# How to convert Sequoia images to reflectance?

| Version | Description | Date |
|---|---|---|
| 1.0 | Initial release | 14 September, 2017 |
| 1.1 | FAQ section added<br>Updated Irradiance List section | 16 September, 2017 |
| | | |
| | | |
| | | |

## 1. Motivation

The main motivation behind this document is the frustration that I had to face because Parrot never made the above question easy. I hope this document proves to a valuable tool for those who want an answer to the above question.

## 2. Acknowledgement

Before starting I would like to acknowledge a number of people on the Parrot developer forum who helped me understand it and indeed it is a collaborative effort and by no means am I the only person to be thankful to.

- seanmcleod
- pk123
- domenzain
- clement.fallet

(Pardon me if I forgot someone)

## 3. Introduction

The basic idea behind whole process is that you measure irradiance using your sequoia and sunshine sensor to estimate the reflectance of object seen by your sequoia. This is done by a simple linear equation with a coefficient $K$ which needs to be computed using a calibration panel. Both sequoia and sunshine sensor irradiances can be easily computed using their respective equations.

## 4. Methodology

Let $I_{sq}$ be the irradiance captured by sequoia at an observation angle of $\theta$ and $I_{ss}$ be the irradiance captured by sunshine sensor. Then reflectance $R$ associated with any pixel value is given by

$$R\ cos(\theta) = K\frac{I_{sq}}{I_{ss}}$$

Where $K$ is the *calibration coefficient* which needs to be determined by calibration methodology before each flight. It should be noted that above equation assumes that sun is at nadir (directly above) and sunshine sensor is directly facing the sun.

### a) Calculation of sequoia irradiance

Calculation of irradiance of sequoia is trivial as document SEQ-AN-01 discusses it in detail. Please refer to that for more details on this subject.

### b) Calculation of sunshine irradiance

Calculation of irradiance of sunshine sensor is discussed in detail here.

$$I'_{ss} = k_1(\frac{v}{g\tau}) = k_1 I_{ss}$$

Where $k_1$ is the coefficient of proportionality, $v$ is CH0 count, $g$ is relative gain factor and $\tau$ is the exposure time of sunshine sensor. All these parameters can be determined using the `IrradianceList`. This list contains a number of sunshine readings for each set of sequoia images. Irradiance for each CH0 value can be computed and an average of all or just last few can be used. It merits mentioning here that gain in `IrradianceList` is just the gain index and effective gain $g$ has to be computed (refer to related section). Exposure time $\tau$ is readily available in the decoded `IrradianceList.`

**Note:** It may seem that we are ignoring $k_1$ here but it is quire justified. The effect of $k_1$ automatically finds itself during the determination of $K$. Therefore, over here we don't need to compute it as can directly use $I_{ss}$ instead of actual $I'_{ss}$.

### c) Calibration coefficient

Determination of calibration coefficient $K$ is an important component in the process of calculating reflectance. It relates the ratio of sequoia irradiance to sunshine sensor irradiance. Idea behind the calculation of $K$ is that first you determine $I_{sq}$ and $I_{ss}$ using a calibration panel with known reflectance $R$. Since all the unknowns in equation are known, $K$ can be easily determined using the following form:

$$K = R\frac{I_{ss}}{I_{sq}}\cos(\theta)$$

Where $\theta$ is the angle of observation of calibration panel.

## 5. IrradianceList

`IrradianceList` is a tag in sequoia exif which contains sunshine sensor data for that particular shot respective to its camera. It is however base64 encoded and thus needs to be decoded first. Once decoded it becomes an array of structures defined below. It should be noted that data is little-endian.

| **Title** | **Size (bytes)** | **Description** |
|---|---|---|
| Time stamp | Uint64 | Monotonic time stamp in microseconds |
| CH0 count | Uint16 | Sensor count value (see sunshine sensor datasheet) |
| CH1 count | Uint16 | Sensor count value (see sunshine sensor datasheet) |
| Sensor gain Index | Uint16 | Gain mode being used to take reading |
| Integration time | Uint16 | Time of integration in milliseconds |
| Yaw | Float | Sunshine sensor yaw in degrees |
| Pitch | Float | Sunshine sensor pitch in degrees |
| Roll | Float | Sunshine sensor roll in degrees |

**NOTE:** `IrradianceList` base64 encoded string is preceded and succeeded by a dot (.) which is not part of base64 encoding. Make sure to remove the starting and ending dot (.) before attempting to decode it.

## 6. Relative gain factor

Sunshine sensor gain can be computed `IrradianceCalibrationMeasurement` exif tag. Consider the following list as a sample.

```
0,600,11,3,1,600,330,84,2,600,3990,1506,3,600,65535,38630
```

The above list can be rearranged as a 4X4 matrix. Each column has the title: **sensor gain index**, **integration time, CH0 count, CH1 count**. It should be noted that integration time and exposure time are being used in this document interchangeably as they are essentially the same thing.

```
0,600,11,3

1,600,330,84

2,600,3990,1506

3,600,65535,38630
```

It should be noted that sunshine sensor has 4 gain settings like any off the shelf ADC. Each row indicates CH0 and CH1 readings for its respective gain settings. Since each device manufactured has its own particular gain due to manufacturing processes so it needs to computed for each and every device. These readings have been taken for same level of irradiance but under different gain settings. This means that relative gain factor of this device in mode 2 with respect to mode 1 is $3990/330$. Similarly, relative gain factor of CH0 in mode 0 with respect to mode 1 is $11/330$.

**Note:** In practice, it really doesn't make any difference and relative gain computation is not required as long as you are taking all the measurements in one gain setting. But if you change the gain mode, the device becomes more sensitive and records a higher number for the same level of input signal. Thus, in order to compare readings taken at different gain levels, one needs to normalize the readings by the relative gain. But unfortunately, as sequoia users we don't have control over gain mode. So, it doesn't hurt to compute this factor and normalize.

In my experience, sunshine sensor always takes the reading at mode 1. This means that all the CH0 values can be compared with each other and doesn't require any normalization by relative gain factor. However, that may change under different lightening environments or with firmware upgrades of sequoia.

Since CH0 is a 16 bit number, it is evident that in case of gain mode 3, the value is simply the max number it can hold and doesn't contain the actual number. Thus it is not possible to compute the relative gain factor for this device for gain mode 3. However, relative gain factor for CH1 can be computed since it is within range.

**Programming Advice:** Since relative gain factor only depends on device manufacturing, therefore it can be computed beforehand and just stored in the form of array for each gain mode. Following array can be initialized for the example given above:

```
RelGainFac[3] = {11/330, 1, 3990/330}
```

## 7. Frequently Asked Questions

### a) Why use $I_{ss}$ instead of $I'_{ss}$ when using the later seems more logical?

If we use $I'_{ss}$ then it means that we need to compute $k_1$. Nobody wants to do that for two reasons. First, we are lazy and we don't want to do extra homework. Second, even if we are not lazy, it's no trivial task to do so. Doing so would require you to know the ground truth of irradiance i.e. exact value of irradiance, which is only possible (as per my understanding) in a controlled facility inside some physics laboratory. And even if we defy all the odds and still measure it, there is not much significance for that. We are already calibrating our camera by $K$ and a little amount of algebra will show that $K$ has the impact of $k_1$ embedded in it.

Hint: Remember how we combine all the constants while solving differential equations.

### b) Relative gain factor is not clear to me. Please explain more.

Well, I believe the related section above pretty much explains it. However, I would try to make it even more CLEAR here. (*Don't worry I ask my math instructor stupid questions all the time and he never minds it; so neither gonna I*).

Suppose you fly a mission and collect a bunch of pictures each of which has a bunch of sunshine sensor readings. That's great. You feel exited that you can uncover the truth about your vegetation now. But wait a second, one of the most stupid programmers in your groups shouts that he made a shocking discovery. Right on his computer screen sits a plot of gain index bouncing up and down, pulled out of all the sunshine sensor readings (out of all the images taken during mission) with respect to time. You feel pissed at him because it really doesn't matter as long as your have CH0 value from sunshine. Wrong! You should be thankful to him. Why? Because he saved your ***.

Suppose for a particular irradiance, your sunshine sensor records CH0=`430` at gain index=1. Very next reading indicates that CH0=`5200` at gain index = 2. You feel irritated and lost as to how is it possible. How did CH0 value could jump so much in so little time! Its simple; your sunshine sensor just changed its gain settings and its evident in gain index=2. All you need to do is normalize your value back. Now you can't make any comparisons among CH0 as long as they are under different gain settings. You need to normalize all of your readings back to one standard. And that standard is nothing but just one of the gain indexes that you select for yourself. For me I like gain index=1 because all of my readings (or most of them) are already at gain index =1. So it just make psychological sense to me to normalize them back to index=1. But if you want to select index=0 or index=2, that's perfectly fine as the choice of reference gain index (standard gain index) is totally arbitrary. However, I couldn't make use of gain index=3 because in that settings my sunshine sensor gets overflown as the max number it can capture is 16 bit (`65535`). So I would also caution you to proceed carefully with gain index=3.

As to how do I calculate my gain value, I would just reuse the numbers in example above. Suppose that I take a reading from sunshine sensor at gain index =1 and CH0 contains value `330`. Now without changing any irradiance, I retake the reading but now at gain index = 2. This time it returns me `3990.` You should note that actual irradiance value has not changed. The changing value of CH0 is due to different gain settings being employed by the circuitry of sunshine sensor (talk to a guy who knows Analogue to Digital Converter and (s)he will tell you about the different gain setting inside it). So all you need to know is how big your reading becomes when you take it from gain index=1 to gain index=2! I would calculate it to be `3990/330 = 12.09` which is my relative gain index for index=2 with respect to index=1. And then whenever I will get a reading at gain index=2, I would like to get the equivalent value for gain index=1. So I would just divide that value by 12.09 to put it back to gain index=1.

Please note that each device can have different gains in different configurations so don't use mine over here. Compute your own using the `IrradianceCalibrationMeasurement` tag. You just need to apply a simple mathematical formula.

## c) How do I decode irradiance list?

Well, you can use any programming language to do that. As far as I know, MATLAB®, python and C# have built-in support for that. Most of the other languages would also have some package/library/API doing that. And even if some language doesn't support that you can easily write your own piece of code. Wikipedia has a good article on how BASE64 encoding/decoding works. Infact, it was where I leant BASE64 encoding/decoding.

## DISCLAIMER

The language used in this document doesn’t mean to offend anyone. It’s just that I like to write lightly because it’s kind of boring to read a research paper.