Chapter 4 : Process and Results

This chapter will explain the process and results of the research done. This chapter will be divided to a few sub-chapters according to the tasks done. In the beginning you will find more explanation of the main problem. Then it continues to explain the way to gather information to solve the problem (gathering data) and the technical solution to the problem. Finally, it ends with prototype implementation and testing.

4.1. Day Mode and Night Mode

The security cameras are expected to be able to run continuously for 24 hours and able to see the scenes in good condition at all times. To cope with that, each camera has a day mode and a night mode to be able to get optimal pictures in bright and dark scenes. In day mode, where there is enough light, the camera will operate normally and generate color images. In night mode, where it is very dark, the camera will remove infra-red (IR) filter, allowing the camera to detect IR light, turn on IR LEDs to brighten the scene with IR light, and generate black and white images.

Normally, luminance is used to classify between dark and bright image. Luminance is a value used to generally measure how bright an image is, and is often used in day and night recognition. There is no problem with using luminance for the day to night switching, but problems will occur when using luminance for night to day switching. What problem? **In night mode, even in dark, the image acquired has high luminance value rivaling bright scenes.** Why? This is because infra-red light also counts into the luminance. If left alone, when luminance drops to certain point and switch to night mode, the activated infra-red light will increase the luminance back and possibly cause the camera to immediately switch back to day mode (oscillation). Therefore luminance alone cannot solve the issue and another method to classify day and night image is necessary. This is one of the reasons why day to night and night to day switching is separated.

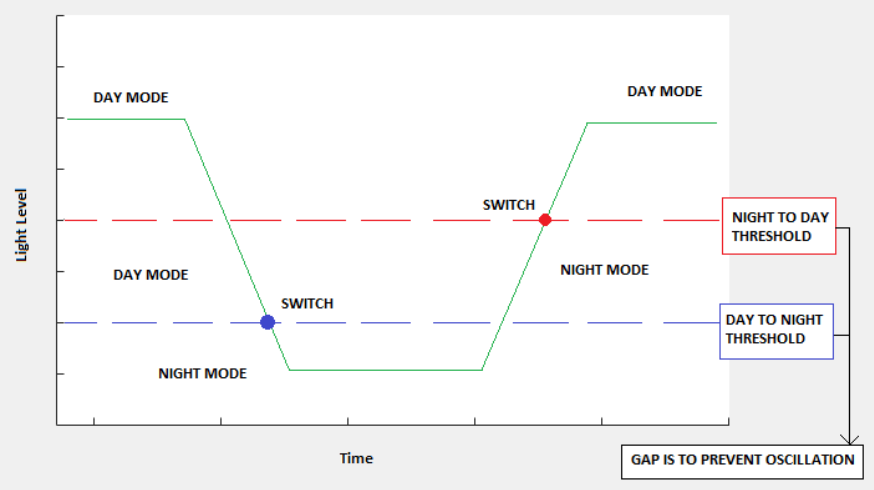

Image 4.1.1. Figure of day-night switching.

4.2. Gathering Data from Camera

During the research phase, library strategy alone does not give enough information about the topic. Then how to collect more information? The easy way to understand the working of the camera is just to try it out! Thus, field strategy is added to collect more information. A command line interface (CLI) program and a testing camera is provided by the company to connect and create log files from camera's internal data.

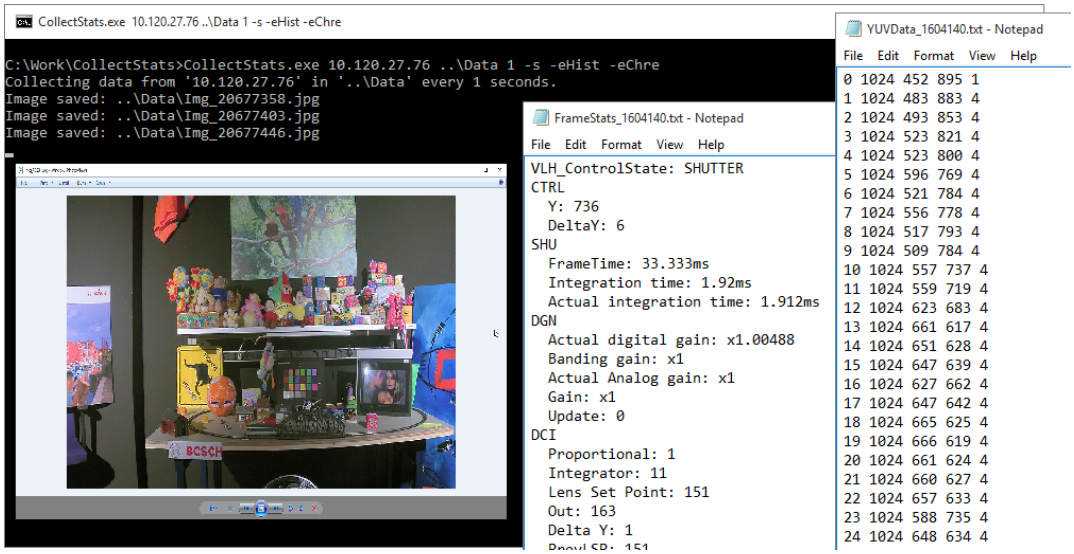

Image 4.2.1. Screenshot of company's logging tool and sample log files

The program allows a lot of data to be collected, including some important ones like: R/G B/G ratio values from the sensor, color temperature, color gains, integration time, and etc. Because of the image pipeline design, the full resolution of the picture's data cannot be obtained, instead, the image is divided into 32 by 32 grid then summed up (so the result is data with 32x32 pixel).

The data collected contains a bunch of numbers and some raw data which are obviously not very easy to read and understand. Therefore, the data is processed to show it in a more human-friendly way. The company tutor gave 2 choice of tool to visualize the data: either using MATLAB or C# application.

MATLAB is a numerical computing environment/platform developed by MathWorks. It is nicely optimized and especially friendly to matrix and statistic calculations which is crucial in the project. MATLAB is also able to create graphical user interfaces and is very nice to visualize the data collected from the camera. C# on other hand, is intended for general-purpose programming. It requires some additional plugin or library to compute complex calculation and display customized graph/plot. By comparing the two, it is obvious that MATLAB holds advantages over C#. Hence, the data is visualized via MATLAB GUI.

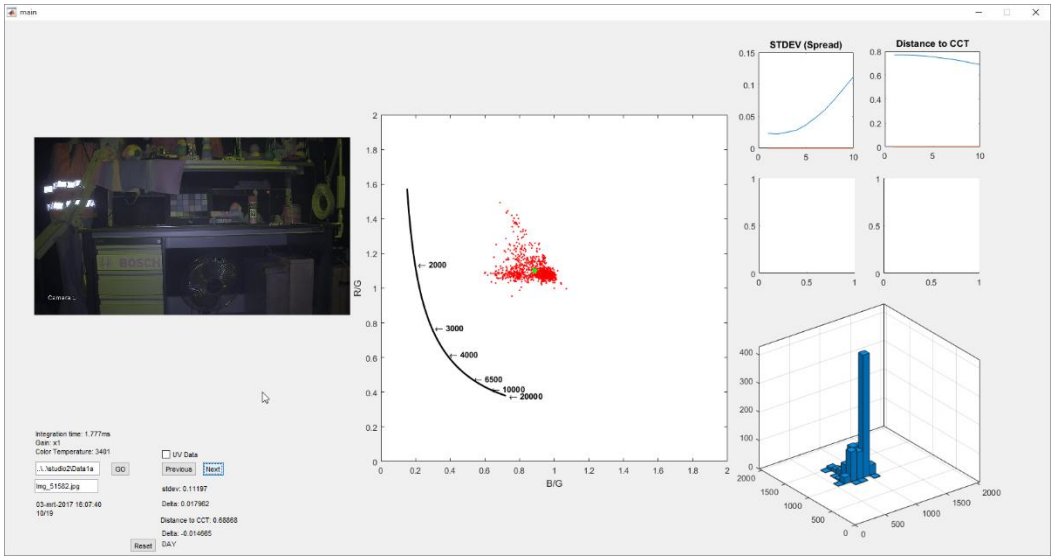

Image 4.2.2. Processing log files with MATLAB GUI

Later on, it is found out that the provided logging tool was not able to log all necessary data that is related to the night to day switching (explained in Chapter 4.3). Because there is no access to the logging tool source code, a new logging tool is developed using Microsoft Visual Studio in C# programming language. The new logging tool is made to be able to log more data from the camera (specifically raw RGB values, number of underexposed and overexposed pixels, and configuration values). In addition, the new logging tool is also able to control the light-room/studio (provided by company) and set camera settings automatically according to the user input. This allows collecting desired data more practical and easy.

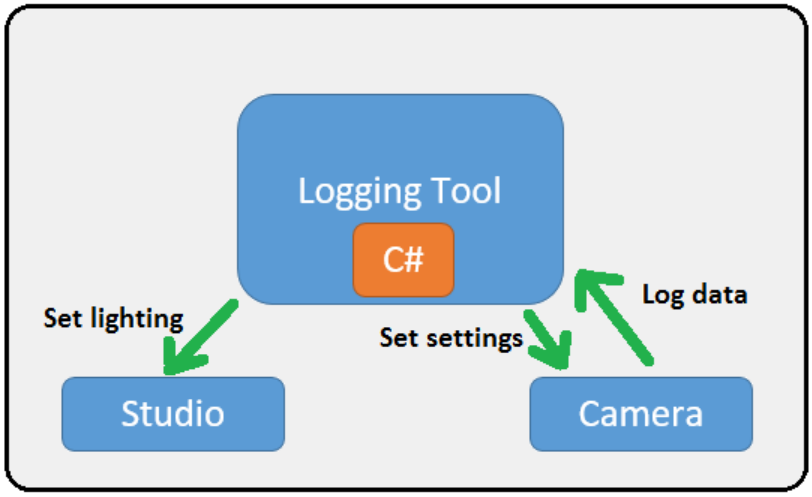

Image 4.2.3. Figure of logging tool process

Why Microsoft Visual Studio and C# programming language? Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft that is used to develop various computer programs. It has a lot of code editor supports which allow easy programming. C# is a high level programming language within .NET Framework that is supported by Microsoft Visual Studio. C# is intended for general-purpose and allows simple programming. The reason to use C# is because the external library available to control the light-room are .dll files which supports C#. The project examples (how to use the external library) is also in C#.

4.3. Observation Results

By evaluating and comparing the camera's log files with different scenes and conditions, a few patterns and behaviors are found. There are 3 major factors that affect these data, and those are:

- Ambient light intensity
- Infra-red light intensity
- Camera settings.

Ambient light is the normal light that our eyes can see, while infra-red light is invisible to human eyes. This is the light that is used during night mode and can be increased by activating the camera's IR LED. There are many settings/configurations in the camera, such as shutter speed, IR LED intensity, etc. Some of them can affect the data received, because depending on the settings the image received can greatly changes.

The camera settings that greatly affect the data are:

- IR LED intensity
- Automatic Level Control (ALC)
- Exposure / shutter speed

To investigate the color information the camera gives, the RGB color space outputted by the camera is mapped into R/G and B/G to allow all RGB values to be visualized into a single 2D plot (refer to Image 4.3.1 until 4.3.4). The result shows that if the only light source is the infra-red light, it only has one color, but more colors will start to appear when there is more ambient light. R/G and B/G values show the diversity of color in the image obtained. By analyzing these values, the process of the color changes can be observed.

Observation of R/G B/G graph with default settings (except IR LED intensity) has resulted as follows:

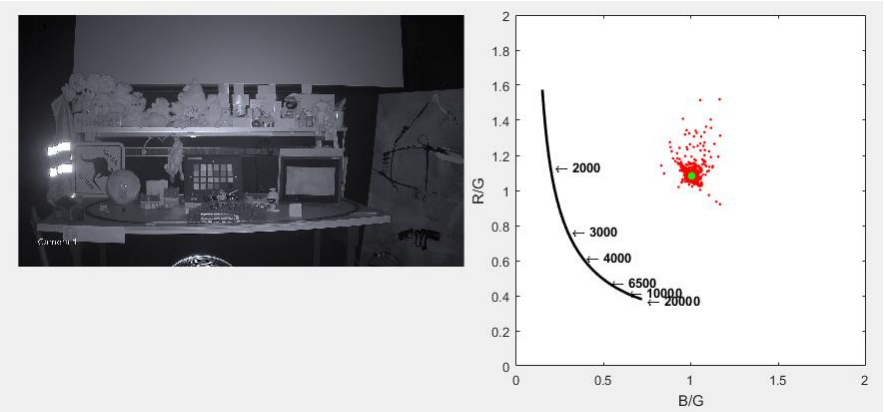

Image 4.3.1. R/G B/G graph of high IR light and low ambient light (Night)

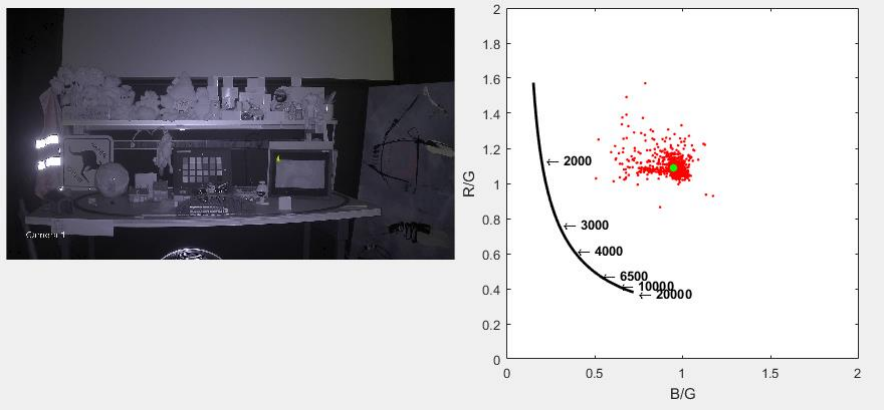

Image 4.3.2. R/G B/G graph of medium IR light and medium ambient light (Day)

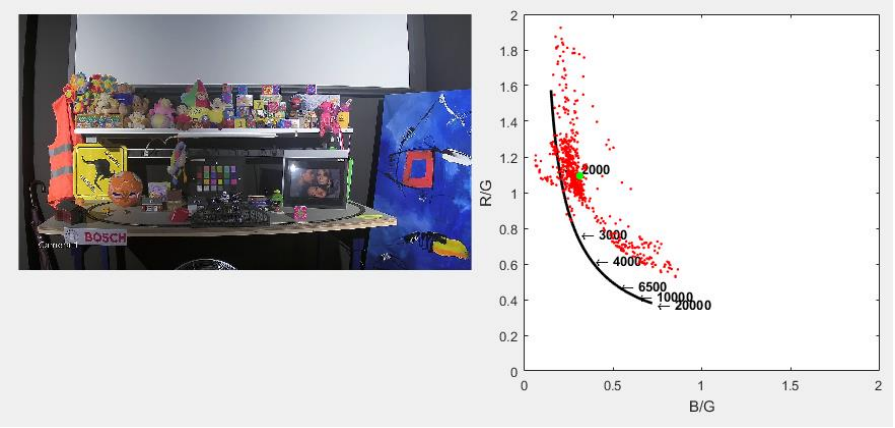

Image 4.3.3. R/G B/G graph of low IR light and high ambient light (Day)

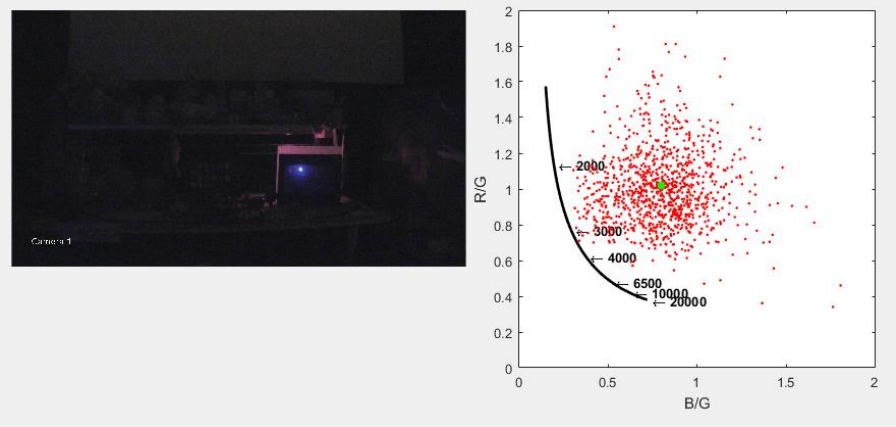

Image 4.3.4. R/G B/G graph of low IR light and low ambient light (Night)

From these data, we can now analyze the difference between day and night conditions based on the patterns and behaviors of the graphs. **Dark scene with enough IR light will display a black/white image and the R/G B/G values tends to accumulate around one point. The R/G B/G values will then start to move closer to the blackbody (the black curve) as the scene gets brighter.** These facts can be used as a way to tell whether it's bright or dark. It is also worth mentioning that RGB intensity increases along with the light level (brighter scene). Therefore, RGB intensity is relevant and also important in determining day and night condition.

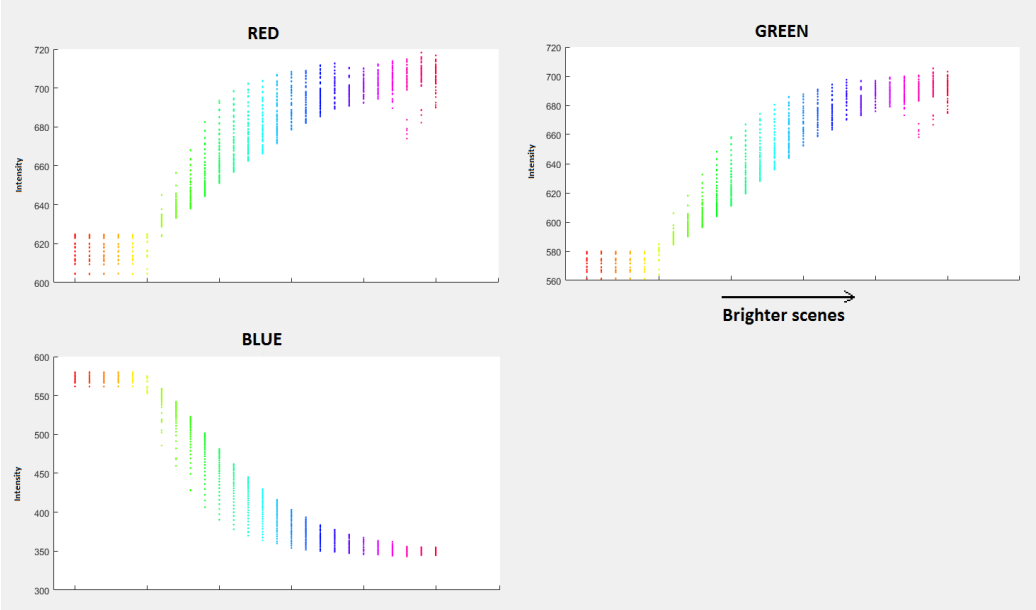

Image 4.3.5. Sum of RGB intensity of different dark/bright scenes.

As for when to perform the switching exactly, ALS can be used as a reference to determine whether it's night or day. This is because ALS is already an established way to determine day and night.

4.4. Determine Measurements

As you can see in the images of previous sub-chapter, it is possible to tell between day and night condition using R/G B/G graph. But without any prior knowledge, looking at the graph won't give you a clear idea whether it's day or night, because the information is still in qualitative form. Therefore, it's necessary to process these values to become more readable, such as numerical value to show how bright the scene is (lux/ambient light). Doing this is important, especially in the next step (Chapter 4.5).

From the previous sub-chapter, the camera's internal data that stands out and greatly changes when the scene brighten or darken are:

- R/G and B/G ratio
- RGB intensity
- Configuration values (camera settings)

R/G and B/G ratio hold importance of determining whether the scene is dark or bright (as mentioned in previous sub-chapter). RGB intensity can be calculated and processed into luminance, which can show how strong the light in one pixel/color is. Configuration values is just as mentioned in previous sub-chapters, can affect the data received.

All pixels in one frame of the received data needs to be processed statistically into a single value called measurements. These

measurements can be used as a reference point to determine when to switch.

According to CIE 1931 color space, a standard color space relating color physics and color perceived by human eye, RGB values contribution ratio to luminance is:

R : G : B = 1 : 4.5907 : 0.0601

As shown above, the green's weight overwhelms the other colors. Because of that, **it is decided that total luminance measurement is calculated using only the green values** (therefore, total luminance = total green values). This is an important step to reduce processing load while still having good values.

Based on the observation results in previous sub-chapter, R/G B/G values of scenes with IR light have the tendency to accumulate around one point. This is because there being only one source of light, that is IR light, and the image will become over-saturated. This can be used to separate between IR light and ambient light. The R/G B/G values within a certain distance from a certain point in the graph will be assumed as IR light, and the rest will be assumed as ambient light.

The certain distance and certain point that are separating ambient light and IR light, and is called Region of Interest (ROI). This certain distance (ROI size/radius) and certain point (ROI origin) are made tunable. This is to allow the algorithm to adapt in a flexible way and become reusable in various cameras that might have different sensors, processes or have updated features. But by allowing values to be tunable, this also means those values will require to have a good initial value to function properly.

With the ability to determine luminance, even more measurements can be calculated:

- Total luminance
- Ambient Sum / Luminance (pixels outside ROI)
- IR Sum / Luminance (pixels inside ROI)
- Ambient %
- IR %
- R/G B/G spread

From these measurements, it may be possible to determine day and night condition.

4.5. Data Classification

After determining the measurements, it is important to separate and classify the behavior of measurements in bright and dark condition. All the data collected are processed into measurements and displayed into

a 2D graph. The problem is which measurement is used and visualized in the graph and how to classify it.

Classification is done using ALS as references. By using the ALS, 3 classes can be determined:

- Night scene (light level under 50)
- Day scene (light level above 80)
- Either (light level between 50 and 80)

Light level is the number of output level of the studio lamps provided by the company which is used inside light-room.

The reason of having the "Either" class is to prevent the risk of switching from night mode to day mode and from day mode to night mode repeatedly, which is also known as oscillation (refer to Chapter 4.1). The day to night switching utilizes a completely different algorithm, so this risk is something important to be aware of. Therefore, the "Either" class is a way to distance the switching condition of day to night switching and night to day switching to prevent oscillation.

Now, how to choose which measurement to use? It is logical to choose the measurement which is most sensitive to light level changes. From the measurements mentioned in previous sub-chapter, ambient sum does show the best values from the others.

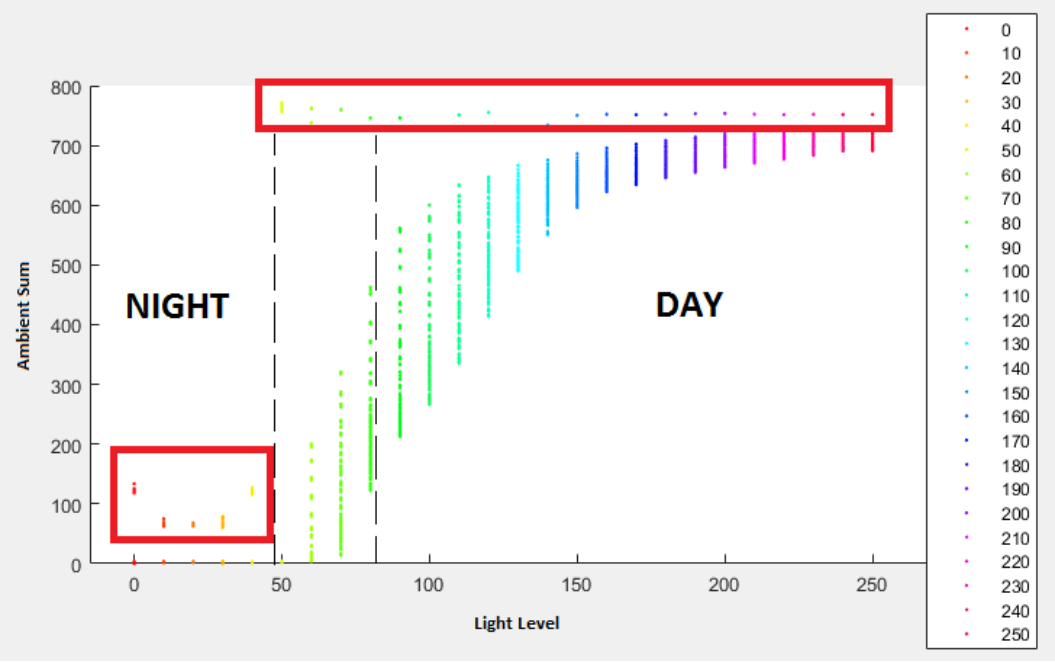

Image 4.5.1 2D plot between light level and ambient sum

The image above shows the ambient sum of the collected data using default settings under certain light levels with varying IR LED intensities. As you can see, ambient sum values increase significantly between night and day classifier. This is why ambient sum is appropriate measurement to use in the algorithm.

There are exceptions to this classifications:

- Dots marked inside the red boxes are night scenes (produced by using very low IR light)
- Data can yield greatly different results by changing the camera settings

The first exception, the dots inside the red box mark, poses a major problem during data classification. Using a very low IR light will create a dark image with a lot of noise (random pixels, grains in the image), resulting in a very unreliable data. **It is then decided to exclude very low IR light condition from the classification and also ignore such condition in the algorithm.** The reason being: all cameras that have day/night switch function are always accompanied by IR LED, so this problem of not having enough IR light in dark scenes is redundant.

If low IR light condition is ignored, then classifying the scenes is simple. Ambient sum measurement can show how bright and colorful the scene is, if the image is starting to show color then it means there is enough ambient light. As shown in previous image, it is proven that the brighter the scene is, the higher the ambient sum is. Therefore, ambient sum is suited for night to day switching and is best measurement to determine when the camera should switch.

4.6. Switching condition

In previous paragraph, it is proven that ambient sum is a reliable value to determine night to day switch condition. As observed in Image 4.5.1 (ignoring the data in the red box), ambient sum will rise above a certain value when it is starting to get brighter. Then it is important to determine this certain value and make that value as the night to day switching point (called threshold). When the ambient sum is higher than the threshold then the camera should switch to day mode.

It is mentioned before that measurements can be greatly affected by the camera settings. If the threshold defined is static, changing the camera setting can make the day night switching decision to miss. Then how to solve it? There is a need to adapt switching condition (threshold) under various settings. This method is also utilized by the ALS-based algorithm.

As mentioned in previous sub-chapters, IR LED intensity, ALC, and exposure can affect the data received. The algorithm must take account of these camera settings as well. The higher IR LED is, the less the ambient sum is. The higher ALC is, the higher ambient sum is. The higher exposure is (slower shutter time), the higher ambient sum is. Based on these settings, this formula is developed:

Threshold = L norm x IR multiplier x ALC multiplier x Exposure multiplier

L norm = static base value
IR multiplier = multiplier based on IR LED setting
ALC multiplier = multiplier based on ALC setting
Exposure multiplier = multiplier based on exposure setting

To get the threshold, the multiplier values must be calculated. The multiplier values are calculated like this:

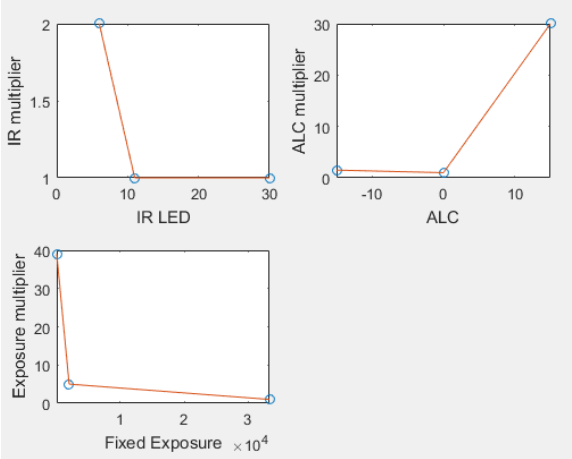

Image 4.6.1. Figure of multiplier calculation

The multiplier is calculated based on the 3 points which will be interpolated between each other using linear interpolation. The multiplier value is dependent to its respective settings (e.g. ALC multiplier is dependent to current ALC setting). These 3 points and L norm are also tunable like ROI size and ROI origin for the same reason. It is not recommended to use more complicated interpolation or curve, as it will add more computation load when implemented into the camera.

Beside the formula shown, the threshold also have to follow this rule:

Min Threshold ≤ Threshold ≤ Max Threshold

This is to prevent the threshold from overflowing or getting values that is too high or low. The minimal and maximum value of the threshold are also made tunable.

4.7. Camera's Firmware Modification

After developing a new algorithm, the question is "How to implement it in the camera?". To allow the new algorithm to be implemented in the camera, modifying the camera's firmware is necessary. It must disable the current night to day switching using ALS and use the new algorithm instead.

While the ALS algorithm can be disabled, the new algorithm needs to go through many processes of testing and validation before truly implemented into the camera's firmware. So in this case, the firmware is modified to be able to accept command signals to switch from night mode to day mode, which will be sent by an external program (the prototype software).

The tool and language used is Microsoft Visual Studio and C++. There is no other option for this, because that is what the company has used to develop the camera firmware. The firmware source code is pulled from GIT, which is the version control system used by the company.

4.8. Prototyping

The next step to the complete the algorithm is to implement it into a working software to execute night to day switching. The tool chosen for this is MATLAB. This is because most of the measurement calculations are already in MATLAB (when visualizing the log files). There are also a lot of matrix calculations, so MATLAB is still the proper way to do this.

The application made need to have these following functionalities:

- Connect to a Bosch security camera
- Calculate measurements
- Change tunable values
- Execute night to day switching

Connecting to a Bosch security camera is required to obtain necessary data used in the algorithm. These data will be then processed into measurements. The prototype uses an external program to connect and obtain these data from the camera. The external program is derived from the logging tool and is developed in C#. Why use an external program? MATLAB is not optimized for string operation and therefore, is slower than using an external program.

Next, the threshold needs to be calculated. The Threshold is calculated based on current camera settings and some tunable values. Changing the tunable values is also crucial, as this application is made for the company to easily try out the algorithm in various way by tuning those values.

Finally, how to execute the switching from night mode to day mode? There is no way to send the switching command directly from the MATLAB program to the camera. Instead, the application requires an external program called Metis to write directly to the camera registry to perform the switching. Metis is a software developed by Bosch Security Systems for their security camera development processes and is intended for internal use only. The prototype will send a command to Metis to write a specific registry and the modified camera firmware will read that specific registry in order to switch to day mode.

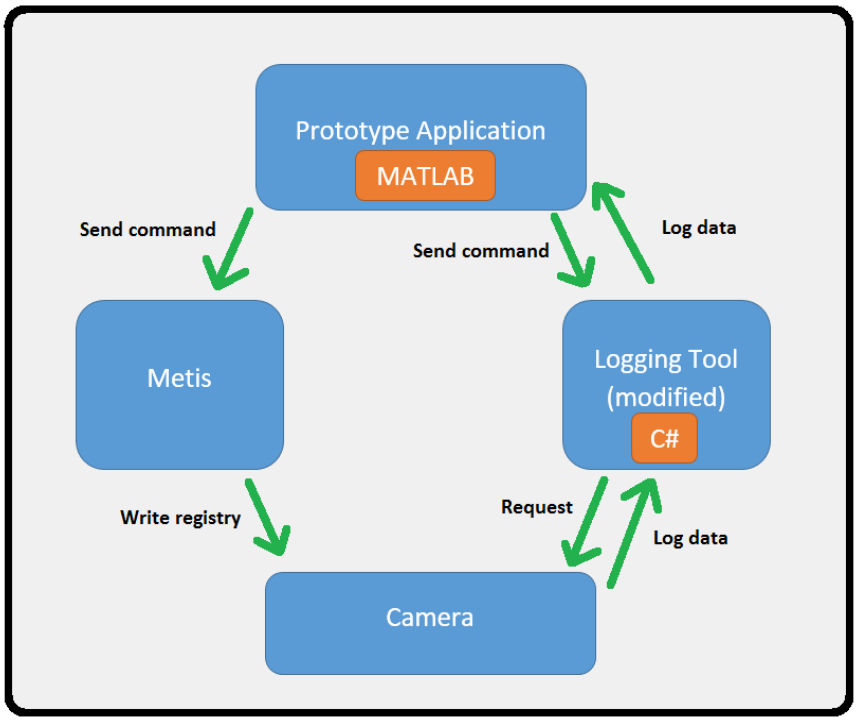

Image 4.8.1. Figure of application process.

From the abovementioned functionalities, this GUI is designed:

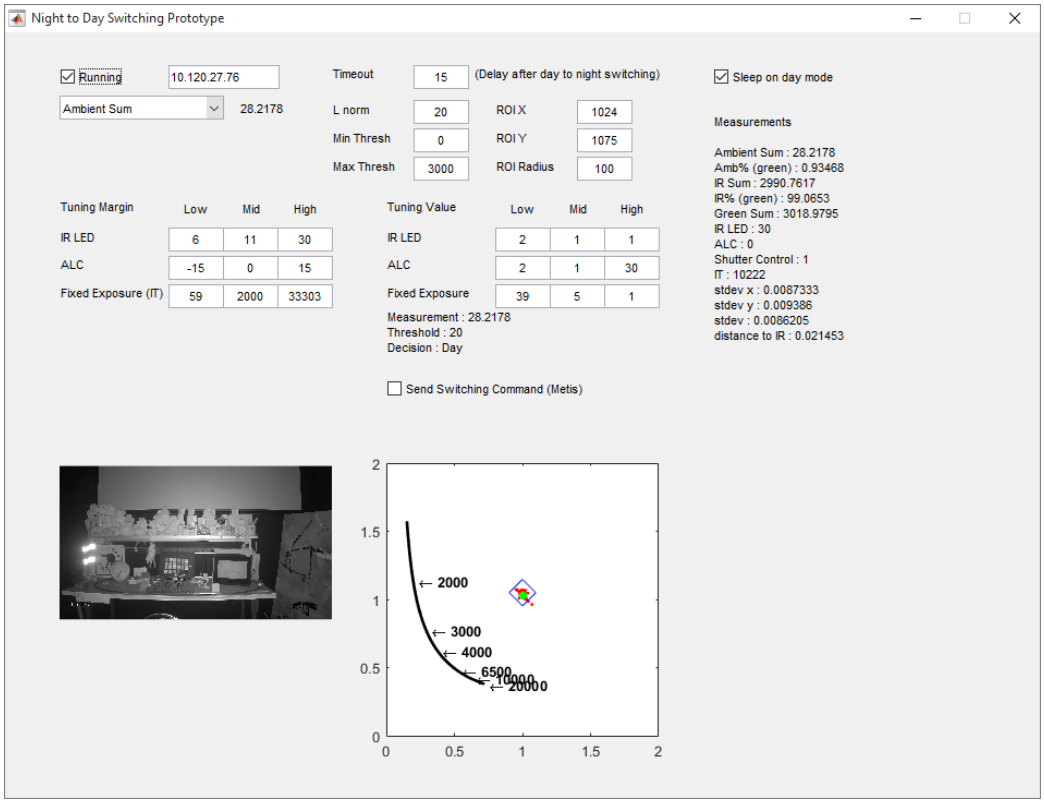

Image 4.8.1. GUI of the prototype application.

***** Details of GUI design can be found in Attachments

Beside the required functionalities, there are some more added features. They are timeout and sleep function. Timeout is a functionality to prevent the oscillations immediately after switching to night mode by temporarily disabling the night to day switching. When switching to night mode, the camera needs a little time to adjust to IR light before giving proper image. Without timeout functionality, the camera may switch back to day mode immediately if the camera is capturing images that is too bright. The duration of the timeout is tunable because the time needed for the camera to adjust varies.

The other functionality is sleep. There is no need to perform calculations at all times. At day mode, the program will skip all process until the camera switch back to night mode. The user is allowed to switch this option off if desired.

4.9. Testing

With the completion of prototype application, it is now possible to test and compare it with ALS. The comparison is done by getting the switching point of the two night to day switching mechanism and compare the result. The target of the new algorithm is to get values as close as possible to the values from ALS.

Because the way of the two algorithm operate is different, it is expected to have some difference, hence 100% accuracy is not possible. Thus, the comparison is done using mean squared error and absolute error method. Both are methods to measure how close are the predicted/resulting values compared to the target values.

$$E_i = \frac{1}{n}\sum_{j=1}^{n}\left(P_{(ij)} - T_j\right)^2$$

Image 4.9.1. Mean Squared Error formula (gepsoft, 2014),
P is predicted value and T is target value

$$\epsilon = \left| v - v_{\text{approx}} \right|$$

Image 4.9.2. Absolute Error formula (Wikipedia, 2017),
V is value to be compared and Vapprox is the approximated value

Between the two methods, absolute error method gives out more realistic number. Then what is the point of using mean squared error? In fact, mean squared error is more widely used. Mean squared error generally gives bigger error number than absolute error, but squared error has a convenient mathematical properties. These properties may not matter now, but it is more preferred and may matter more in the future.

Here is the comparison result between ALS performance and prototype application with initial tunable values:

No.	Camera Settings			Light intensity the camera switched at	
	Exposure	**ALC**	**IR intensity**	**ALS**	**New algorithm**
1	1/30	0	30	52	65
2	1/30	0	6	51	52
3	1/30	15	30	63	69
4	1/30	15	6	63	54
5	1/30	-15	30	51	67
6	1/30	-15	6	49	64
7	1/500	0	30	79	66
8	1/500	0	6	80	53
9	1/500	15	30	131	82
10	1/500	15	6	131	58
11	1/500	-15	30	61	70
12	1/500	-15	6	60	54

√ Mean Squared Error = 28.20 Absolute Error = 19.83
√ Max Mean Squared Error = 185.43 Max Absolute Error = 183.67
% Mean Squared Error = 15.2% % Absolute Error = 10.8%

***** Data shown in table only use 1 type of light (RGB+Y Full). Complete test result can be found in Attachments
****** The light level is the output level from the studio lamp available in the light-room provided by the company

The resulting switching points shown in the table are somewhat off the target. This is because the tuning values are not properly tuned yet. The values used are initial values which are based on only a few data samples. But if desired, the experts in the company can change the tuning values themselves to get a better result from the prototype application.

Overall, the result is acceptable because there is no oscillation problem and the computation is not too tasking. It can be concluded that the developed algorithm is a viable option for night to day switching mechanism.