MSc Summer Project Interim Presentation Script

July 12, 2019

1. For my summer project I am writing a program in Python to interface an EMCCD camera used to images ions in ion trap experiments

2. I will first briefly introduce what Ion traps and EMCCD cameras are and explain its basic principles of operation. Then I will list the main tasks that I have been set and finally I will report my current progress and my next steps for the project.

3. Ion traps. Ion traps, such as Penning or Paul traps are devices which are used to levitate small clouds of ions or even single ions in free space inside a vacuum chamber. In a typical experimental setup like the one shown in this schematic, lasers are used to manipulate the ions and an imaging system consisting of photomultiplier tubes and an EMCCD camera is used to detect the ion's fluorescence. The photograph on the right shows the new linear Paul trap which the ion trapping group is currently building.

4. EMCCD cameras. A Charged Coupled Device is a silicon based semiconductor chip which captures light and converts the photons into digital data in the form of electrons. The structure of an Electron multiplying CCD is identical to that of a conventional CCD, but it is much more sensitive which makes EMCCDs capable of single photon detection such as fluorescence from single ions. Hence EMCCD cameras are widely used in ion trap experiments

5. This diagram shows the schematics of an EMCCD camera. The image area is exposed light and the accumulated charge is then shifted down to storage area and then to the shift register where each pixel is read out one by one. The main feature which differentiate the EMCCD camera from a conventional CCD camera is the extension of the shift register with the gain register, which is what causes the significant improvement in low light detection. It is also possible to read out parts of the detector array by selecting a region of interest. This can be useful when the pixels outside the region of interest is background noise.

6. My tasks. My main objectives for the summer project are to write a program in Python 3 around an existing library which can interface our Andor iXon Ultra EMCCD camera. The camera also comes with a commercial software which is fine for general imaging. However, the Python program can be written such that it is tailored to the experimental requirements. Once the Python program is complete, I will use it to acquire images of ions and find a method to distinguish between a bright and dark ion. Finally, I will try to find the optimal parameter settings in order to obtain the best image with the shortest exposure time.

7. The program. I have written two separate programs: One to control the EMCCD camera and take pictures, and the other to load the images acquired by the camera. The program controls the camera by loading a dynamic link library which contains all the functions of the camera. To make the programs user-friendly, both programs are object oriented and have a graphical user interface created using the QT Creator software. The UI file from QT Creator was then converted into Python script with the PyQt5 module, hence there is no need to write the GUI in Python from scratch and once created, the GUI code can be left untouched.

8. The diagrams here shows the overall architecture of the two programs. The main code of the camera control program imports the code for the library and graphical user interface to form the complete program which can control and communicate with the EMCCD camera. like the camera control program, the code for the image loading program also imports its corresponding GUI code.When you run main program, images taken by the camera in the form of a .txt file can be loaded and displayed on the GUI.

9. This image shows the program I wrote for controlling the EMCCD camera. Other than the basic controls to turn the camera on and off, the program can also set the temperature of the camera cooler and choose acquisition mode between a single image with single scan or multiple images with kinetic scan. The program can also set the exposure time, the EMMCD gain and the region of the detector array being read out. Other controls include the triggering mode of the camera and controls of the camera shutter. Once all the required camera parameters has been set, the snap button can be pressed to acquire the image and saved using browse and save buttons.

10. This animation shows my image loading program in use, other than displaying the image, the vertical and horizontal projections of the image are also plotted. When the 'show max pixel' check box is checked, the brightest pixel in the image is marked with a cross after pressing the load button. For multiple images taken with the kinetic scan mode, the images can either be displayed all at once, or one by one using the drop-down menu.

11. After the programs were completed, the basic properties of the camera were tested with noise readings. The images were compared with that taken using the commercial software to ensure that my program was working as expected. The first tests were the affect of exposure time and EMCCD gain on the mean noise reading per pixel.

12. The graphs on the top shows the mean count per pixel against exposure time for images taken with the Python program and commercial software and the graphs on the bottom shows the mean count per pixel against EMCCD gain for images taken with the Python program and commercial software. As you can see, the mean count per pixel showed a linear dependency with both the exposure time and the EMCCD gain, and as expected, the data from the Python program and commercial software are almost identical which hence verifies that the Python program is operating as expected.

13. Next, images of a single ion were taken at different exposure times. The pixels of the camera have a dimension of 16 by 16 microns, and the imaging system has a magnification of 10. Since the diameter of the ion image is approximately 6 pixels, hence the diameter of the ion fluorescence is approximately 9.6 microns. It can be seen that at lower exposure times, the image becomes more noisy.

14. Here are images of single ions at even lower exposure times. It can be seen that at 0.1 second exposure, the ion can just about be detected by eye due to the significance of background noise. The time scales of the ion trap experiments are about 10 milliseconds, hence it is important to have a reliable method to detect the ions at such exposure times.

15. Next steps. I will now be finding a reliable method to distinguish a bright ion from a dark ion and investigate the minimum exposure required for an ion to be detected. The simplest method is to look at the signal distributions for a bright ion image and a dark ion image. The bright ion can no longer be distinguished from a dark ion if the two distributions have a large overlap.

16. After that I will investigate difference in image quality between when camera is triggered externally by the experiment and when start of the experiment and exposure are triggered by the camera. The EMCCD has a 'keep clean' cycle which clears the sensor to ensure that it is charge free before the next exposure. Triggering the camera externally may interrupt the keep clean cycle and therefore produce a noiser image. The camera gives off a 'fire signal' during exposure. You can take advantage of this 'fire signal' to trigger the start of the experiment at the end of a 'keep clean' cycle.

17. To sum up, two Python programs with a GUI were created to control the EMCCD camera and to display the images acquired by the camera. A method or algorithm is now required to reliably distinguish between bright and dark ions at short exposure time and finally the camera triggering mode which gives the highest signal to noise ratio will be investigated.