1. INTRODUCTION

Light microscopy has become an essential tool in biology and the biomedical sciences. A light microscope uses lenses to focus light on the specimen to magnify it, creating an image that can be used to observe many biological processes, such as the uptake of food, cell division and cell movement. The low cost of light microscopes makes them useful in many different areas, such as education and medicine. [1]

However, one of the biggest steps in light microscopy is the idea of fluorescence microscopy. Fluorescence microscopy is one of the popular methods for studying the dynamic behavior of live cell imaging. Fluorescence microscopy can be used to stain different molecules with different colors, allowing multiple types of molecules to be tracked simultaneously. However, cells are susceptible to phototoxicity, especially with short wavelength light. Among the various types of fluorescence microscopy, light sheet fluorescence microscopy (LSFM) is one of the best methods for good image quality, fast 3D imaging and low phototoxicity. [2]

Nowadays there are many ways to establish communication between a computer and devices like microscopes. One of the methods to perform this type of communication is to use Internet of Things (IoT) messaging protocol such as MQTT (Message Queuing Telemetry Transport). MQTT allows automation, and there is a library in Python called "Paho" that implements a client class to add MQTT support. It can be used to make connections and automate devices such as refrigerators, thermometers, microscopes, and many other devices.

With MQTT, the microscope can be flexibly configured by communicating with it, for example, by specifying the type of fluorescent dye, whose required settings are received, changed, and sent back to the microscope. Furthermore, light-sheet microscope that was used in this internship project always did two channels, so it always produced two image datasets from two views after running the experiment. For that it was necessary to delete one file using Python for the sake of space used in the hard drive. These two steps represent a small step towards automating the microscope configurations.

Due to some factors, such as changes of refractive indices in medium and sample as well as the imperfection of optical components like lenses, the image captured by the microscope appears a bit blurred and needs further processing to get a better image. Thus, by further processing the output image, interesting information can be retrieved, and multiple data sources can be integrated for knowledge extraction, which can help scientists such as bioinformaticians and physicians to further develop diagnoses and therapies.

In this report, first a background about fluorescence and LSFM will be presented. Then the methods to control the microscope via the use Python will be explained. After that, different possibilities of image processing techniques for LSFM will be discussed.

2. Light sheet fluorescence microscopy (LSFM)

2.1 Theoretical background

2.1.1 Fluorescence:

Fluorescence is a phenomenon in which light is emitted by a molecule that has absorbed light or electromagnetic radiation from another source. Upon absorption, the system is excited by high-energy light, causing the electrons within the molecule to transition from the ground state to an excited state. Once this state is reached, the electrons relax back to the ground

state after a fluorescence lifetime and release their stored energy in the form of an emitted photon. Normally, the emitted light has a lower energy than the absorbed radiation[3].

2.1.2 Fluorescence Microscope:

The "fluorescence microscope" refers to any microscope that uses fluorescence to generate an image. One of the components of the fluorescence microscope is the light source. The light coming from the light source is filtered through an excitation filter to produce an excitatory light of the desired wavelength. This excitatory light is then reflected by a dichroic mirror onto the sample preventing it from reaching the detector. The sample is thus excited by the fluorescent light. The emitted fluorescent light then passes through the emission filter, which blocks extraneous excitation light and allows only the emitted fluorescent light to reflect through the dichroic mirror to the detector. [4]

2.1.3 Light Sheet Fluorescence Microscope:

Light sheet fluorescence microscopy (LSFM) is an effective imaging approach that unifies the speed of wide-field imaging with optical sectioning and low photobleaching. In contrast to epifluorescence microscopy, the sample is illuminated from the side, generally 90° from the direction of observation, so that the excitation light is placed only where it is essential which reduces the photo-damage and stresses induced on a living specimen [21]. In addition, the imaging speed is orders of magnitude faster and the good sectioning capability reduces background noise, producing images with better contrast than other microscopes such as the confocal microscope [5].

2.1.4 QUVI SPIM:

A dual-view light sheet microscope (QuVi, Bruker, Germany) was used in this project. The system enables fast and simple transition between living and cleared tissue imaging. It can be used for high-throughput long-term imaging of 3D cell culture models such as organoids[6]. The used light sheet microscope uses JSON (JavaScript Object Notation) for sending data over the server and for storing messages. It can communicate with Server via MQTT (Message Queuing Telemetry Transport). Upon completion of an experiment, the images captured by the microscope are stored in the form of stacks (see 2a vi).

2.1.5 MQTT:

MQTT is a simple network protocol which can transport messages between devices. It is designed to connect to multiple devices where these devices can communicate with each other by subscribing and publishing various messages to one another. The Paho is a Python client that provides client class with support for MQTT[7]. It has simple helper functions for receiving and sending the messages during MQTT communications. The MQTT protocol defines two types of network entities: a message broker/ server and several clients. An MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. An MQTT client is any device that runs an MQTT library and connects to a MQTT broker over a network. During these communications the messages should be sent under specific topics. In MQTT, the word "Topic" refers to a string that the broker uses to filter messages for each connected client[8]. So different types of topics are

required according to the messages to be sent to the various types of clients. For example, in our microscope, the "gui" topic is used to send general settings messages, the "gui/datahub" topic is used for camera settings messages, and "gui/directory" is used to send the file directory messages.

2.1.6 HDF5:

The used light sheet microscope saves the data in HDF5 (Hierarchical Data Format Version 5) format. The HDF5 is an open-source file format that supports large, complex, heterogeneous data. HDF5 uses a "file directory" like structure that allows the user to organize data within the file in many different structured ways. [9]

2.2 METHODS:

Instead of using the GUI software LuxController for configuration, Python was used to control the microscope with the aim of configuring it and making it ready for the experiment. The library in Python called “Paho” that implements a client class to add MQTT support was used to send and receive different types of messages to the microscope on various topics.

First, messages were sent encoded in JSON format to get current information from the microscope. For example, messages were sent to see the current stage positions from the microscope for setting it later. Messages to set the current optical configurations and illuminations were also received accordingly. Then, the same was done for filter wheels and exposure time. These all messages were sent under the same "gui" topic.

Also, the region of interest for the camera was set and this was sent under another topic "gui/datahub". And for creating, changing, and selecting the file directory, the messages were sent under the topic "gui/directory".

After sending all these messages in their specific topics to the QuVi microscope via MQTT to receive the messages with the configurations, these received messages were packed into a list in Python. The listed messages with the configurations were then filtered and edited by some functions written in Python. After these messages were processed, they were sent back to the microscope to set it up, and the microscope was prepared to perform the experiment.

At the end of the experiment, the image datasets were saved in the HDF5 file format which can be processed further using Python.

2.3 Results:

As shown in figure 1, the send_get_stages function shown in number 1 had a dictionary type message which was used to receive the message from the microscope which is shown at number 2. Number 3 shows a Python script which was used to modify this received message for changing the values and at number 4 one can see the message in JSON format which had a modified values of x-

axis sent back for setting the microscope. Different topics such as “01C3BAF8/gui” were used to send the message and topics such as “01C3BAF8/# “was used to subscribe.

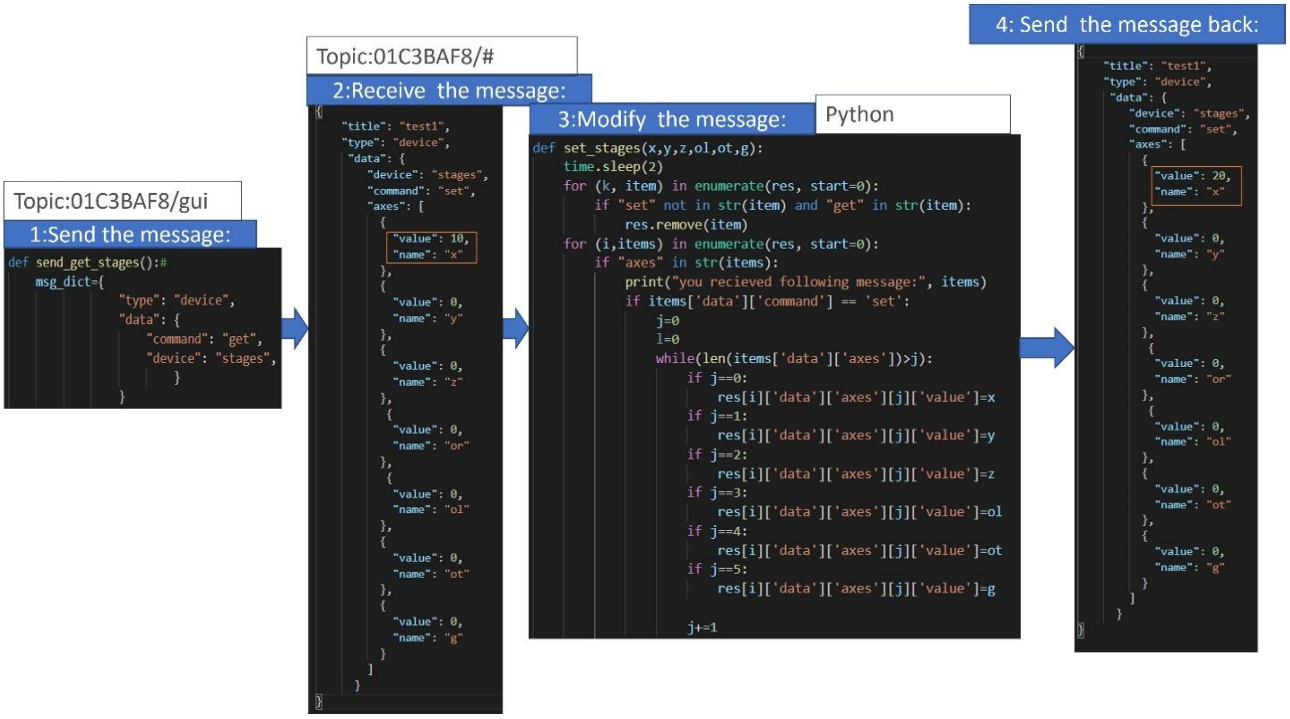


Figure 1. An example showing a workflow for MQTT

Various features were implemented to extend the goal of configuring the microscope. One of them was the automatic deletion of files originating from the microscope by default, if necessary. For example, if you only need the image of one channel, the other channel can be ignored, but the QuVi SPIM microscope is not configured to do this and therefore saves both files from long and short channel, so a script was written to automatically delete these files, if necessary, at the end of the experiment. The figure below shows another idea to extend and make settings easier for selecting different types of fluorophores like “DAPI”, “FITC”, etc.

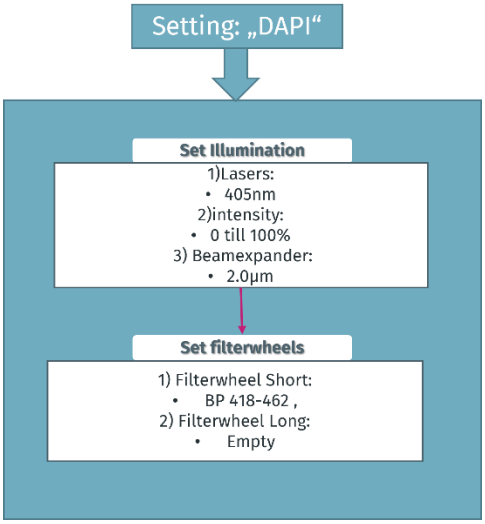


Figure 2. An example of combining two functions for adjusting the settings for imaging fluorophore “DAPI”.

When setting up such fluorophores, it was necessary to choose the correct illumination settings and filter wheel settings to match the characteristic excitation and emission spectra of the fluorophores used, as shown in the figure 2 for "DAPI".

2.4 Discussion:

The results show that scientific instruments such as the QuVi SPIM microscope can be configured and automated by implementing MQTT in Python. In addition, many advanced ideas such as fluorophore configuration can be used to facilitate microscope setup and experimentation for scientific work such as biomedical research. More research is needed for even better automation to solve specific tasks.

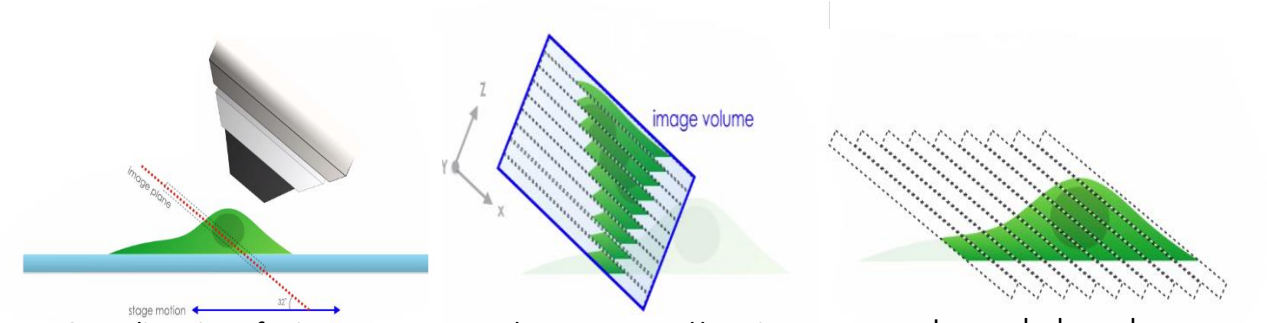
3. Image Processing:

In LSFM, the required detail and contrast of the image depends on the refractive index, optical properties, and other sample-dependent parameters, so the image produced by the microscope is somewhat blurred. To improve the quality and get a better image, the image should be further processed to extract more useful insights from it.

3.1 Theoretical background:

3.1.1 Deskewing:

Deskewing is the process of straightening an image that has been scanned or written crookedly — that is an image that is slanting too far in one direction, or one that is misaligned. In LSFM, the x and y coordinate directions of the individual slices are not perpendicular to the z-axis of the stage due to the oblique light sheet angle (see Figure 3a), and the acquired image stack appears skewed (or sheared), as shown in Figure 3b. We need to shift the individual z-stacked images so that the sample is no longer distorted, using affine transformations so that the image is deskewed and the undistorted image is revealed, as shown in Figure 3c.



a. Scan direction of microscope towards z-axis

b. Image saved by microscope

c. Image deskewed

Figure 3. Processes for describing deskewed and skewed images. Source: Adapted from [11]

3.1.2 Translation

A translation is a geometric transformation that moves every point on a figure the same distance in every direction. This requires a translation / shift vector, a type of transformation that moves a figure in the coordinate plane from one place to another. For example, in Figure 4, the first object in purple is translated along the x, y, and z axes by a translation vector so that it moves in the same shape and space as the second object in brown. For the light sheet microscope used, a translation must be performed so that the two images from

the dual view lie in one coordinate plane and can be fused together later on.

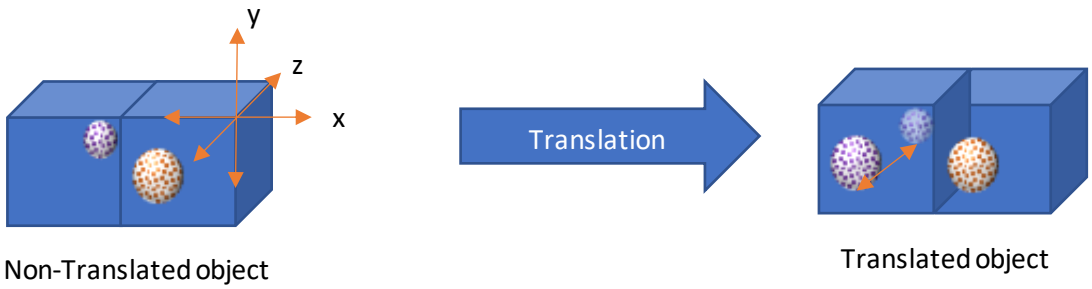


Figure 4. images describing translation

3.1.3 Deconvolution:

In fluorescence microscopy, the acquired image is always a blurred representation of the actual object under the microscope as shown in the figure 5a.

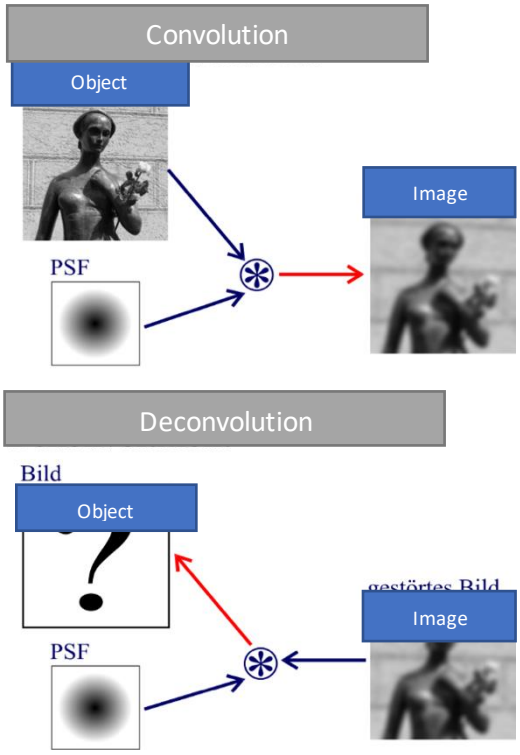


Figure 5. Images showing about convolution and deconvolution. Source: Adapted from [14]

This blurring is described by the so-called point-spread function (PSF), which can be seen using the example of the center movement along the z-axis of a diffraction limited source, such as fluorescent beads. The PSF describes what a single point in the object looks like in the image. The PSF is a valid measure of the quality of an optical system as it shows how points in the image are out of focus. The shape of the PSF depends on the type of microscope and the microscopy parameters used. If the PSF is known, it can be used to unfold the captured microscope image to bring it closer to the true object. The goal of deconvolution is to find the true object using knowledge of the PSF and the captured image, as shown under deconvolution in the figure 5.

Deconvolution reverses the imaging convolution in an iterative fashion: a model of the object is made and iteratively improved upon by convolving it with the PSF and comparing the result to the actual image. At the end of the process, the model of the object is a better approximation of the true object, with improved resolution and signal-to-noise ratio relative to the acquired image.[12]

3.1.4 Fusion:

Image fusion combines two or more registered images of the same object into a single image that is easier to interpret than the two originals, resulting in new images with better isotropic resolution[13]. As shown in the figure 6, the images from view A and view B of a microscope are fused together, in which regions are selected from one image or the other based on local quality. One another method to do fusion is to fuse two images in Discrete Cosine Transform (DCT) domain by picking coefficients with maximal magnitude. [15]

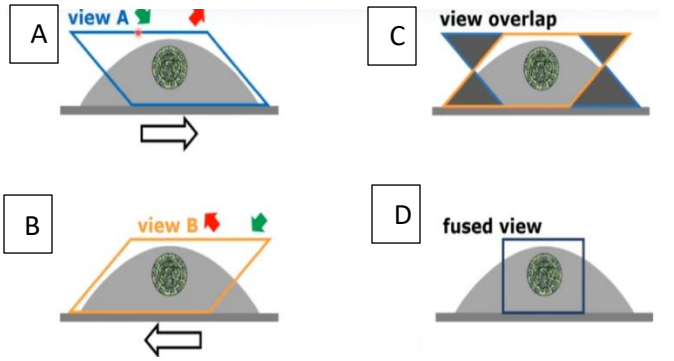


Figure 6. images describing steps of fusion. Source: Adapted from [20].

3.1.5 Napari:

Napari is a fast, interactive, multidimensional image viewer for Python. It is designed for browsing, annotating, and analyzing large multidimensional images [19]. It is used in this project to visualize images.

3.2 Methods

Two different packages were used for image processing with Python. One of these was pyCUDAdecon, which is being developed by Talley Lambert [18], and another package called dexp, which is being developed by Royer Lab [15]. These two packages were used to perform similar image processing tasks such as deskewing, transforming and deconvolution of LSFM images with Python. The default parameters for image and PSF were used for the Lucy Richardson deconvolution function available in both packages.

In the pyCUDAdecon package, an integrated parameter for deskewing and rotating was available within the deconvolution function. These parameters were assigned a specific angular degree, for example 45° for the used microscope. Other functions such as affine transformation functions to perform translation, rotation and deskewing were also tested, all of which are available in the pyCUDAdecon package.

Then the dexp package was evaluated, which contains further new functions. Images were cropped and, if necessary, provided with a scale bar after processing using the functions available in dexp. Deskewing, fusion and many other image registrations like translation were done to make the image clearer and more corrected. This Python package automatically used different backends depending on GPU and RAM availability.

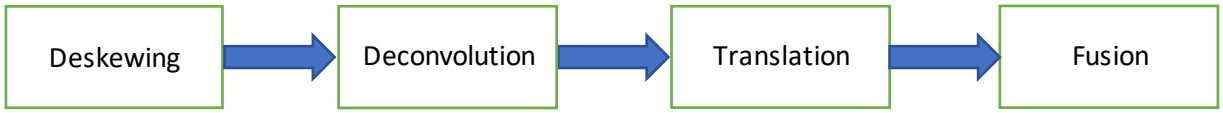


Figure 7. Basic workflow of an image processing

In this project, the processing workflow depicted in Fig. 7 was applied. First, deskewing was done, then deconvolution, then translation and finally fusion was done using both packages one after another. At the end of the image processing, two images were fused together to obtain an optimal result. After the image editing, napari was used to visualize the images, which is also available in these two packages.

3.3 Results and Discussion:

The Python package pyCUDAdecon could only use CUDA-based NVIDIA GPUs in its functions. While dexp can use NumPy (RAM-based) backend and CUPY (CUDA-based backend with 12 GB graphics card) with many other new features. Dexp has more features for image registration, but it is currently still under development and therefore has a limited functionality due to poor documentation and unclear errors. Both packages have a napari-based visualization function and both packages have command line interface. The dexp package, on the other hand, has many sub-module functions.

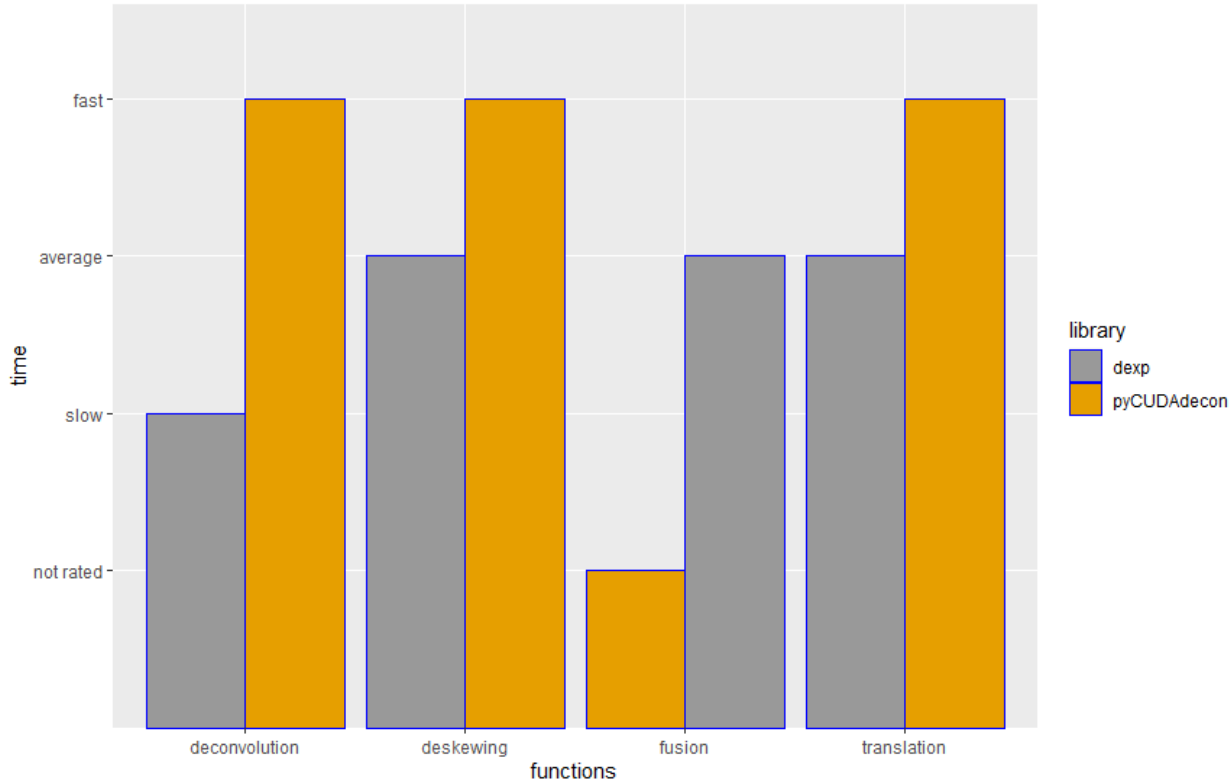


Figure 8. Bar chart showing evaluation of different image processing functions based on image processing time.

The evaluation in this line chart (as shown in figure 8) was based on processing time. The pyCUDAdecon package doesn't have features like fusion, hence it was not rated. pyCUDAdecon is faster than dexp as

pyCUDAdecon uses only GPU based backend, but some failures due to default limitation GPU usage in an operating system may occur sometimes. Dexp has a good translation feature that took 10 minutes to complete. The deconvolution function of dexp uses RAM based backend and takes about 55 minutes for one run. The fusion gives an average result that also takes about 10 minutes using GPU based backend. Using functions with the GPU backend as in pyCUDAdecon would reduce the processing time, but for the functions with RAM-based backend as in the dexp package, the time required would become enormous.

4. Summary and Outlook:

The aim of this internship project is to test the use of MQTT to control the microscope using Python programming language and then do further image processing to improve these images for scientific purpose. The main thematic focus of this work was to develop an API in python that could control the microscope using MQTT.

The first part of the project dealt with QuVi SPIM and MQTT. The QuVi SPIM microscope which is an inverted light sheet fluorescence microscope was used and MQTT which has a library available in Python programming language was implemented to control the microscope. After developing certain functions such as function for changing the stage positions, setting the filter wheels and other tasks for controlling the microscope, also some additional functions were implemented in order to extend the ideas for making the work easier like selection of fluorophores and deleting unnecessary files that are given out by the microscope automatically after the end of the experiment.

The second part of this project describes about many images processing functions like deskewing, translation, deconvolution, etc. from two different python packages (pyCUDAdecon and dexp). Both of these packages were used and tested using an image of MCF10a organoids which was provided. Benchmarking was done according to time and quality of output after the images were processed.

This project dealt with the use of an informatics tool such as Python to control, configure and automate a microscope via MQTT, which can contribute to software development for such scientific instruments for biomedical research. It also dealt with the further processing of the images obtained after performing the experiment with the microscope, which can be further used in scientific research. And the evaluation of image processing libraries like pyCUDAdecon and dexp in Python gave a better overview for the development of these libraries in the future.

5. References

[1] M. J. Sanderson, I. Smith, I. Parker, and M. D. Bootman, "Fluorescence Microscopy," *Cold Spring Harbor protocols*, vol. 2014, no. 10, Oct. 2014, doi: 10.1101/PDB.TOP071795.

[2] P. A. Santi, "Light Sheet Fluorescence Microscopy: A Review," *http://dx.doi.org/10.1369/0022155410394857*, vol. 59, no. 2, pp. 129–138, Feb. 2011, doi: 10.1369/0022155410394857.

[3] "Fluorescence is the emission of light - AAVOS International." https://aavos.eu/glossary/fluorescence/ (accessed Sep. 27, 2021).

[4] "Fluorescence microscopy a simple overview | ONI." https://oni.bio/nanoimager/super-resolution-microscopy/fluorescence-microscopy-overview/ (accessed Sep. 27, 2021).

[5] "Single Plane Illumination Techniques (Light-sheet Microscopy) - Institute for Molecular Bioscience - University of Queensland." https://imb.uq.edu.au/research/facilities/microscopy/training-manuals/microscopy-online-resources/image-capture/single-plane-illumination-techniques-light-sheet-microscopy (accessed Sep. 27, 2021).

[6] "QuVi SPIM | Bruker." https://www.bruker.com/en/products-and-solutions/fluorescence-microscopy/light-sheet-microscopes/quvi-spim.html (accessed Sep. 27, 2021).

[7] "Beginners Guide To The MQTT Protocol." http://www.steves-internet-guide.com/mqtt/ (accessed Sep. 27, 2021).

[8] "MQTT Topics & Best Practices - MQTT Essentials: Part 5." https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/ (accessed Sep. 27, 2021).

[9] "Hierarchical Data Formats - What is HDF5? | NSF NEON | Open Data to Understand our Ecosystems." https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5 (accessed Oct. 08, 2021).

[10] "Lattice_Lightsheet_Deskew_Deconv/00_Lattice_Light_Sheet_Deskew.ipynb at master · VolkerH/Lattice_Lightsheet_Deskew_Deconv · GitHub." https://github.com/VolkerH/Lattice_Lightsheet_Deskew_Deconv/blob/master/examples/00_Lattice_Light_Sheet_Deskew.ipynb (accessed Oct. 06, 2021).

[11] "Deskewing Lattice Light Sheet Data :: HMS CBMF." https://cbmf.hms.harvard.edu/avada_faq/deskewing/ (accessed Oct. 06, 2021).

[12] "Point Spread Function (PSF) | Scientific Volume Imaging." https://svi.nl/Point-Spread-Function-(PSF) (accessed Oct. 06, 2021).

[13] A. de Juan, A. Gowen, L. Duponchel, and C. Ruckebusch, "Image Fusion," *Data Handling in Science and Technology*, vol. 31, pp. 311–344, Jan. 2019, doi: 10.1016/B978-0-444-63984-4.00011-9.

[14] "Dekonvolution – Wikipedia." https://de.wikipedia.org/wiki/Dekonvolution (accessed Oct. 22, 2021).

[15] "GitHub - royerlab/dexp: Dataset EXploration & Processing." https://github.com/royerlab/dexp (accessed Oct. 22, 2021).

[16] "Deskewing Lattice Light Sheet Data :: HMS CBMF." https://cbmf.hms.harvard.edu/avada_faq/deskewing/ (accessed Oct. 22, 2021).

[17] C. Saranya and S. Shoba, "Comparison of Image Fusion Technique by Various Transform based Methods", Accessed: Oct. 22, 2021. [Online]. Available: www.ijert.org

[18] "Home - pycudadecon." https://www.talleylambert.com/pycudadecon/ (accessed Oct. 21, 2021).

[19] "Home - napari." https://napari.org/ (accessed Oct. 22, 2021).

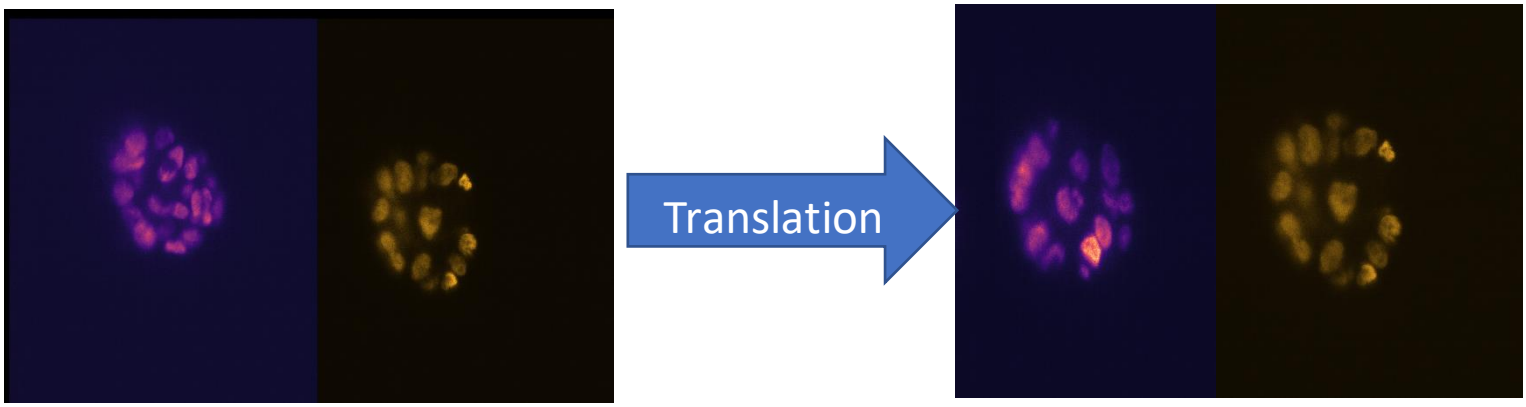
[20] "Advances in light sheet microscopy a universal tool for 3d bioimaging | Bruker." https://www.bruker.com/en/news-and-events/webinars/2020/advances-in-light-sheet-microscopy-a-universal-tool-for-3d-bioimaging.html (accessed Oct. 22, 2021).

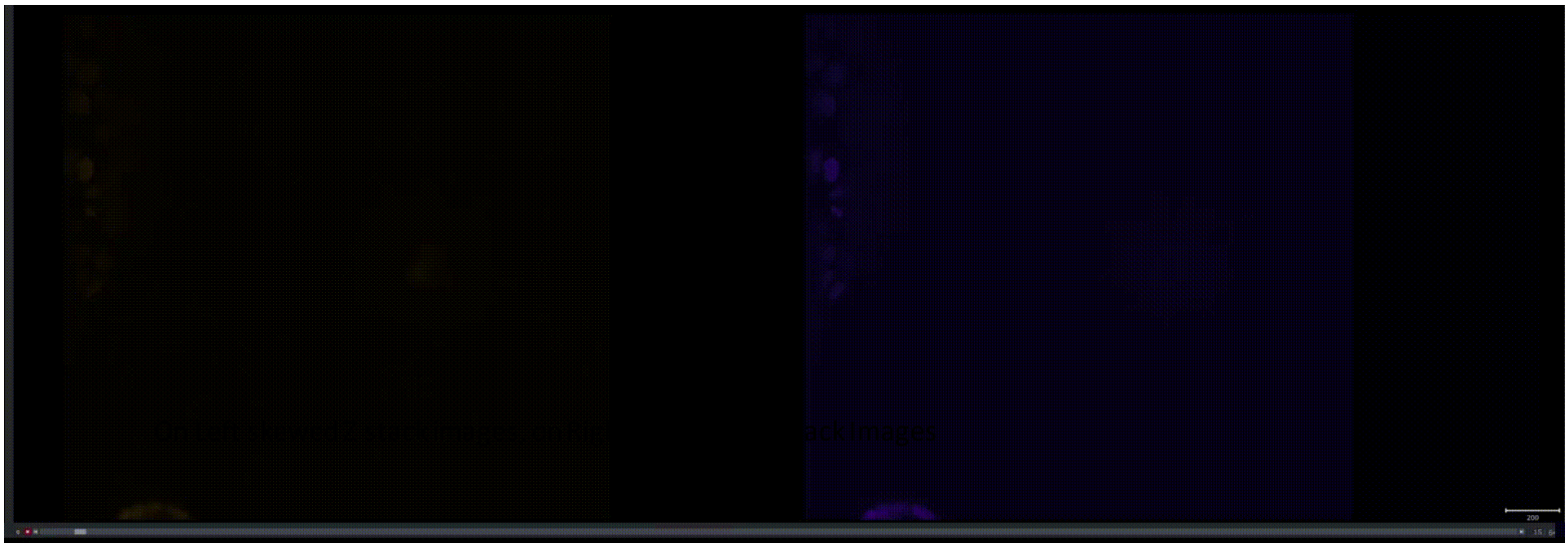
[21] "Light Sheet Microscopy: Transforming 3D Fluorescence Imaging | Features | May/Jun 2020 | BioPhotonics." https://www.photonics.com/Articles/Light_Sheet_Microscopy_Transforming_3D/a65704 (accessed Nov. 08, 2021).

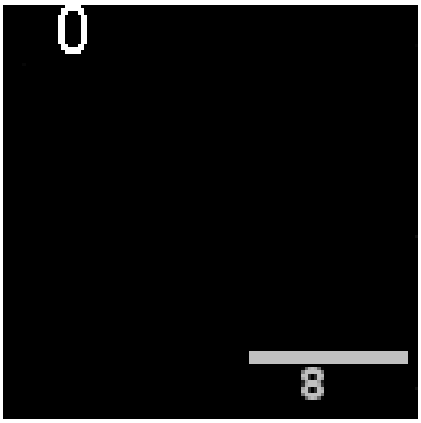
6. Acknowledgment:

I would like to thank the following people, without whom I would not have been able to complete this internship, and without whom I would not have made it through my internship project! The Intelligent Imaging team at Berlin Institute of Health, especially to my supervisor Alexander Sudy, whose insight and knowledge into the subject matter steered me through this internship. And special thanks to the group leader and professor Christian Conrad, who gave me such opportunity to meet such a motivating and wonderful group and made it possible for me to engage in such a nice internship project. And thanks also to many of group members who were constantly encouraging me throughout the internship. And my biggest thanks to my family and friends for all the support you have shown me during this internship.

9) Appendix:







An example of PSF