# PT Grey Image Acquisition

## Documentation and User Guide

Version 1.0

December 3rd, 2018

Rachel McIntosh (rmcintos@mit.edu)

MIT Media Lab: Conformable Decoders

# 1. Overview

PT Grey Image Acquisition is a command line-based python script that acquires and saves images from a setup of multiple Point Grey Blackfly GigE cameras. The script utilizes FLIR's Spinnaker SDK and PySpin Python library to interface with the cameras. The user can choose from three different modes of image acquisition: manual, timed and continuous. After image acquisition is completed, all images are rotated 270 degrees for easier viewing and saved under the PNG file format. All images are taken in grayscale, and image naming follows the requirements specified for use with MultiDIC.

# 2. Program Flow and Details

## 2.1 General Program Flow

The program has five main steps, as shown in figure 1a. Step 1 is to gather input from the user on the script and camera settings. Step 2 is to prepare for image acquisition. This includes establishing an instance of a PySpin system, initializing the cameras in accordance with the user-defined settings, and creating the folders to save the images to. The script creates a main folder of the format "Camera Run Year-Month-Day Hour Minute Second". Within this folder, folders for each camera are created of the format "Camera Number". Step 3 is to query the cameras for images and temporarily store each image pointer in a Python dictionary; image saving is put off until image acquisition is finished to decrease the time required between each round of images. Step 4 is to save all images in the appropriate camera folders. Images are saved in the format "camXImage_Y" and are saved under the PNG file extension. Images are also rotated 270 degrees for easier viewing. Step 5 is to deinitialize the cameras and then release the system instance and the camera pointers, as is required by the Spinnaker library. Failing to complete the final step may interfere with future runs by causing the Python Kernel to restart or causing the cameras to malfunction. This can be solved by closing the command window or IDE that the script is being run from and/or turning the cameras off and then on again.

## 2.2 Settings

The user can choose from several settings, including the image acquisition mode, the image color mode, the image saving locations, and the use of pre-defined camera nicknames. There are three acquisition modes: manual, timed and continuous. In manual mode, the user presses the enter button to take an image. In timed mode, the user specifies the total number of images to take and the time delay between each image. It is recommended that time delays are 1 second or longer. In continuous mode, the user specifies the total number of images to take. The script then takes these images as quickly as possible. Figure 1b depicts the required settings for each acquisition mode. Cameras are set to "single frame mode" for manual and timed modes.

Image color mode refers to the choice between mono8, mono12, and mono16; the numerical value refers to the number of bits per pixel in the image. All images are taken in grayscale as MultiDIC uses grayscale images and the amount of data within the image is decreased, thus

increasing the speed at which the script can acquire and save images. There are pros and cons to each image mode. Mono8 allows for the fastest possible acquisition, and thus it is recommended for continuous acquisition mode or timed acquisitions with a small delay. However, mono8 results in the lowest resolution images. Mono12 is recommended for manual and most timed acquisition runs, as it provides the ideal balance between resolution and speed. Mono16 is not recommended, as it slows the script significantly during the saving process and often results in corrupted images.

The user can also specify the directory to save the images to by entering a valid path. The default path listed was used for camera runs with the specified equipment setup. The option to use pre-defined camera 'nicknames' was also used with the specified equipment setup. When enabled, folders and images are named based on the number listed on the camera in the specified setup – for example, camera 1, camera 2, etc. When disabled, the camera's serial number is used for naming instead.

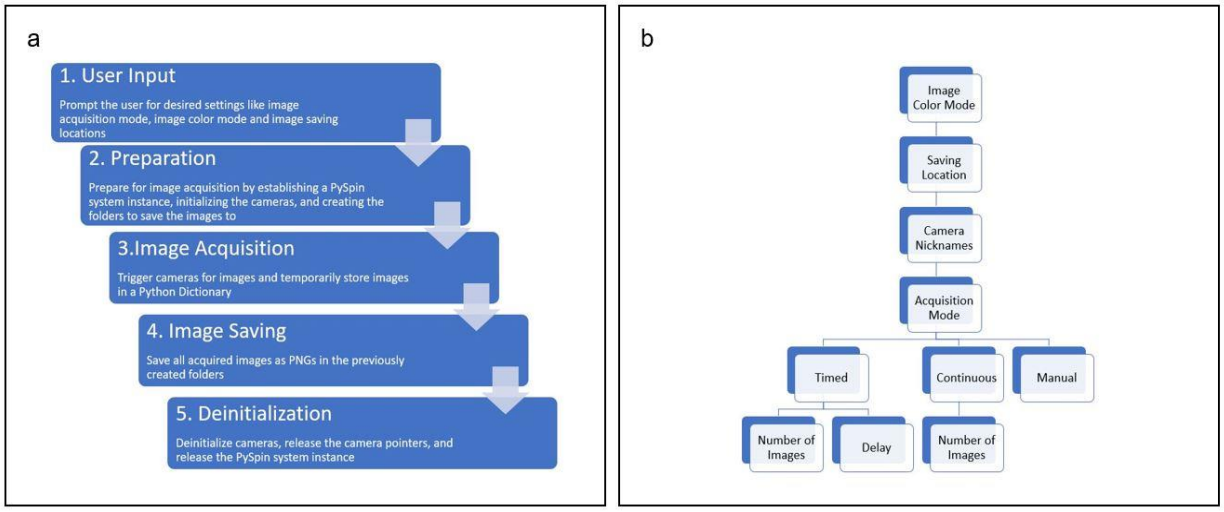


*Figure 1: Program flow (a) and settings diagram (b)*

## 2.3 Image Acquisition and Saving

Although the triggers for acquiring an image differ based on acquisition mode, images are acquired in the same way regardless of setting. After the user or script triggers acquisition of a new round of images, each camera is sent the command to begin image acquisition by looping through the list of camera pointers. After all cameras are ready for acquisition, the list of cameras is looped through once again. Each camera is queried for the next image, which is stored in an image buffer. A deep copy of this image is created and added to the Python dictionary of images; the keys for this dictionary are the names of the images in the format "camXImage_Y". The image buffer is then released to prevent buffer overflow; note that failing to copy the image and releasing the original image buffer can result in the program hanging during acquisition. When in continuous mode, this process (i.e., looping through the cameras to query for a next image) will occur the number of times specified by the user. Once all cameras are queried for an image, all cameras receive the command to end image acquisition.

It is important to note that because of the manner in which the cameras are queried for an image, image acquisition is not truly simultaneous. This time difference is negligible, however, and has shown to be a better option than a true parallel computing process. Attempts to create a script with parallel threads or multiprocesses resulted in corrupted images and/or problems with timing.

Image saving occurs after all images are acquired as saving is the most time-consuming process in the script; waiting to save allows the script to acquire images with as little delay between rounds as possible. All images are saved as PNGs in the appropriate camera folder with the naming scheme "camXImage_Y". Further, all images are rotated 270 degrees using the Python Image Library (PIL), as the cameras in the specified equipment setup are turned on their sides. This rotation allows the final saved image to appear normal and upright to the user.

### 2.4 Performance

On the machine used during development, the average largest difference between the timestamps of images of the same round was 26ms in mono8 mode. In continuous mode using mono8 and acquiring 20 images, the program is able to capture image at an average rate of 9.38 fps with a standard deviation of 0.07.

## 3. User Guide

### 3.1 Installation Requirements

PT Grey Image Acquisition was written in Python 3.6, and uses the PySpin, datetime, time, os, and PIL Python libraries. The download and documentation for the PySpin library can be found on FLIR's website under "Spinnaker for Python". The script can be run from the command line or from a Python IDE; the script was developed and tested using Anaconda's Spyder. It is also recommended to use FLIR's FlyCapture2 Viewer to show live streams of individual cameras while manually focusing them.

### 3.2 Setup

The cameras used in this setup must be focused by hand. It is recommended to use FLIR's FlyCapture2 Viewer to view the camera livestream while adjusting. The user can select a camera by serial number to view the livestream, as shown in figure 2. Cameras are focused by turning the focus ring to infinity and then adjusting the iris ring until the desired brightness and focus is achieved. A diagram of the lens used in the setup can be found on the Computar website. Minor adjustments to the iris will need to be made when switching between subjects of different skin tones. If running in continuous mode, the packet delay must be increased; for our setup of six cameras, a packet delay of 10,000 works well. The FlyCapture2 Viewer should be closed completely before running the script.

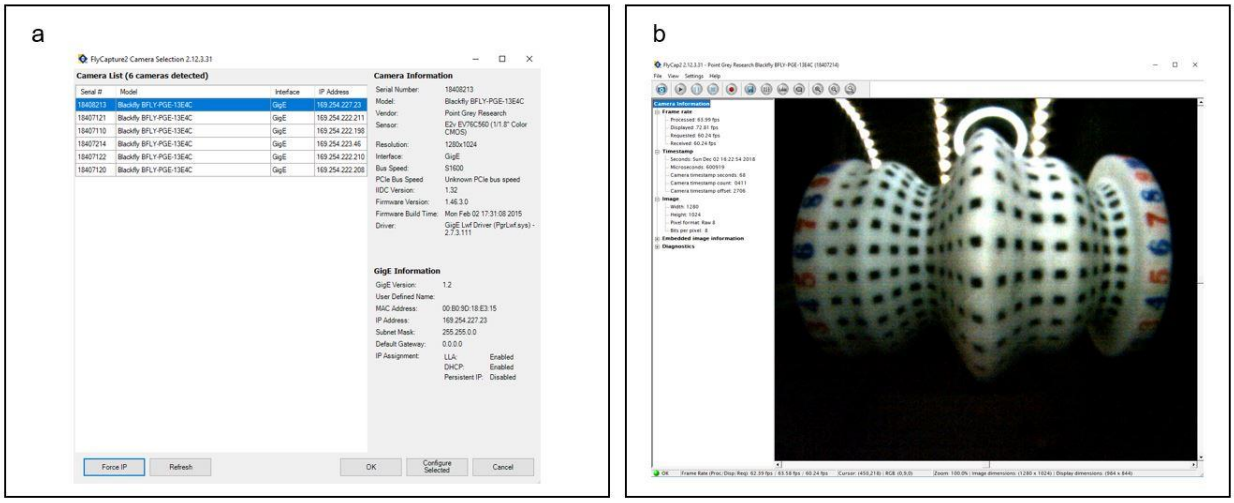


*Figure 2: FlyCapture2 during manual camera adjustments*

## 3.3 User Input

The user will be taken through several prompts to apply settings for the camera run; an example of user-script interactions is shown in figure 3. If the user provides an invalid input, the script terminates and must be re-run. Only manual mode requires user interaction during image acquisition. There are several print statements throughout the program to signify when major actions have been performed, such as acquiring a round of images or saving the images.

## 3.4 Image Acquisition

User interaction differs between the different acquisition modes. The use cases and typical interactions with each mode are listed below.

### 3.4.1 Manual Mode

Manual mode should be used for runs in which the user wishes to trigger the cameras manually. After the cameras are initialized, the user can specify when to take an image by pressing the ‘enter’ button. Once the images for that round are acquired, the user can press ‘enter’ once again to take the next round of images. The user signifies that the camera run is complete by entering ‘e’ in the prompt, at which point the images are saved and the script terminates. An example interaction in manual mode can be found in figure 3a.

### 3.4.2 Timed Mode

Timed mode should be used for runs in which the user wishes to automatically take pictures but with a delay of at least one second in between each image round. The user specifies the desired number of images as well as the time delay between each image. Once the cameras are initialized, there is a 5 second delay before a tone sounds and the script begins taking images. After acquiring a round of images, the script will “sleep” for the specified time delay. Once the desired number of images are taken, the tone sounds again, all images are saved, and the script

terminates. The current image round number will be printed to the screen as the program progresses. An example interaction in timed mode can be found in figure 3b.

### 3.4.3 Continuous Mode

Continuous mode should be used for runs in which the user wishes to automatically take pictures with no time delay between each image round. The user specifies the desired number of images. Once the cameras are initialized, there is a 5 second delay before a tone sounds and image acquisition begins. The current image round will be printed to the screen as the script progresses. Once the desired number of images are taken, the tone sounds again, all images are saved and the script terminates. An example interaction in continuous mode can be found in figure 3c.


*Figure 3: Example runs in manual mode (a), timed mode (b) and continuous mode (c)*

## 3.5 Saved Images

The images will be saved in the path specified during the script and camera setup. The default location path is "…\ Documents\Camera Runs", which was used for the specified equipment setup. If a new path location is entered for this setting, the validity of the specified path is

checked during the input phase of the script. The entire run will exist in a folder with the naming format “Camera Run Year-Month-Day Hour Minute Second”. Within this folder there will be several folders in the format “Camera Number”. Within each camera folder are the images acquired from this image for the run, with the naming format “camXImage_Y.png”. An example of the folders created for an image run are shown in figure 4. All images are saved in grayscale and rotated 270 degrees. Image naming is compatible with MultiDIC.

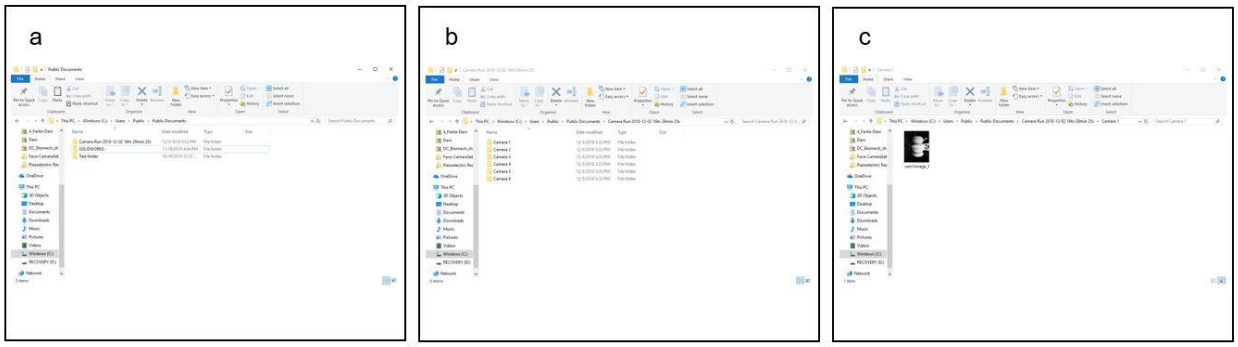


*Figure 4: Saving location folder (a), Camera Run folder (b) and an example Camera folder (c)*