Quarto Template Report

Hailey Knolton Ruby Peterman

2024-07-11

Table of contents

1 Quarto Report Template

This Quarto project is a template that you can use for developing your own final report. It suggest a potential structure along with some demonstrations of the advantages of using Quarto for technical documentation (e.g., in-line data processing and plot generation, references, clean formatting, figures, etc.)

You can find the source code for the whole project in the accompanying Github repo.

If you have questions or problems with getting it up and running on your own personal machine, the best way to get help is to open a new issue on the Github repo.

File Structure

The suggested structure for this repository is designed to maintain clarity and keep related files together.

```
.
  _book
  _environment.yml
  _quarto.yml
  _site
  ieee.csl
  index.qmd
  references.bib
  sections
      abstract
          abstract.qmd
      acknowledgements
          acknowledgements.qmd
      conclusion
          conclusion.qmd
      introduction
          images
          introduction.qmd
      references
      results
```

```
        data
            test_data.mat
        images
            test_data_plot.png
        results.qmd
styles.css
```

Tree diagram created using tree.nathanfriend.io.

2 Abstract

This paper presents the continuation of the design of an open source, inexpensive, and modular In Vivo Imaging System (IVIS) from the Harvey Mudd Biophotonics Lab during the summer of 2024. The authors contributions include the implementation of fluorescence imaging and laser speckle imaging on an improved physical system design. The system was created using widely available materials and tools including acrylic, a laser cutter, a 3D printer, and a Raspberry Pi computer. The experiments done to verify fluorescence imaging resulted in the expected relationship between fluorescence and fluorescent dye concentrations with a set exposure time of 0.1 seconds. The experiments done to implement and verify laser speckle imaging have assisted in the development of its abilities as an IVIS imaging method, but need further work to provide accurate results. The results from the experiments demonstrate the improved capabilities of the new IVIS system and pathways for future developments.

3 Introduction

3.1 1.1 In Vivo Imaging Systems

In Vivo Imaging Systems (IVIS) are optical imaging devices used in scientific research to create 2D and 3D representations of biological organisms and processes non-invasively. These systems use advanced imaging techniques, such as bioluminescence and fluorescence imaging, to visualize and track various biological activities within an organism over time. This approach to optical imaging can assist in drug development, understanding disease behaviors, or other biological processes in their natural context [Refaat]. Most IVIS Imaging methods implement optics in order to extract information about a subject based on the optical properties of the subject and the technique being applied. Many commercial IVIS systems can provide additional capabilities such as X-Ray, temperature control, computed tomography (CT), or accessories [KU, Revvity]

While IVIS systems are able to create detailed images over a broad range of applications, they can be limiting due to their inaccessibility. IVIS systems tend to cost upwards of $100,000, which may bar smaller or less-funded research institutions from purchasing their own system [bostonind]. Renting the use of an IVIS system is typically in the hundreds of dollars range as well, which further restricts the accessibility of in vivo imaging [OSU]. In addition to being high-cost, commercial IVIS systems are also restricted to the imaging applications they've been developed for with little room for customization or modularity [Source?].

The goal of the OpenIVIS project is to create a low-cost, open source and modular version of an IVIS system for biological imaging. Free open-source software (FOSS), free open-source hardware (FOSH) and the increased accessibility of rapid prototyping techniques, such as 3-dimensional (3D) printing, would allow for any institution to implement a version of this system in their research. An IVIS system with a modular design would also permit users to replace the imaging techniques used in order to best advance their work. Additionally, this would pave the way for implementation of in-vivo imaging techniques not currently available in most commercial systems such as Laser Speckle Contrast Imaging (LSCI). SHOULD maybe reference previous work on this, especially the CSM/HMC paper.

3.2 1.2 Flourescence Imaging

One of the most common capabilities of IVIS systems is fluorescence imaging. Fluorescence is a highly sensitive analytical tool that is used to measure extremely low concentrations of a compound in a solution [Williams]. The Jablonski diagram shown in Fig. # depicts the fluorescent process. When light is absorbed by a compound, molecules of that compound will become excited and raise to a higher energy level. Fluorescent compounds usually contain conjugated double bonds, where a certain number of electrons have greater mobility than the other electrons in the molecule [Williams]. This greater mobility allows for more molecules to become excited when the light is absorbed. When these molecules return to their ground state, some of the energy is emitted as fluorescence.

The energy that makes up light are called photons. Photons that absorb and excite molecules hold a certain amount of energy that determines their wavelength, or color [Saleh]. When a molecule emits a photon as it returns to ground state, the energy in the photon that is emitted is less than in the photon that was excited. This means that the resulting photon will have a longer wavelength and a different color [Saleh].

In order to image fluorescence, the absorbed and emitted photons of light must be controlled based on their wavelength spectrums. The excitation wavelength spectrum and the emission wavelength spectrum can often overlap, allowing the camera to capture photons of both wavelengths. An example of these spectrums is shown in Fig. #. In order to see the fluorescence of a compound, only the emitted light must be captured by the camera. Implementing an optical filter can help to control what wavelengths are captured. Optical filters allow for wavelengths of a certain range to be the only wavelengths detected by a camera by filtering out other wavelengths Fluorescence imaging has a variety of applications including medical imaging, environmental monitoring, and biological research. A common application of fluorescence is to non-invasively analyze biological molecules in vivo. Most IVIS systems use fluorescence for this purpose, and OpenIVIS will also demonstrate this fluorescence capability.

3.3 1.3 Laser Speckle Contrast Imaging

Laser Speckle Contrast Imaging (LSCI) is an optical imaging technique used to track movement, such as blood flow, by visualizing blur. When a diffuse object is illuminated with coherent light, it produces scattered light waves which can be visualized as a random interference pattern called a speckle image [Boas]. In order to determine the size of the speckles in the speckle pattern, autocorrelation can be applied to the image. Autocorrelation compares the intensity of the speckle pattern at two different points by multiplying values across the entire image. The autocorrelation can also be found by taking the fourier transform of the images intensity distribution [Wikipedia]. Eq. 1 shows a speckle pattern's autocorrelation calculation by taking the Fast Fourier Transform of the image's intensity I(x,y). The transform is then multiplied by its complex conjugate, noted by the asterisk, in the fourier domain to obtain

Fcc in Eq. 2. Finally, the inverse fourier transform returns the calculation to the spatial domain, resulting in the autocorrelation FA as shown in Eq. 3. Given the autocorrelation of a speckle image, the size of the speckle in pixels can be determined by finding the width of the autcorrelation's peak at half of its maximum, often referred to as the full width half max (FWHM). Most speckle images have a speckle size of one to two pixels.

$$F_{fft}(f) = FFT\{I(x,y)\} Eq.1$$

Fcc(f) =Ffft(f)Ffft*(f) Eq. 2 FA(t) = IFFT{Fcc(f)} Eq. 3

Static speckle images have high contrast patterns but when movement is imaged, the fluctuations in intensity can cause the contrast between neighboring speckles to decrease. The speckle contrast, K, can be derived as the standard deviation of pixel intensity over the mean pixel intensity, as shown in Eq. 4. Moving objects, such as blood flowing in a vein, causes the speckle pattern to shift, or decorrelate [Briers]. When this occurs, the intensity of neighboring speckles will become more similar, decreasing their contrast value.

K = Eq. 4

In order to compute the contrast of a full speckle image, a small window is applied to the original speckle pattern, typically 5x5 or 7x7 pixels large. This window, often referred to as a "neighborhood" is run over the entire image, computing the speckle contrast K for the intensities at each location before shifting over by 1 pixel at a time. The image is then reconstructed using the respective K values in order to produce a laser speckle contrast image. A speckle image is shown in Fig. #a, and its corresponding LSCI reproduction is shown in Fig. #b. By comparing the contrast patterns between different speckle images over time, the velocity of the movement being imaged can be determined.

4 Methods

4.1 2.1 Physical Construction

4.2 2.2 Electrical Components and Software

4.3 2.3 Experiments

4.3.1 2.3.1 Verification

4.3.2 2.3.2 FITC

4.3.3 2.3.3 LSCI

5 Results

5.1 3.1 System Verification and Setup

5.2 3.2 Flourescence Imaging

5.3 3.3 Laser Speckle Contrast Imaging

The advantage to using Quarto as compared to another method of documenting your work is that you can include executable Python code inline with your project. This means that you can provide your data and have your figures re-render without needing to execute separately and then include them into the project.

5.4 Demonstration

5.4.1 Create sample dataset

For example, let's consider a toy example. First, we'll create and save a data set. In your case, you'll likely skip this step since you'll likely have a dataset from elsewhere to use.

First, let's create a Python code block to create some random data.

```
import numpy as np

N = 100 # Set number of data points

x = np.linspace(0, 10, N) # Create x vector
y = 5*x + np.random.rand(N)      # Generate linear data with simulated noise
```

Next, let's save this data in our `/data/` subfolder. You can choose various formats, but one convenient option is to save the data in a `.mat` file. This way it can be opened and processed in Matlab as well as Python. It also allows you to nicely structure the data and will make the loading and plotting code we develop Matlab compatible.

```
import scipy.io as sio  # Load Scipy IO module which has the function to save and load .mat i

data_to_save = {'x': x, 'y': y} # Create dictionary to story the data
sio.savemat('data/test_data.mat', data_to_save)  # Save the data to a .mat file
```

5.4.2 Load and Plot Data

First let's load the data from the `.mat` file.

```
data_load = sio.loadmat('data/test_data.mat')
```

Then, let's plot it.

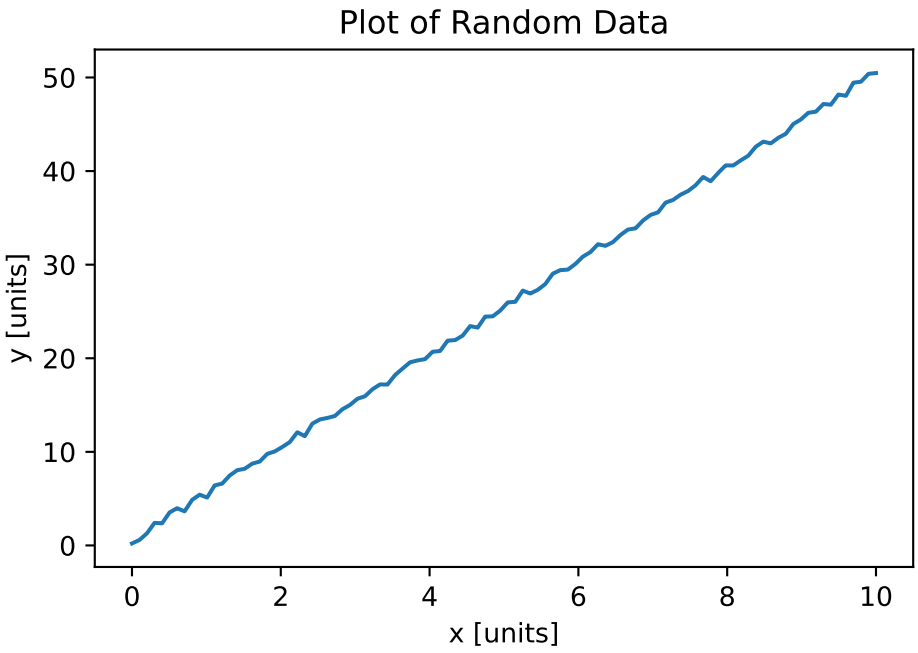
> **Warning**
>
> Note carefully that because of how `savemat()` works, we need to index into the **zeroth element** of the dictionary values in order to get to the data we want.

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.plot(data_load['x'][0],data_load['y'][0]) # Note that we are using the loaded data here ar
ax.set_xlabel('x [units]')
ax.set_ylabel('y [units]')
ax.set_title('Plot of Random Data')
```

```
Text(0.5, 1.0, 'Plot of Random Data')
```



5.4.3 Save and Plot Images of Figures

We can even go ahead and save the plots as images if we want to have the flexiblity to include them elsewhere.

```
fig.savefig('images/test_data_plot.png', dpi=300)
```

Then we can directly display as a Quarto figure as usual.

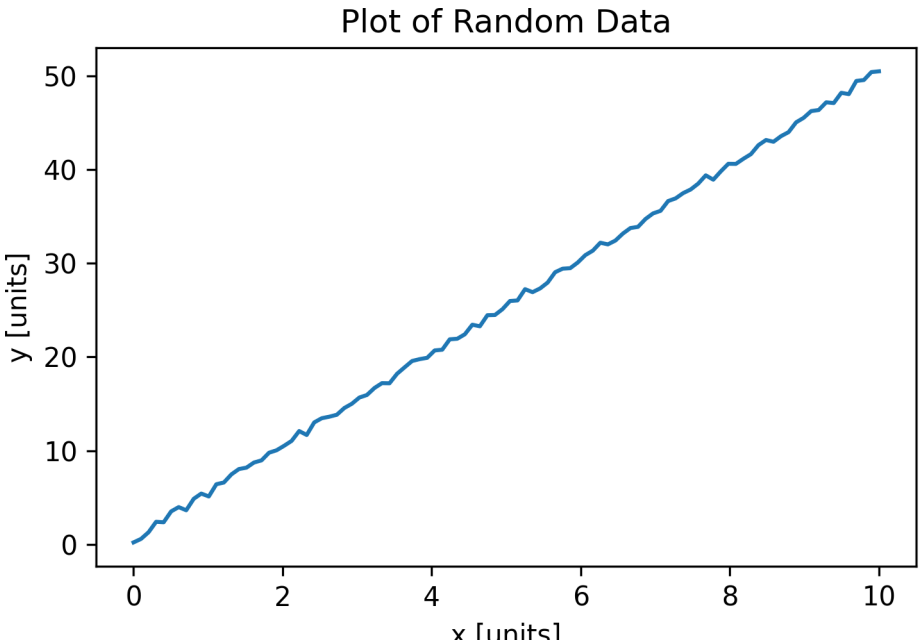


Figure 5.1: Displaying image of plotted random example data.

6 Conclusion

The conclusion summaries the main results of your paper. It generally mirrors the abstract, but in slightly more detail.

It is also common for the conclusion to include a discussion of the data, limitations of the work, and directions for future work.

7 Acknowledgments

The acknowledgements section should always recognize the funding sources which supported the work. This is particularly important for research funded by organizations like the National Science Foundation (NSF) or the National Institutes of Health (NIH). Acknowledging funding properly is very important, so make sure to check in with your PI if you have any questions and to confirm that you have properly recognized the funding sources.

7.1 Funding Acknowledgement

A sample acknowledgement of funding by an NSF grant might read something like the following.

> This research was supported by the National Science Foundation under Grant No. [Your Grant Number]. The authors would like to thank [any other contributors, institutions, or facilities] for their support and collaboration. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation."