel data with possibility for compression to reduce transfer time.

## 2.1 Header

The header identifies the details of image contained and the expected length of data. The header is four bytes long or eight bytes long. The first byte contains the image details as formulated below. The other three represent the number of 32 bit words in the data block in big-endian order. The additional four bytes are for custom resolutions, first two for width, next two for height.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image Details [7..6] | Image Details [5..4] | Image Details [3..2] | Image Details [1] | Image Details [0] |
| Resolution | Location | Bayer Filter | Bit Depth | Compression |
| 0 = VGA | 0 = NW | 0 = RGGB | 0 = 8 bit | 0 = No compression |
| 1 = 2000 x 1500 | 1 = NE | 1 = BGGR | 1 = 10 bit | 1 = Compression |
| 2 = 2592 x 1944 | 2 = SW | 2 = GBRG |  |  |
|  | 3 = SE | 3 = GRBG |  |  |

3 = Other – Specify by appending 16 bit width and height after header (4 more bytes)

## 2.2 Data

The image is stored as a bitmap, which could be compressed. In 8bit mode, each byte represents a single pixel. In 10bit mode, each 32bit word contains 3pixels. Pixels are stored starting in the top left corner, read across the columns, then down the rows. Padding is added to each row to make it a multiple of 4 bytes long.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pixel 0 | | | | | | | | Pixel 1 | | | | | | | | Pixel 2 | | | | | | | | Pixel 3 | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bit | 31..24 | | | | | | | | 23..16 | | | | | | | | 15..8 | | | | | | | | 7..0 | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Null | | Pixel 0 | | | | | | | | | | Pixel 1 | | | | | | | | | | Pixel 2 | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bit | 31..24 | | | | | | | | 23..16 | | | | | | | | 15..8 | | | | | | | | 7..0 | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dword *n-1* | | | | Dword *n* | | | |
| Pixel | *n-4* | *n-3* | *n-2* | *n-1* | *n* | Padding | Padding | Padding |
| Byte | 0xFF | 0xDD | 0x01 | 0x69 | 0x99 | 0xFF | 0xFF | 0xFF |

### 2.2.1 Compression

The image data can be compressed to save space. This is indicated in Image Details[0], and the data size bytes need to be updated to reflect the change. The data block is contains all the information the compression uses. The compression is lossless and must be able to run on a microcontroller with limited memory. The chosen compression algorithm is miniLZO, <http://www.oberhumer.com/opensource/lzo/>. It compresses a block of data at a time which allows it to minimize memory usage. It works well for redundant data such as images.

# 3 Image Pipeline

This section describes the pipeline a image takes from initial capture to viewing on the ground.

## 3.1 Capture

The camera takes a photo of a subject and transmits the raw pixel data to the IHU. Throughout this section, the camera is taking a photo of earth:



The following image is the raw image captured by the camera, 100 x 75 pixels, 8 bit, 7.5kB. This image was generated from the image above. Notice the effect of the Bayer filter creating a checkerboard effect.



This raw image format is stored in temporary memory until the IHU can transfer it to the onboard SD card.

## 3.2 Satellite Storage

The IHU converts the raw bitmap into the Cougs in Space Raw Image format (.csri). It does this by compressing the bitmap and adding the header information.

## 3.3 Decompression

The image is received on the ground as a csri file. The ground control software reads the header information and accordingly decompresses the data block back into a bitmap.

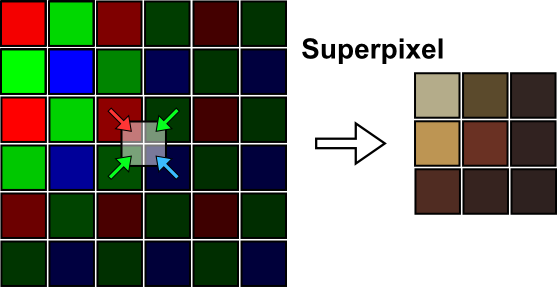
## 3.4 Debayering

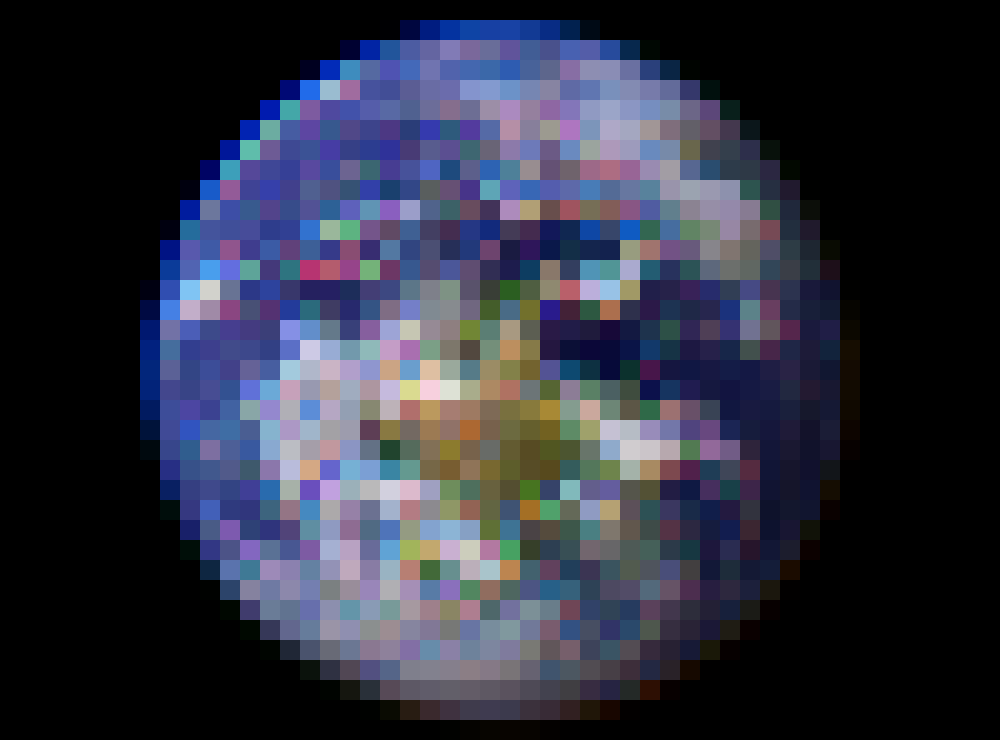
The image is a grayscale image through a Bayer filter. The image is converted into a full color image by interpolating the missing color information in a process called debayering. There are various method to debayering.

Before debayering, the image is as below:

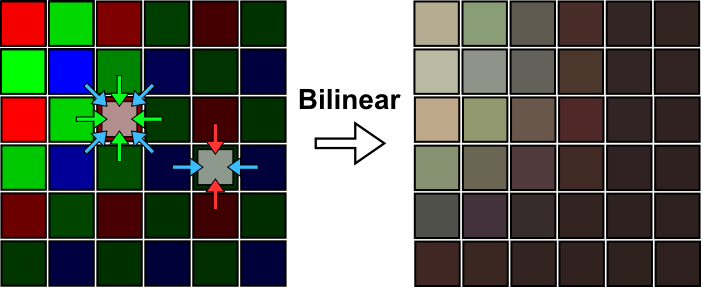


The following image depicts the superpixel method. The resolution of the image is halved on each dimension to merge four subpixels together in one pixel.





Another method is bilinear interpolation, as seen below. A pixel’s independent color channels are interpolated from the average of the pixels nearby.



## 3.4 Ground Storage

Once the image has been debayered, it can be exported as a standard png file. The raw csri file shall be stored as well for archival purposes.