Advanced Image Processing and Analysis

ECE 4438B/ECE 9202B/ECE 9022B

BIOMED/BIOPHYS/CAMI 9519B

Winter 2019

Instructor

Elvis Chen, PhD, LL (echen29@uwo.ca)

Robarts Research Institute

Department of Electrical and Computer Engineering

Biomedical Engineering Graduate Program

Department of Medical Biophysics

Advanced Image Processing and Analysis (AIPA)

- Offered to 3 departments (undergrad/grad)
  - ECE 4438/ECE 9202/ECE 9022
  - BIOMED 9519
  - BIOPHYS/CAMI 9519
- Time and Place
  - Monday 2:30-3:30pm, SH-3007
  - Tuesday 5:30-7:30pm, SEB-3109

Theme and key questions

- Theme: Digital image analysis with examples drawn from medical imaging
- Questions:
  - How to design and implement algorithms to delineate structures of interests (Segmentation)?
  - How to design and implement algorithms to align different images (Registration)?
  - How to evaluate the performance of these algorithms (Validation)?

Applications

- Entertainment
  - Movies (special effects)
- Informatics
  - Self-driving car
  - Adjuncts to computer vision and computer graphics
- Augmentation
  - Medical interventions

What/Motivation (Examples)

https://www.youtube.com/watch?v=4NU9ikjqjC0

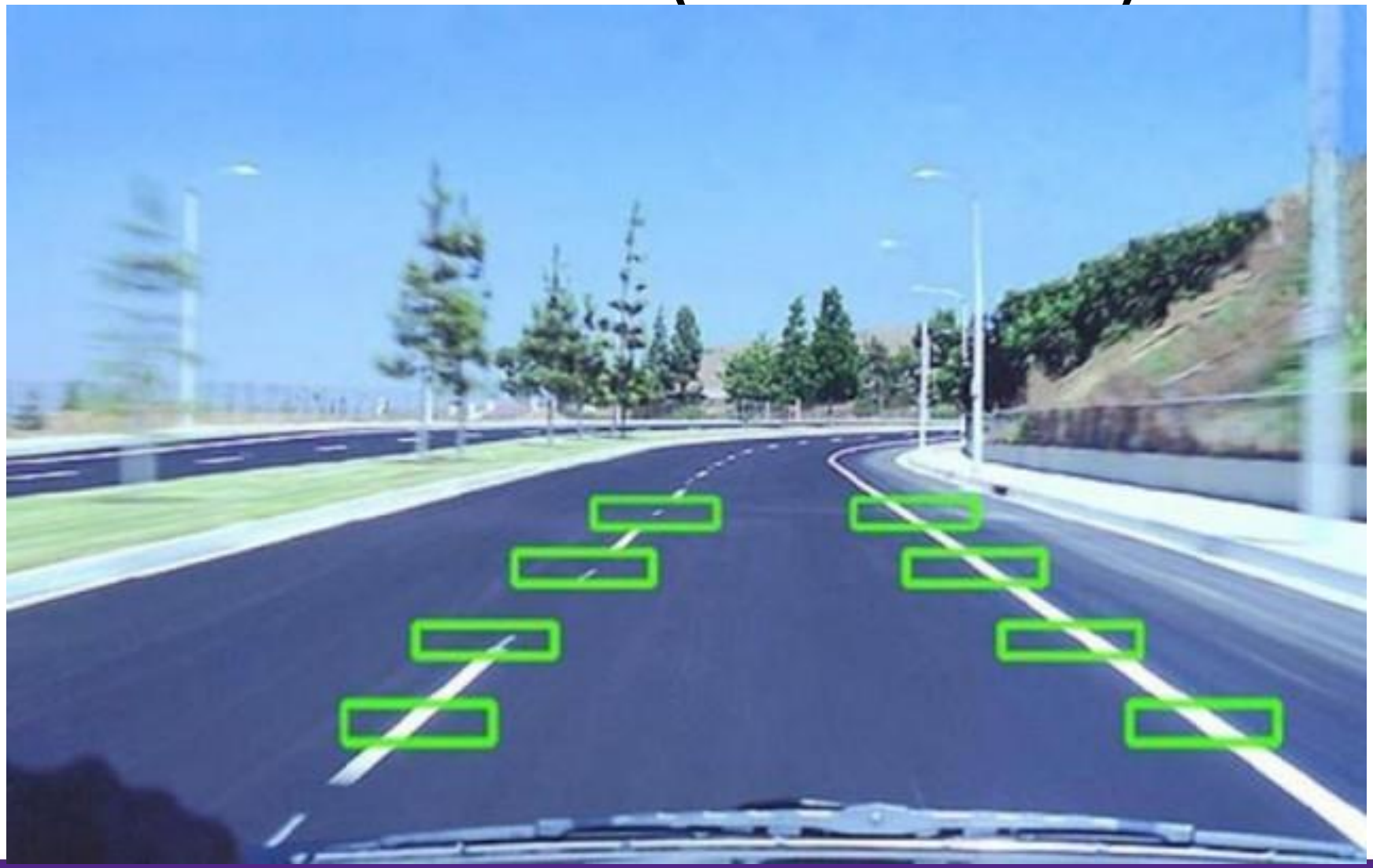
Applications

- Entertainment
  - Movies (special effects)
- Informatics
  - Self-driving car
  - Adjuncts to computer vision and computer graphics
- Augmentation
  - Medical interventions

Motivation (Land Departure Warning)

https://www.youtube.com/watch?v=QeE_-NrXTIA

Motivation (Informatics)

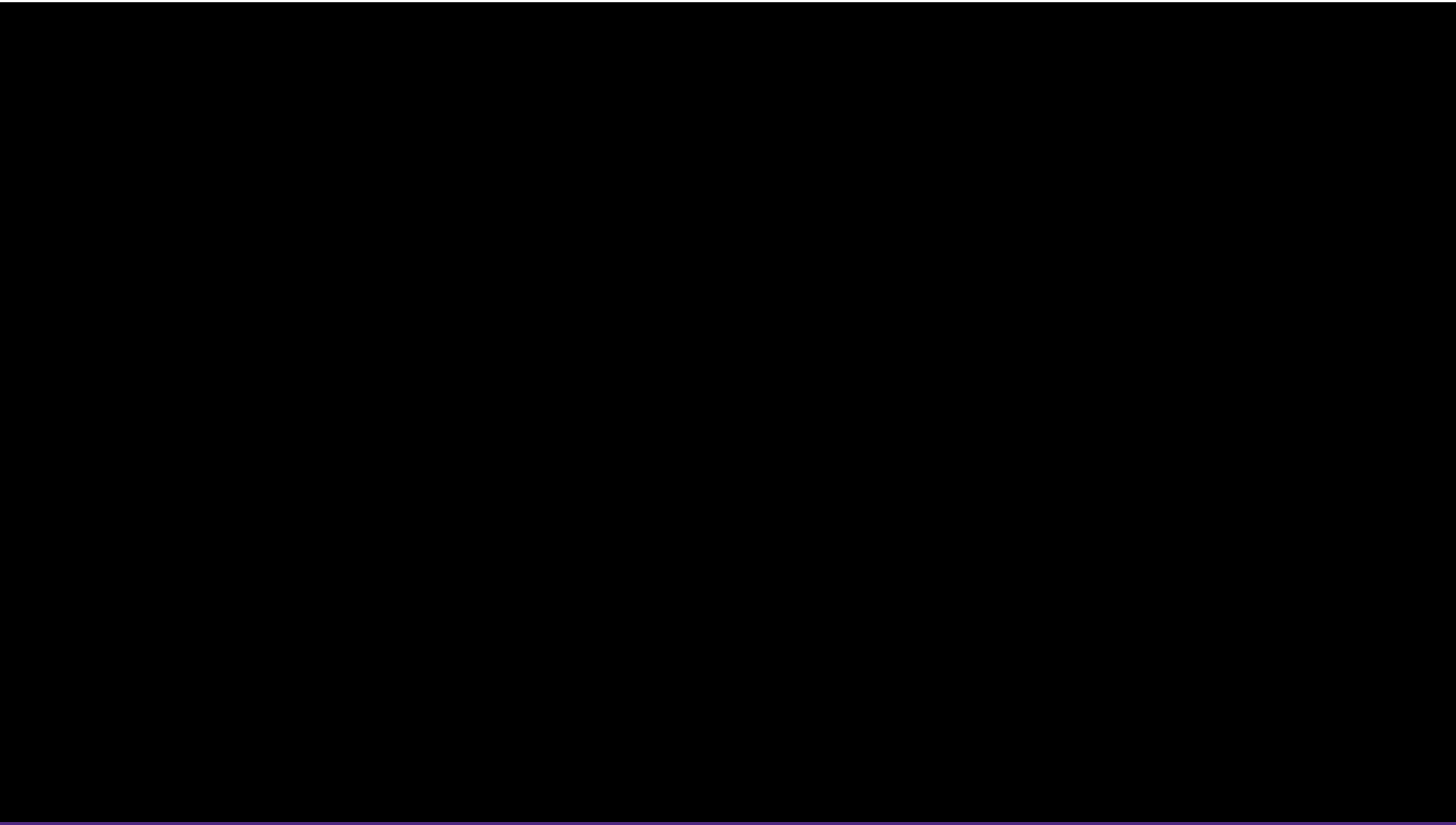


http://g3ict.org/resource_center/newsletter/news/p/id_411

Motivation (Interventions)

https://www.youtube.com/watch?v=e9ASH8IBJ2U

Motivation



https://www.youtube.com/watch?v=SKpKlh1-en0

Theme and key questions

- Theme: Digital image analysis with examples drawn from medical imaging
- Questions:
  - How to design and implement algorithms to delineate structures of interests?
    - Segmentation
      - Manual
      - Semi-automatic
      - Automatic

Theme and key questions

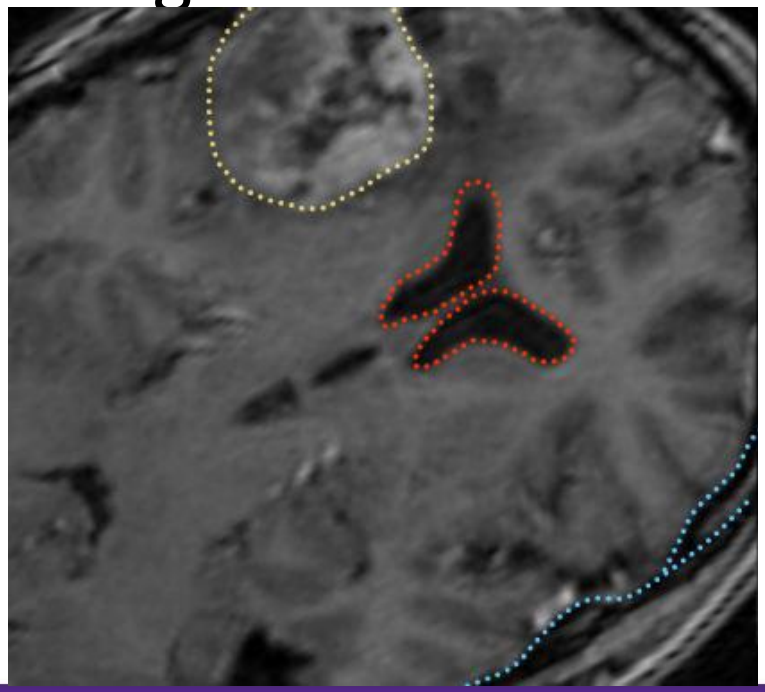
- Theme: Digital image analysis with examples drawn from medical imaging
- Questions:
  - How to design and implement algorithms to align different images?
    - Registration
      - Rigid registration
      - Deformable registration

Theme and key questions

- Theme: Digital image analysis with examples drawn from medical imaging
- Questions:
  - How to evaluate the performance of these algorithms?
    - Evaluation

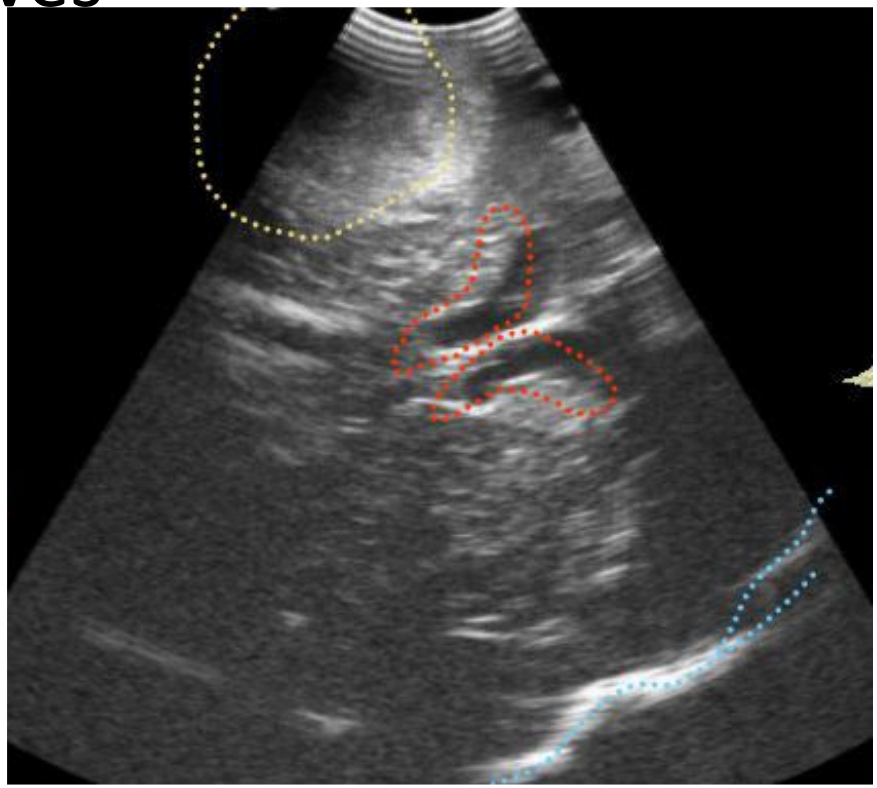
Clinical Context

- Medical Images are often acquired pre-operatively (MRI/CT), used for diagnostics and surgical planning



Clinical Context

- However, intra-operatively, organ deforms/moves



Clinical Context

- Pre-op MRI registered to intra-op ultrasound to compensave for brain shift/motion

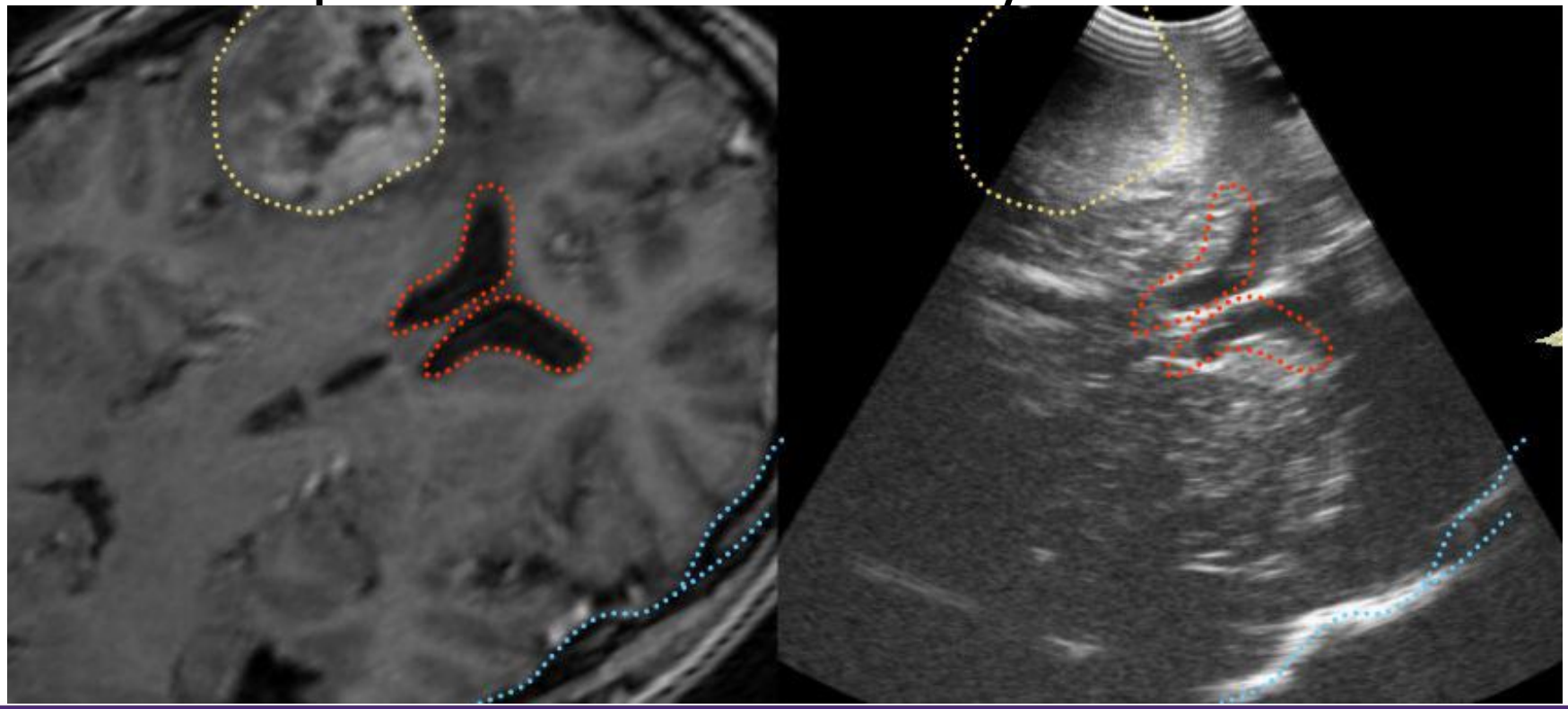
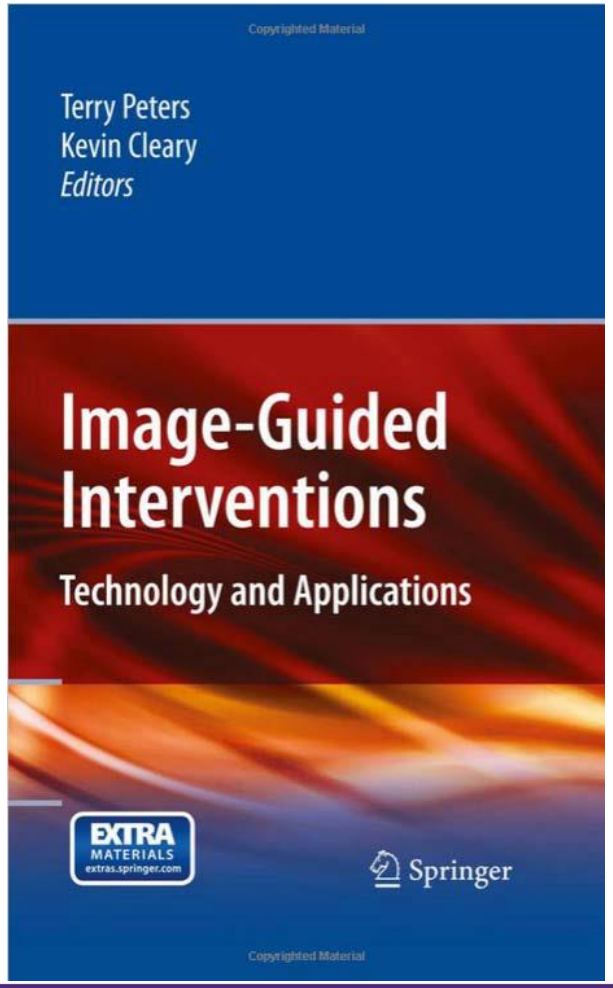


Image-Guided Interventions, Springer, 2008

Clinical Context



- PDF version available free of charge through Western's library: https://journals.scholarsportal.info/details/15239829/v12inone/119_iitraca.xml

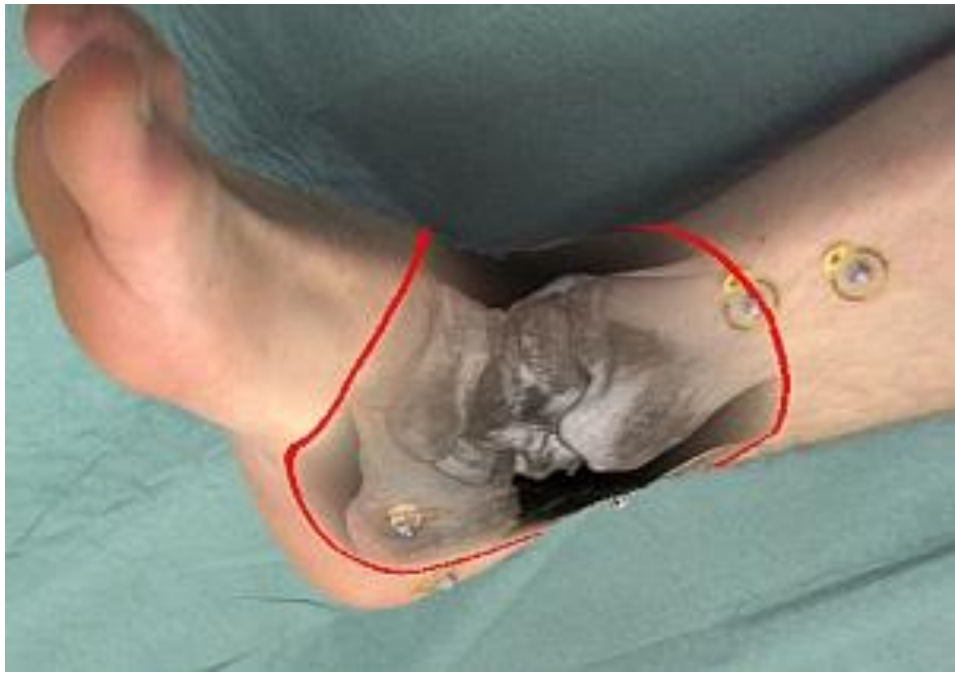
Application: Vessel Stenosis

- Segmentation of an MR angiogram allows 3D depiction of vessels and detection of stenosis (vessel narrowing)
- On the basis of segmentation, the diameter of stenosis can accurately be determined

Image-Guided Interventions, Springer, 2008

Application: Orthopaedic Surgery

- Virtual representation of the ankle determined from pre-op CT, volume-rendered, and registered to patient intra-operative to allow an augmented reality display



Evaluation

	Assignments	Mid-term	Final Exam	Project
Undergrad	20%	30%	50%	n/a
MEng	20%	30%	50%	n/a
MSc	20%	20%	35%	25%
PhD	20%	20%	35%	25%

- Undergrad (ECE 4438B) and MEng (ECE9022)
  - 4 assignments
  - Closed-book mid-term exam (Feb 26th, 2hr)
    - Review session on Feb 12 and 13 (tentative)
  - Closed-book final exam (comprehensive)

Evaluation

	Assignments	Mid-term	Final Exam	Project
Undergrad	20%	30%	50%	n/a
MEng	20%	30%	50%	n/a
MSc	20%	20%	25%	35%
PhD	20%	20%	25%	35%

- Graduate students
  - 4 assignments
  - Closed-book mid-term exam (Feb 26th, 2hr)
  - Closed-book final exam (comprehensive)
  - Project involves implementation of an image processing technique/algorithm
    - Topic chosen with consultation with course instructor
    - With alignment to thesis topic/work

Policy

- Late submission policy: penalized at a rate of 20% per 24 hours overdue, NO EXCEPTIONS
- Any reason for late submission must be
  - Brought to the attention to the instructor BEFORE the deadline
  - With documentation
  - No consideration after the assignment deadline

Policy

- Use of English: in accordance with Senate and Faculty Policy, students may be penalized up to 10% of the marks on assignments, tests, and examinations for improper use of English
- Poorly written work may be returned without grading (except the final exam)

Policy

- Attendance: Any student who is absent too frequently from class will be reported to the Dean (after due warning has been given)
- On the recommendation of the department, and with permission of the Dean, the student will be debarred from taking the regular final examination

Policy

- Mid-term and the final examination cannot be missed:
  - Unless
    - Due to illness
    - Other extreme circumstances (death of family member, etc.)
  - Should consult with the instructor/Department Chair immediately
  - Documentation

Course format

- OWL (https://owl.uwo.ca/)
  - Course notes and assignments can be downloaded
  - Assignments are submitted via OWL
  - OWL entry is "**ECE 4438B 001 FW18"**
- Github
  - Additional course materials (codes and examples) are accessible via github
  - Jupyter notebook

Theme and Topics

- Theme: Digital image processing and analysis, with examples drawn from medical imaging
- Topics:
  - Segmentation
  - Registration
  - Validation
  - Data Augmentation for Deep Learning (time permits)

Implementation Tools

- Python (https://www.python.org/)
  - Interpreted language
  - Power of C++
  - Ease of Matlab
  - Processing + visualization, large number of algorithms, best suited to 2D

Implementation Tools

- Insight Segmentation and Registration Toolkit (ITK, https://itk.org/)
  - Library for image processing
  - Open source
  - Implemented in C++, with binding to many other languages include python and java
  - Processing only, vary large number of algorithms, suited to 2D/3D/4D/nD

Implementation Tools

- SimpleITK (www.simpleitk.org)
  - Open source
  - Simplified layer built on top of ITK
  - Intended to facilitate its use in rapid prototyping, education, interpreted languages
    - Python
    - R

Implementation Tools

- Jupyter Notebook (http://jupyter.org/)
  - Open-source web application that allows one to create and share documents that contain **live code**, **equations**, **visualizations**, and **narrative text**
  - SimpleITK kernel
  - Executes python/SimpleITK codes within a browser (think Matlab) and visualize the results immediately

ITK

```
#include "itkImage.h"
#include "itkImageFileReader.h"
#include "itkRescaleIntensityImageFilter.h"
#include "itkLaplacianSharpeningImageFilter.h"
#include "itkSubtractImageFilter.h"

#include "QuickView.h"

int main(int argc, char * argv[])
{
  // Verify command line arguments
  if( argc < 2 )
    {
    std::cerr << "Usage: " << std::endl;
    std::cerr << argv[0] << " inputImageFile" << std::endl;
    return EXIT_FAILURE;
    }

  // Parse command line arguments
  std::string inputFilename = argv[1];
```

https://itk.org/Wiki/ITK/Examples/ImageProcessing/LaplacianSharpeningImageFilter

ITK

```
// Setup types
typedef itk::Image< float,  2 >   FloatImageType;

typedef itk::ImageFileReader< FloatImageType >  readerType;
readerType::Pointer reader = readerType::New();
reader->SetFileName(inputFilename);

typedef itk::LaplacianSharpeningImageFilter<FloatImageType, FloatImageType >  LaplacianSharpeningImageFilterType;
LaplacianSharpeningImageFilterType::Pointer laplacianSharpeningImageFilter =
  LaplacianSharpeningImageFilterType::New();
laplacianSharpeningImageFilter->SetInput( reader->GetOutput() );

typedef itk::SubtractImageFilter<FloatImageType>           SubtractType;
SubtractType::Pointer diff = SubtractType::New();
diff->SetInput1(reader->GetOutput());
diff->SetInput2(laplacianSharpeningImageFilter->GetOutput());
```

https://itk.org/Wiki/ITK/Examples/ImageProcessing/LaplacianSharpeningImageFilter

ITK

```
  QuickView viewer;
  viewer.AddImage(
    reader->GetOutput(),true,
    itksys::SystemTools::GetFilenameName(argv[1]));

  std::stringstream desc;
  desc << "LaplacianSharpeningImageFilter";
  viewer.AddImage(
    laplacianSharpeningImageFilter->GetOutput(),
    true,
    desc.str());

  std::stringstream desc2;
  desc2 << "Original - LaplacianSharpening";
  viewer.AddImage(
    diff->GetOutput(),
    true,
    desc2.str());

  viewer.Visualize();
  return EXIT_SUCCESS;
}
```

https://itk.org/Wiki/ITK/Examples/ImageProcessing/LaplacianSharpeningImageFilter

ITK

```
cmake_minimum_required(VERSION 2.8.12)

project(LaplacianSharpeningImageFilter)

find_package(ITK REQUIRED)
include(${ITK_USE_FILE})
if (ITKVtkGlue_LOADED)
  find_package(VTK REQUIRED)
  include(${VTK_USE_FILE})
else()
  find_package(ItkVtkGlue REQUIRED)
  include(${ItkVtkGlue_USE_FILE})
  set(Glue ItkVtkGlue)
endif()

add_executable(LaplacianSharpeningImageFilter MACOSX_BUNDLE LaplacianSharpeningImageFilter.cxx)
target_link_libraries(LaplacianSharpeningImageFilter
  ${Glue}  ${VTK_LIBRARIES} ${ITK_LIBRARIES})
```

https://itk.org/Wiki/ITK/Examples/ImageProcessing/LaplacianSharpeningImageFilter

SimpleITK Notebooks

Image Sharpening Example using SimpleITK

This is a simple demonstration of the power of SimpleITK, based on its C++ equivalent shown here:
https://itk.org/Wiki/ITK/Examples/ImageProcessing/LaplacianSharpeningImageFilter

We assume the an image is available and located in the same directory as this Jupyter Notebook. In this case, an image of a woodpicker is copied and saved as 'woodpicker.png" from the URL above.

In [ ]:
```
# import SimpleITK library
import SimpleITK as sitk
```

In [ ]:
```
# Assuming the image is located under the data/image directory, this is how we can load an image
#
# The input to the function is a string (enclosed in '') of the file name.
# The output is the image/array. Note no explicit typing is needed in Python
img = sitk.ReadImage('..\data\images\woodpicker.png')
```

In [ ]:
```
# display the image. If we set up our environment properly using the in-class example, ImageJ will be used to display the image.
sitk.Show(img)

# Note there is a slider bar in ImageJ, why?
```

In [ ]:
```
lap = sitk.LaplacianSharpeningImageFilter() # this is an in-line comment
```

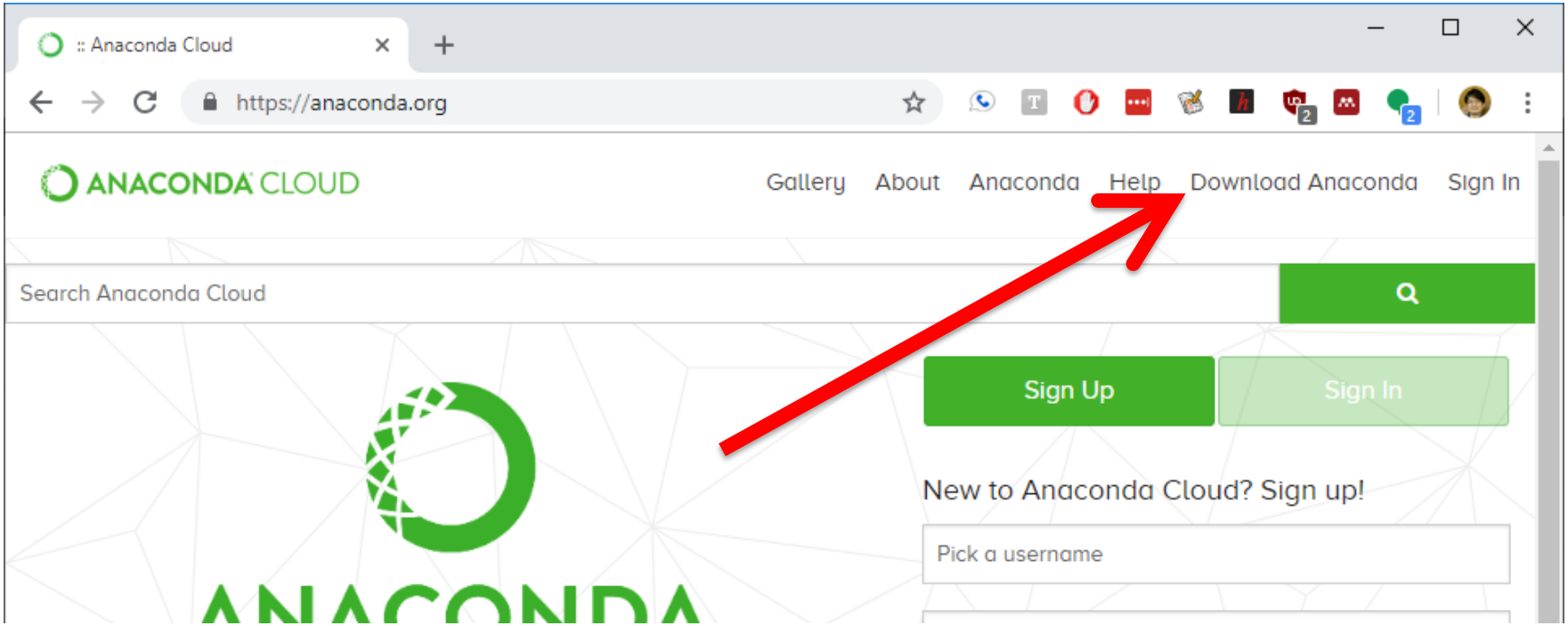
In [ ]:
```
outimg = lap.Execute(img) # input is the original image, output is the sharpened image
```

In [ ]:
```
sitk.Show(outimg) # display the result.
```

This is it! Compare this python code in SimpleITK to the C++ equivalent in ITK.

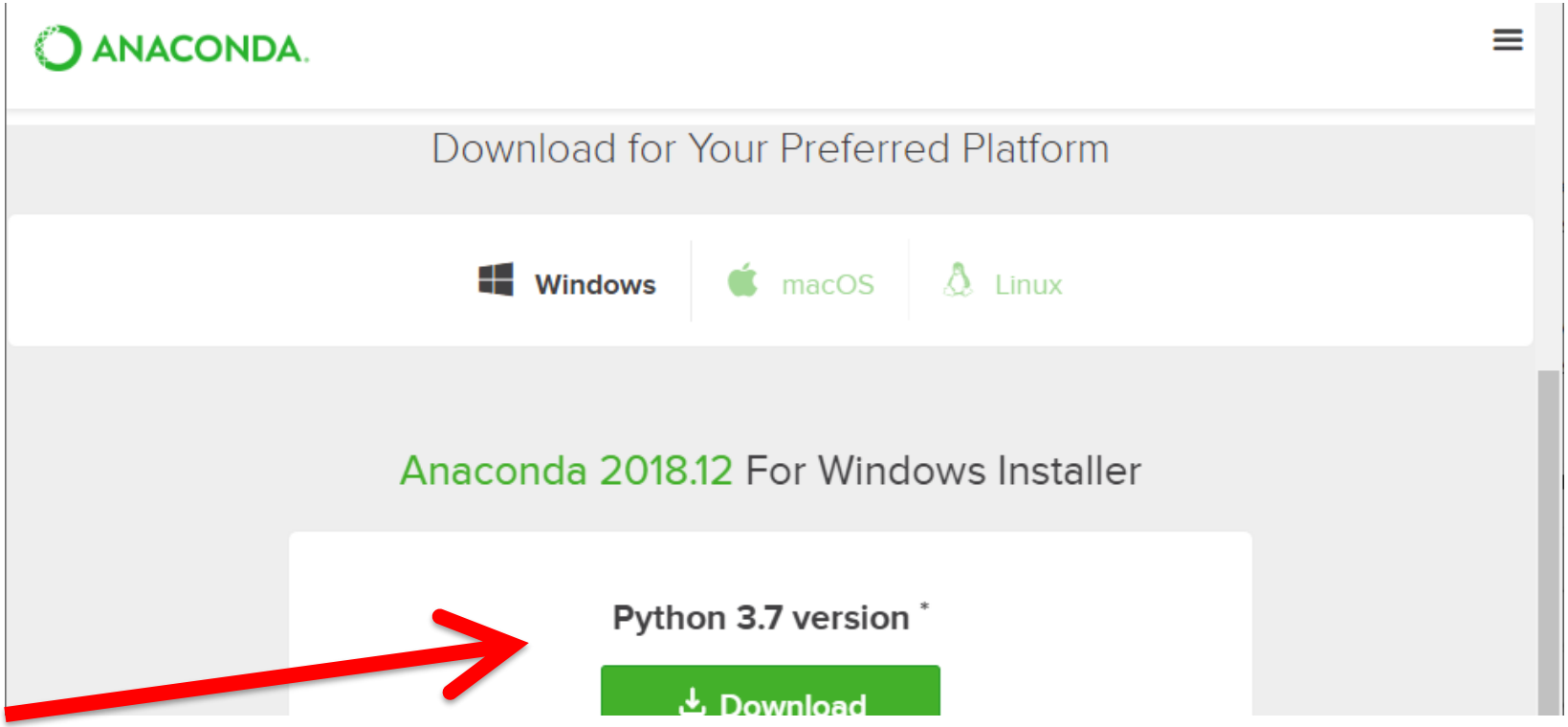
Implementation

- Install Anaconda (https://anaconda.org/)
  - Download (top-right)



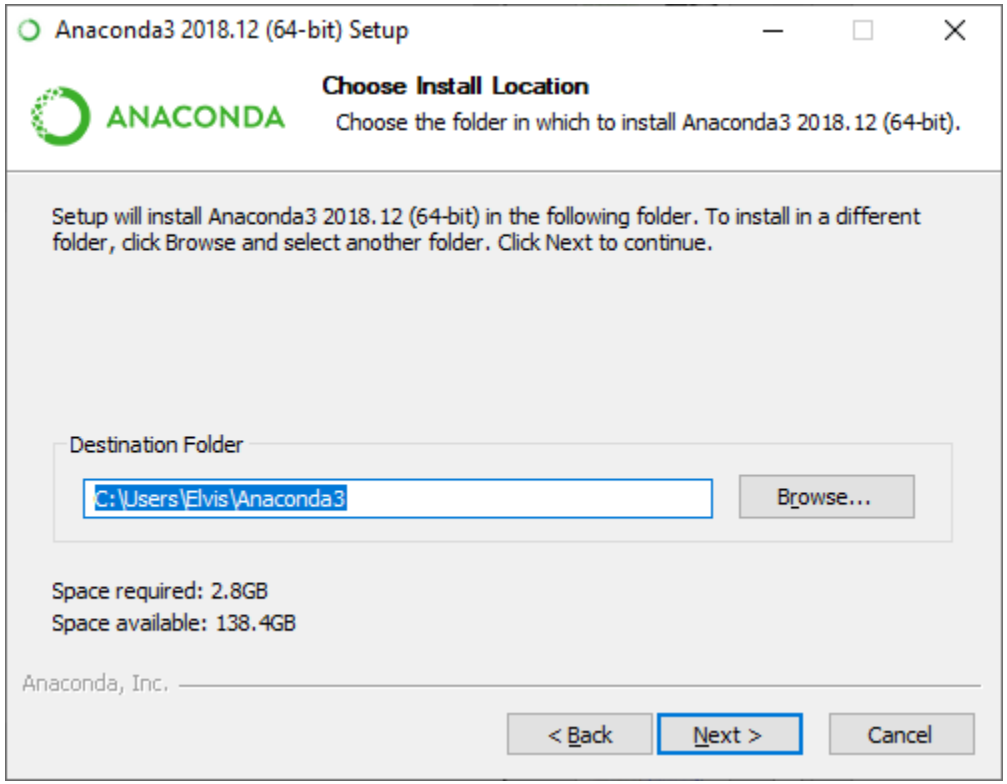
Implementation

- Download Python 3.7



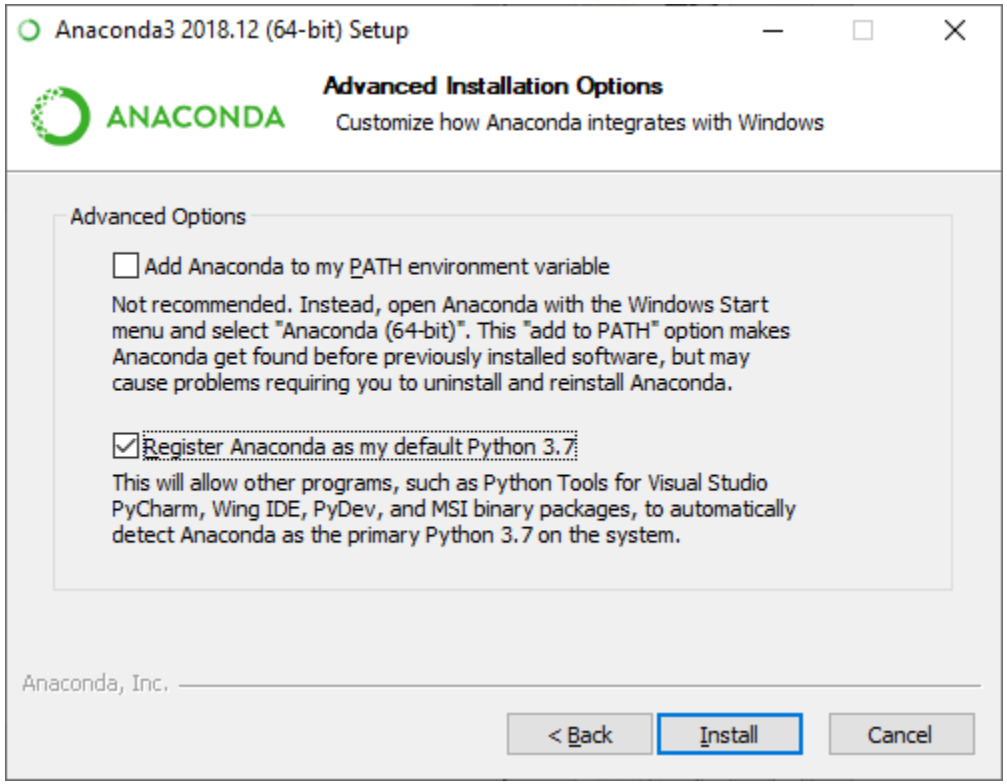
Implementation

- Install



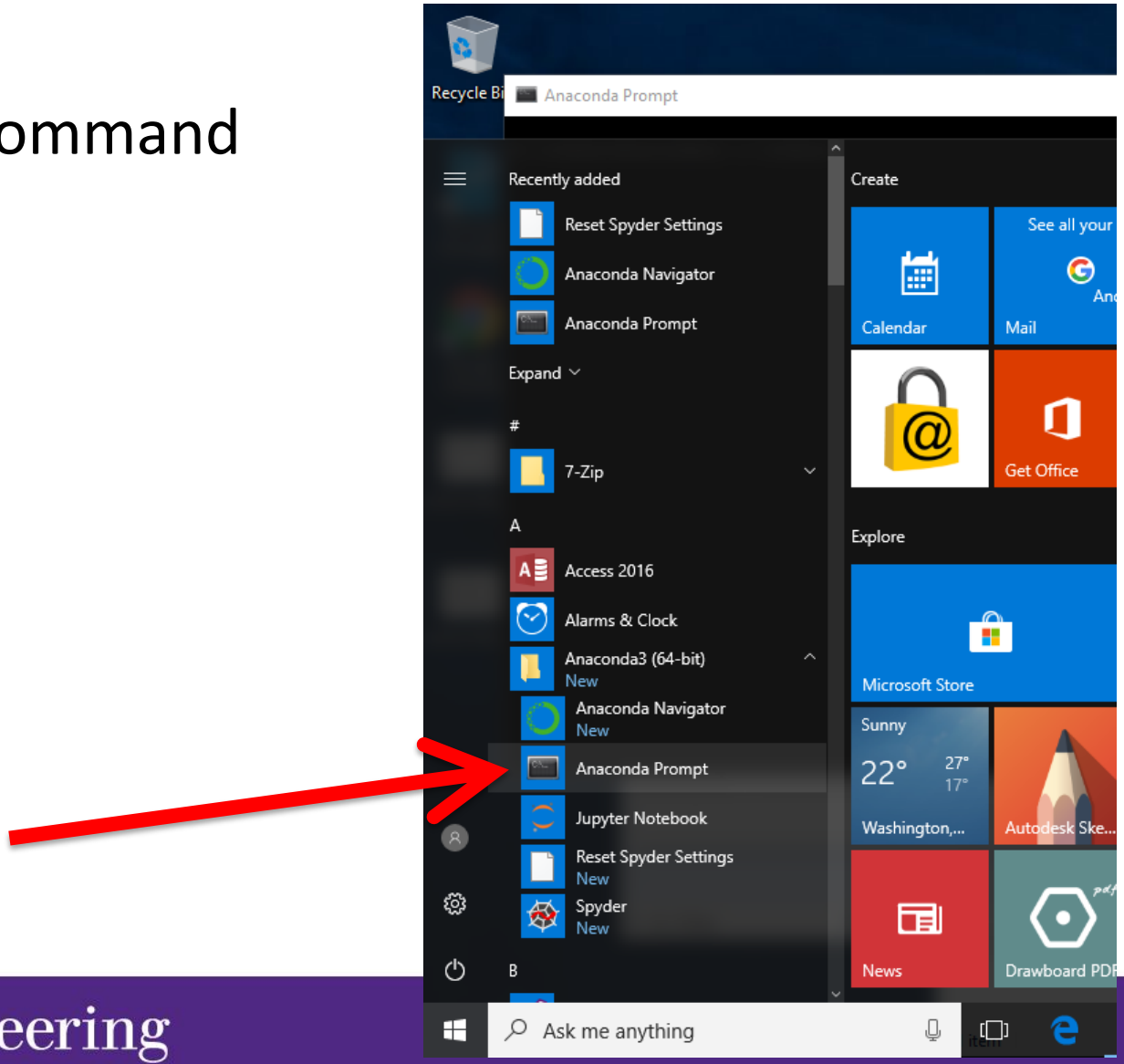
Implementation

- Install



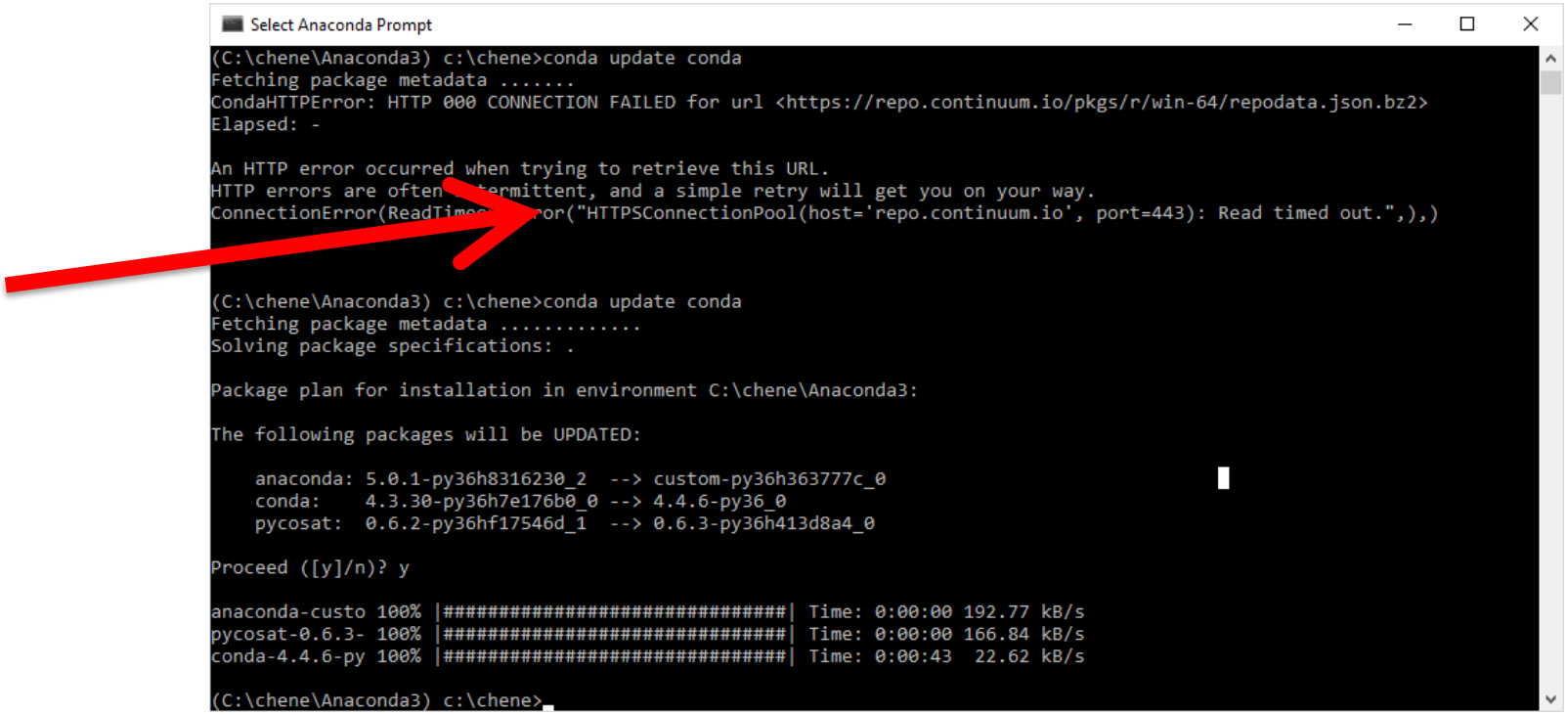
Implementation

- Anaconda Command Prompt



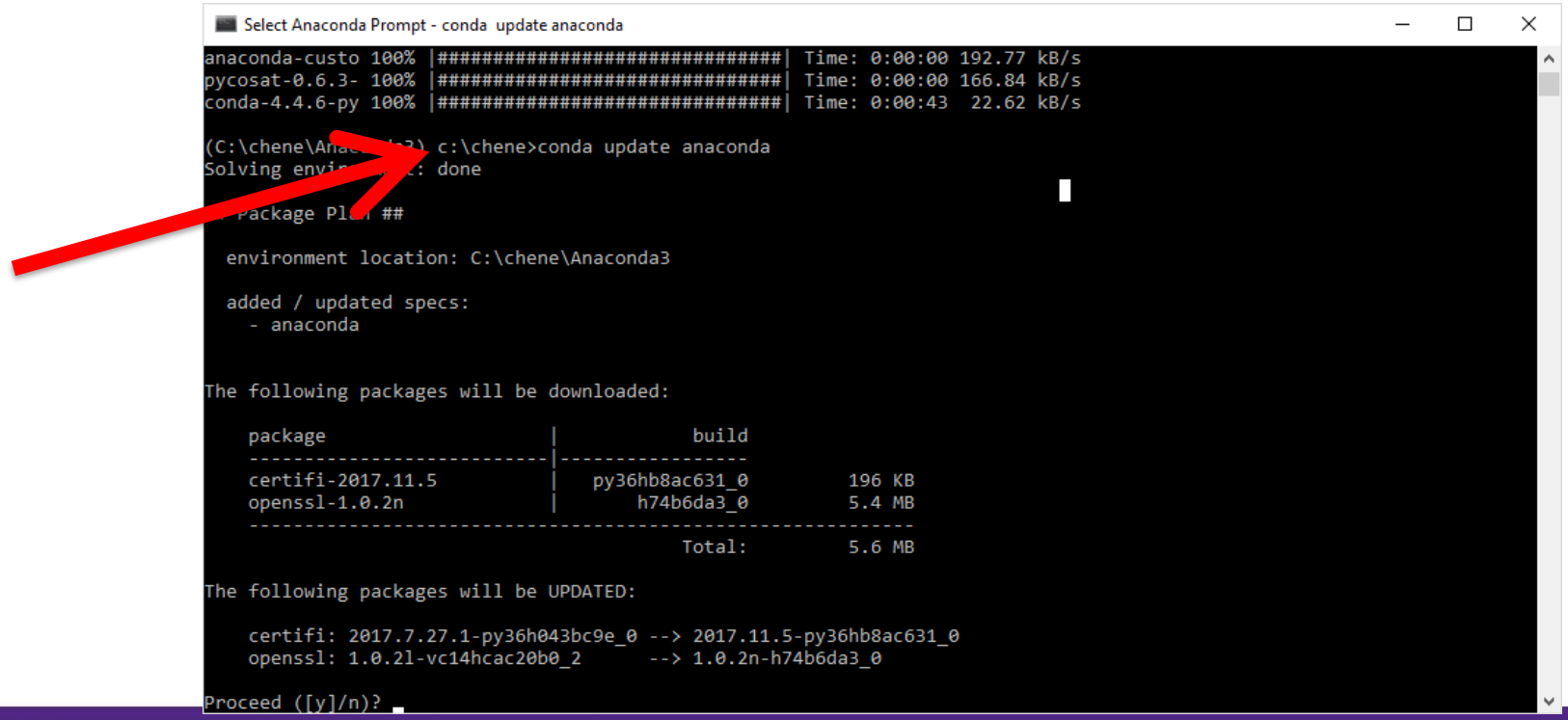
Implementation

- Update using the following 2 commands
- ‘conda update conda’



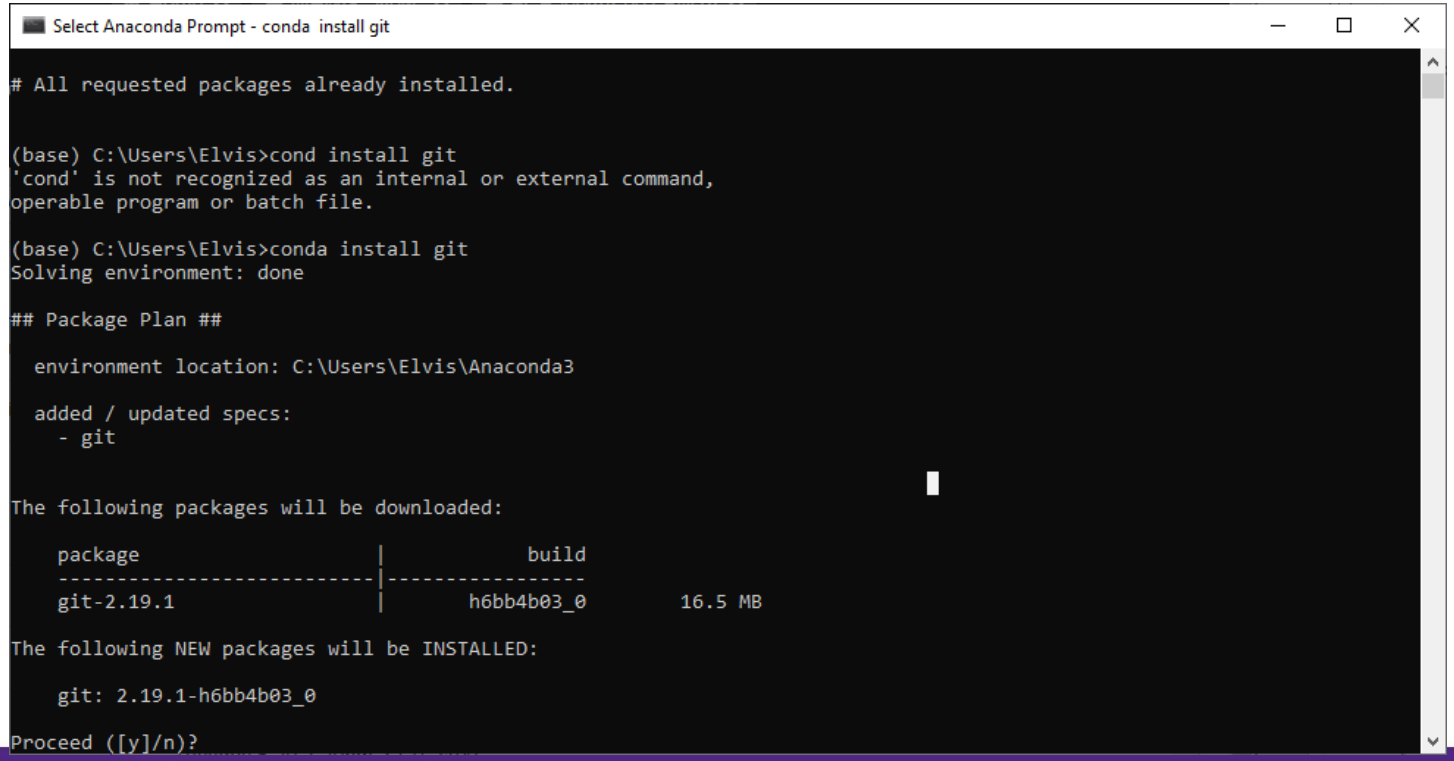
Implementation

- Update using the following 2 commands
- ‘conda update anaconda’



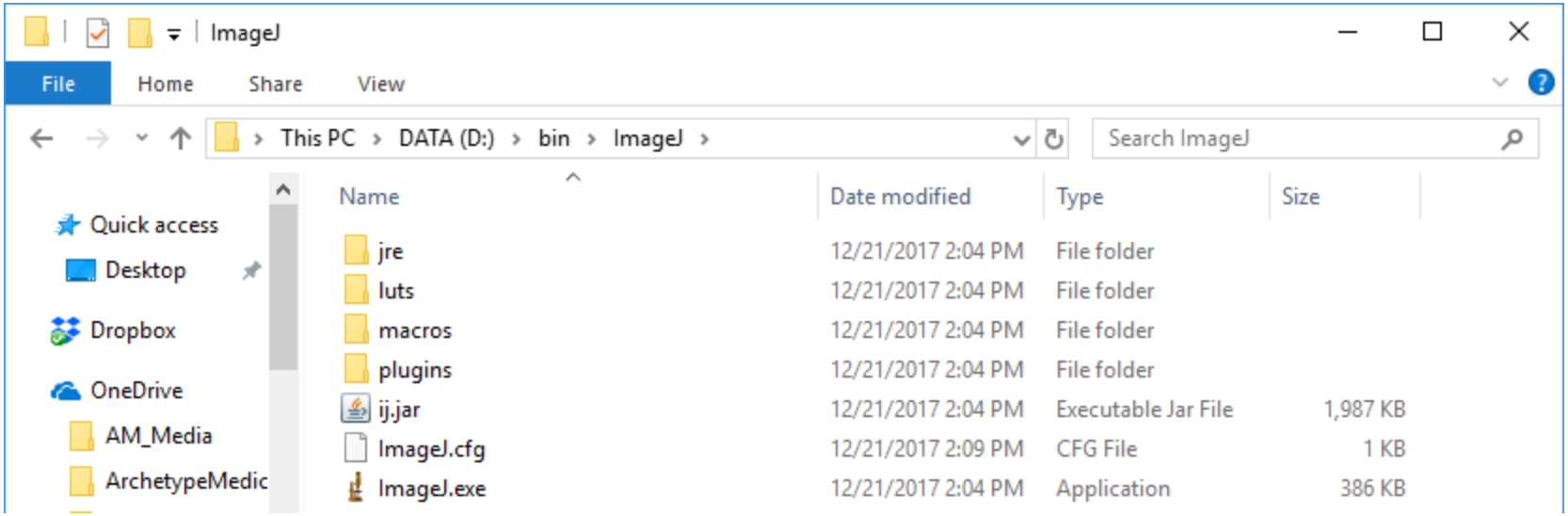
Implementation

- (Optional) install git
- ‘conda install git



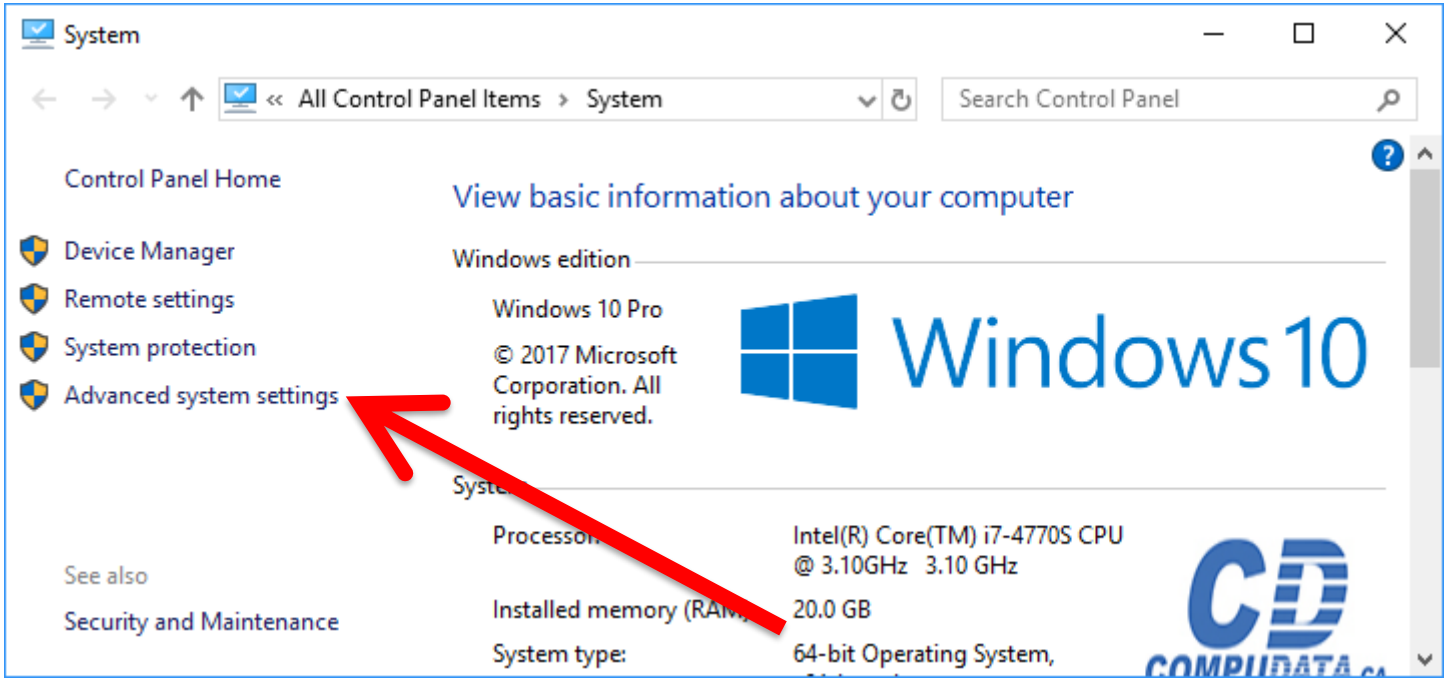
Implementation

- Install imagej
  - Download it from https://imagej.nih.gov/ij/download.html
  - extract the zip file (no 'installation' required)



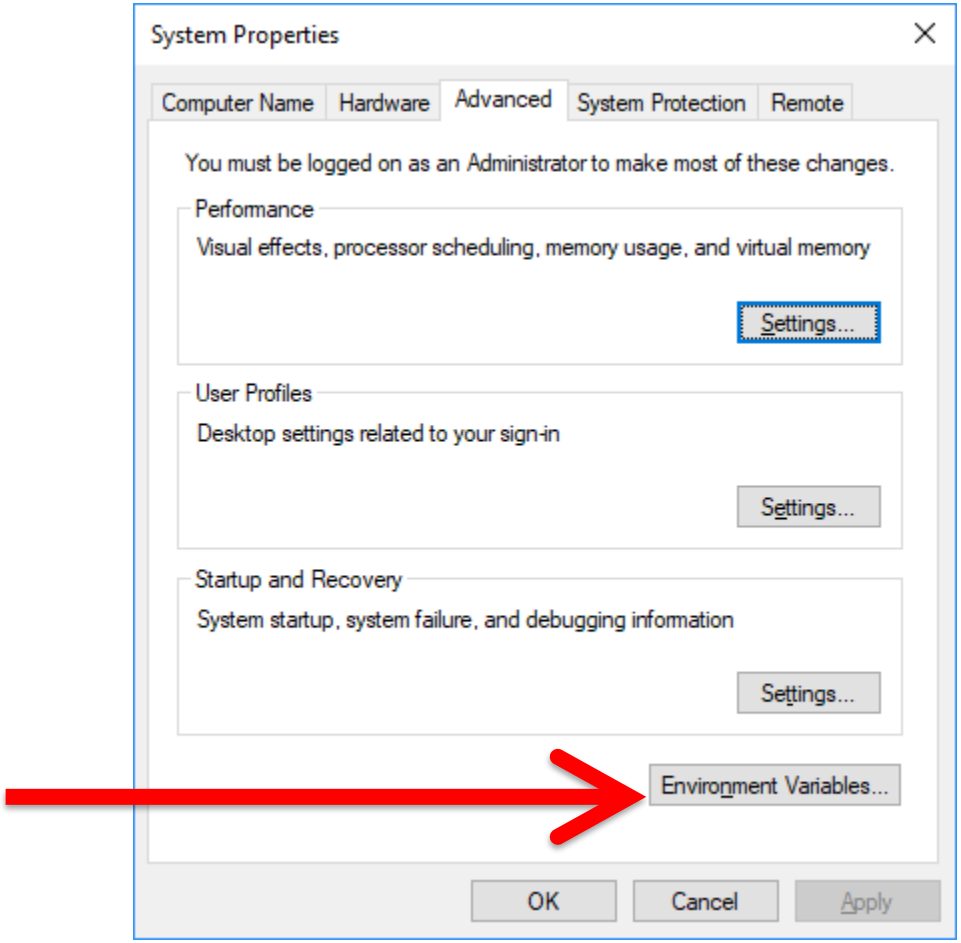
Implementation

- Install imagej
  - Put the location of ‘ImageJ.exe’ to path



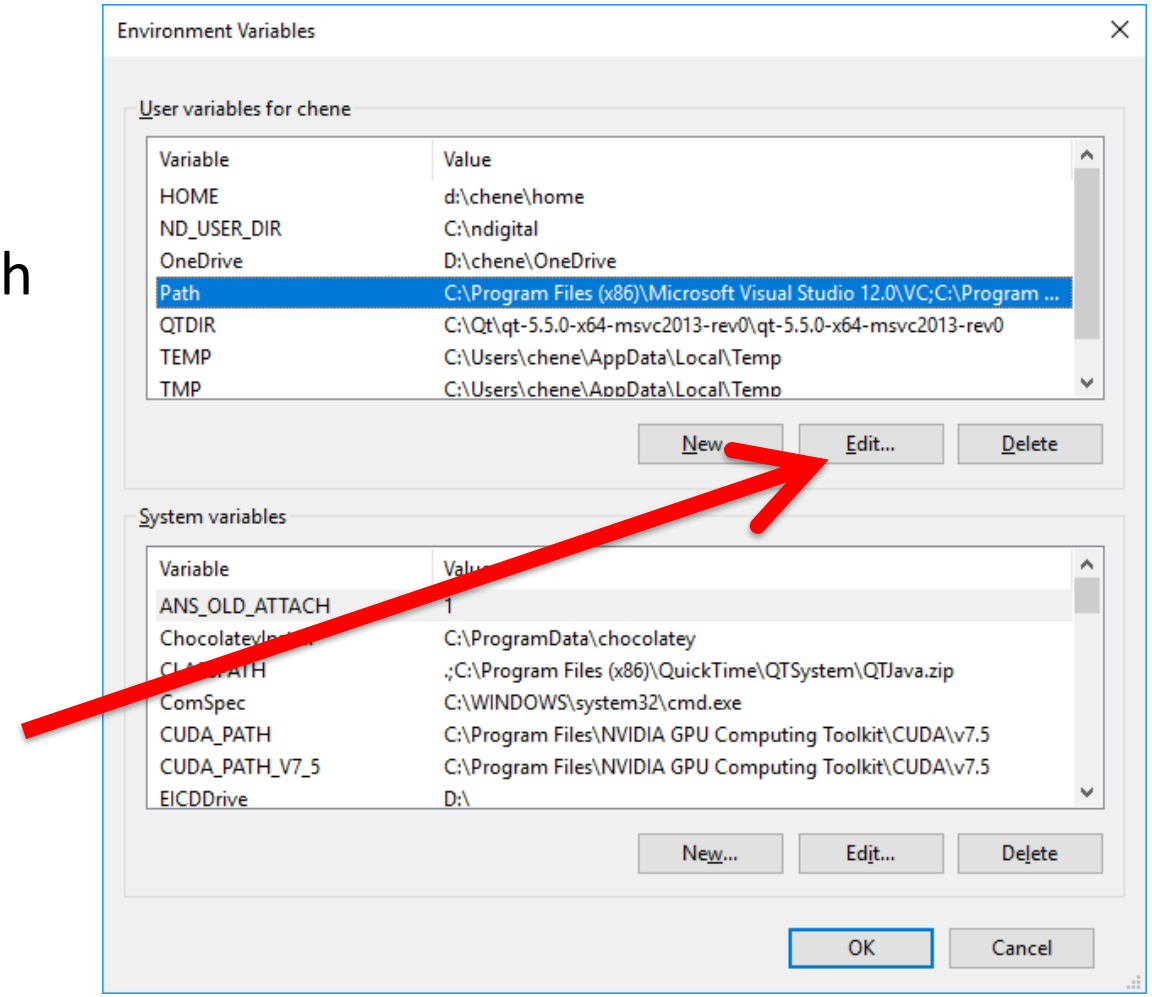
Implementation

- Install imagej
  - Put the location of ‘ImageJ.exe’ to path



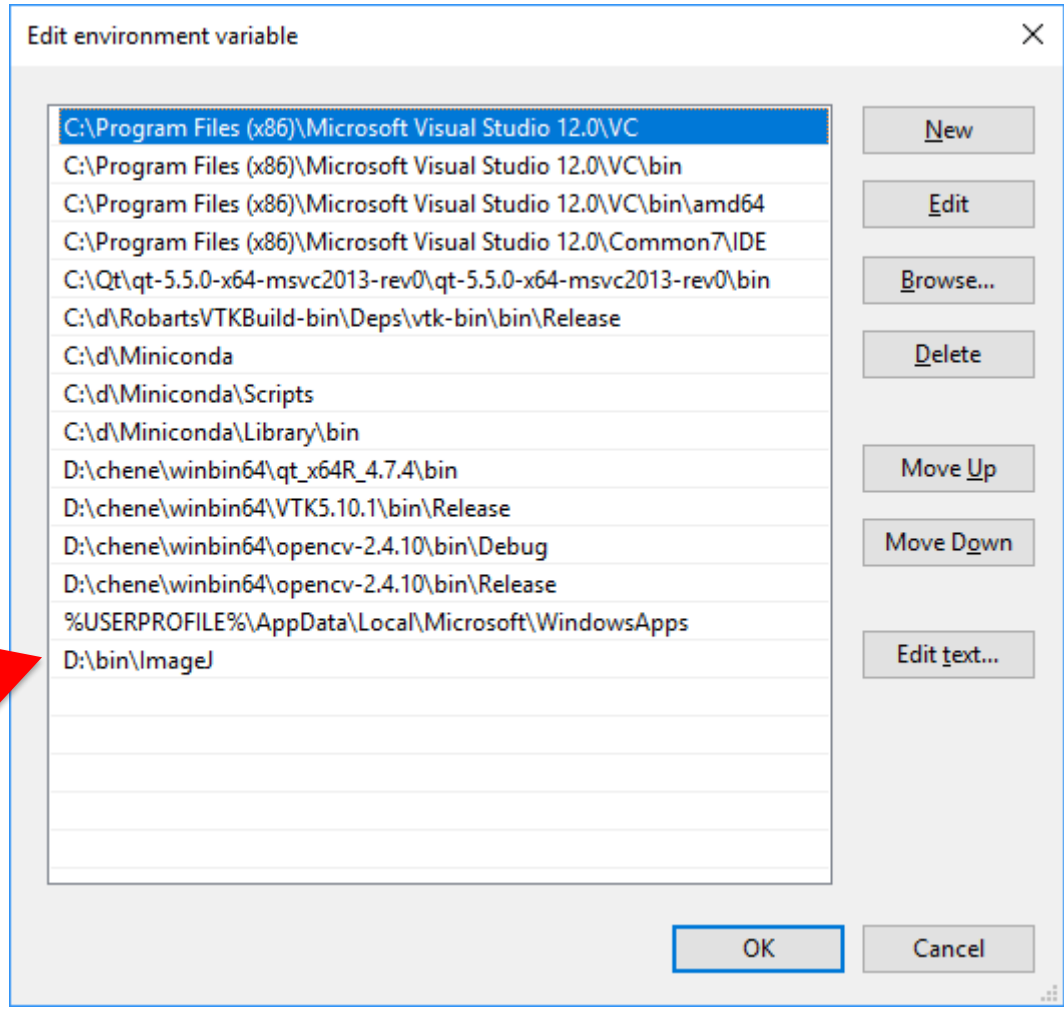
Implementation

- Install imagej
  - Put the location of ‘ImageJ.exe’ to path



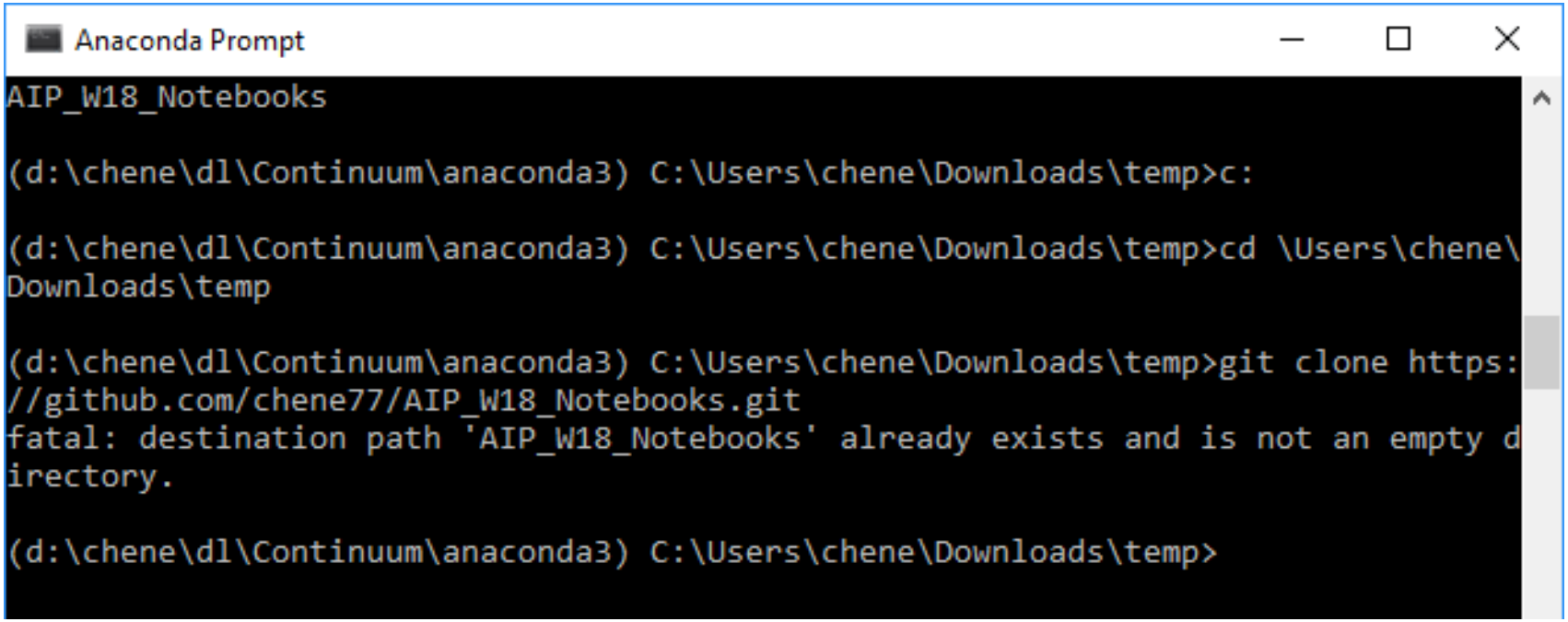
Implementation

- Install imagej
  - Put the location of ‘ImageJ.exe’ to path



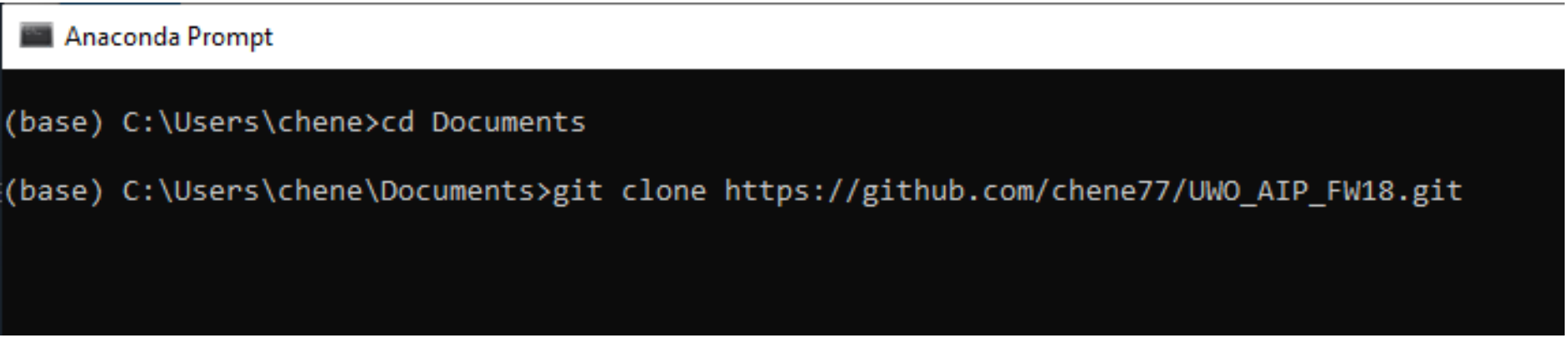
Implementation

- In an Anaconda prompt, create a directory at a location of choice,



Implementation

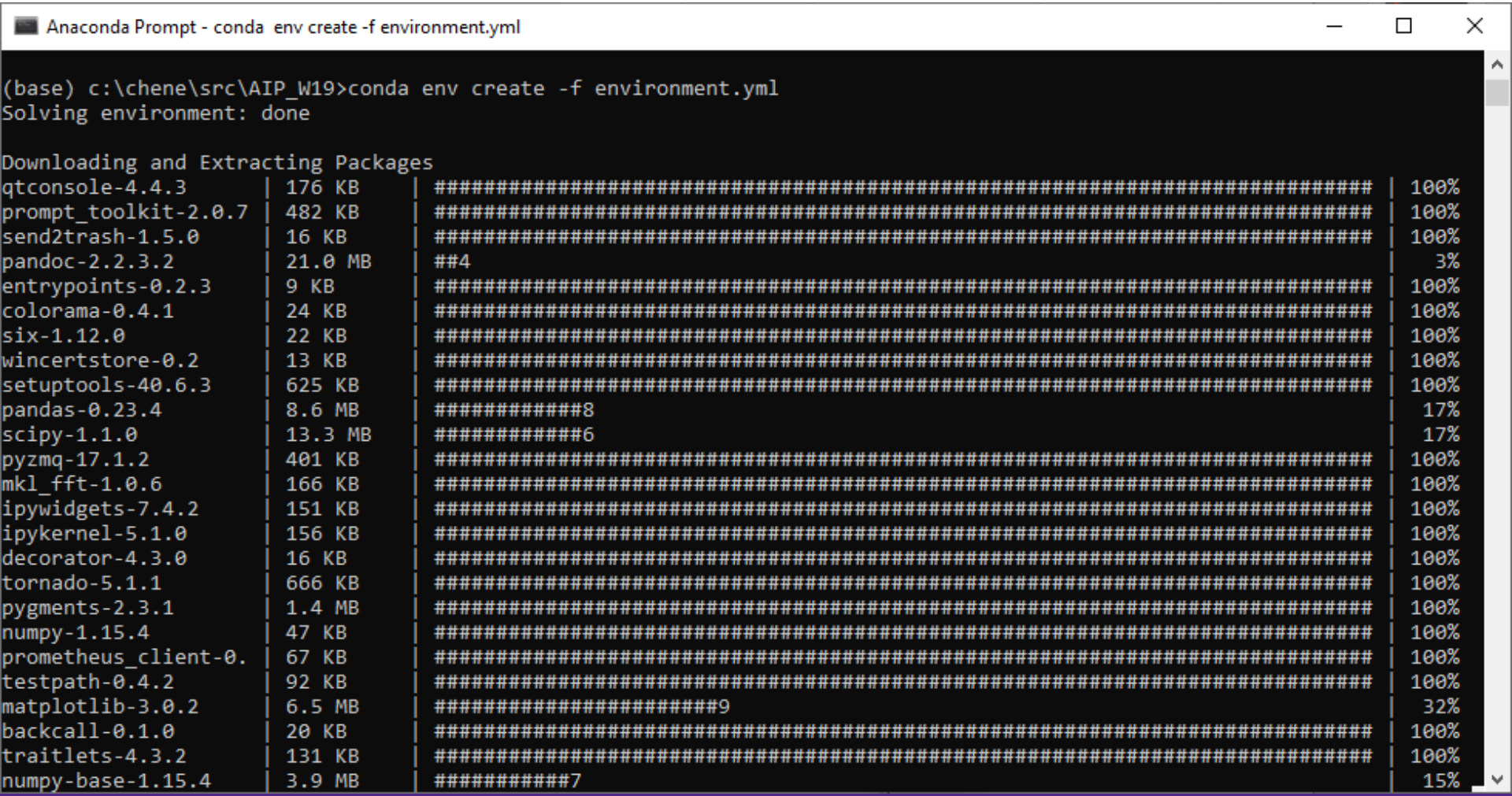
- Using the newly install git, clone the following github repository: ‘git clone https://github.com/chene77/UWO_AIP_FW18.git’



Implementation

- ‘cd’ to the main git directory, create an environment for SimpleITK using the following command ‘conda env create –f environment.yml’

Implementation



Implementation

- If you are getting error message such as “Error([('SSL routines', 'ssl3_get_record', 'decryption failed or bad record mac’)])” than your internet connection is unstable. Repeat the process until it is completed.

Implementation

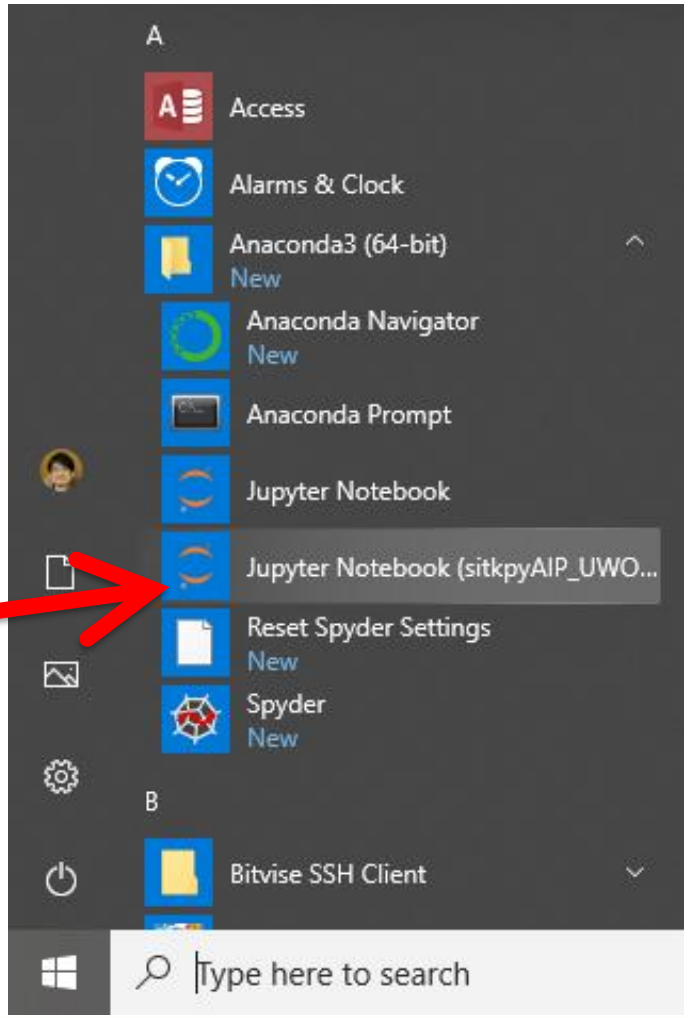
- After the installation is complete, activate SimpleITK environment with the following ’conda activate sitkpyAIP_UWO_W19’

```
#
# To activate this environment, use
#
#     $ conda activate sitkpyAIP_UWO_W19
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:\Users\ECE4438B\Documents\UWO_AIP_FW18>
```

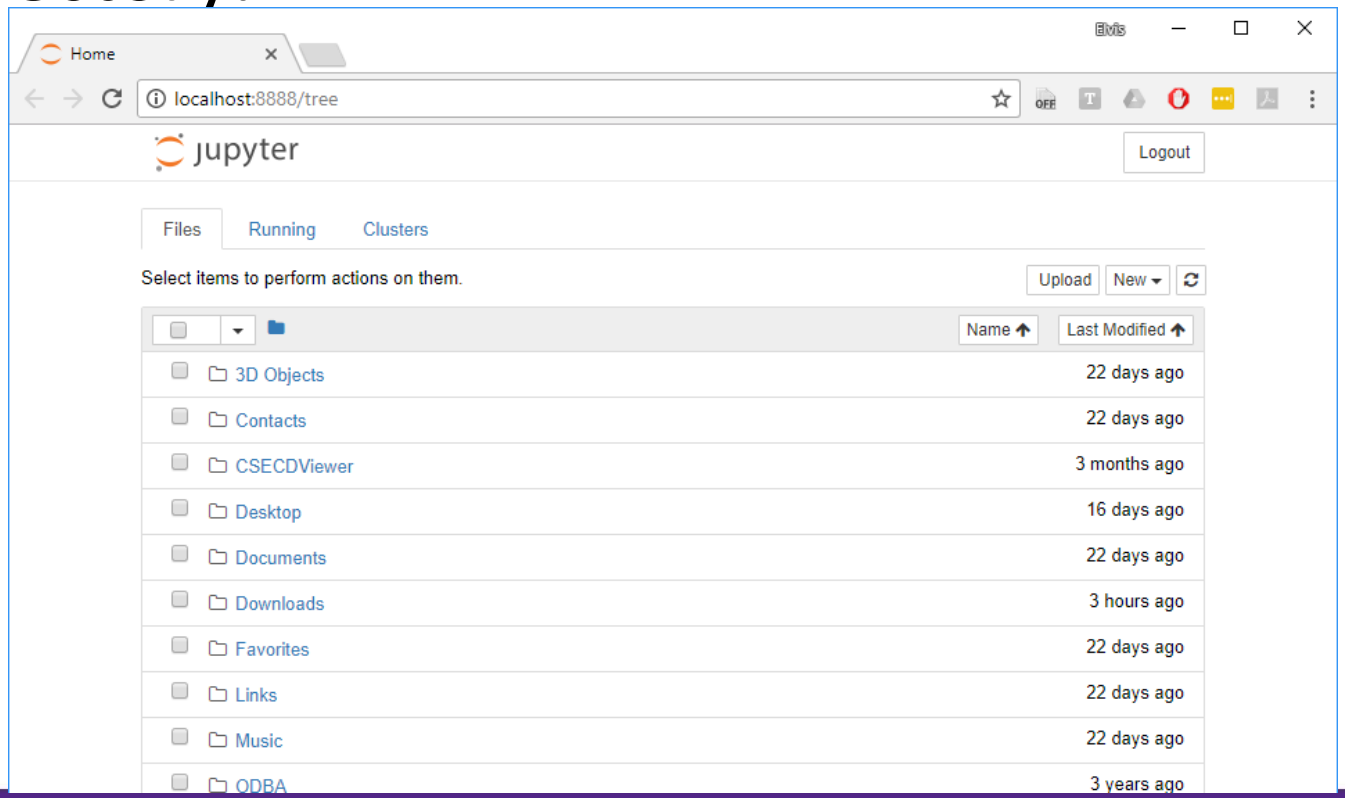
Implementation

- Now you should have a Jupyter Nootbook environment for SimpleITK. Execute it will bring up a web page hosted locally



Implementation

- The ‘root’ directory is the user’s home directory:



Conclusions

- Now we have a working python with a visualization environment
- Continued with the python tutorial…

Useful links

- Jupyter Notebook tutorial
  - https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook

Jupyter Notebook

- Jupyter Notebook (SITKPY) live demo

Questions/Comments

- Instructor
  - Elvis Chen, PhD, LEL
  - echen29@uwo.ca
- Teaching Assistant
  - Reid Francis Vassallo
  - rvassall@uwo.ca
  - 1hr/week, email for appointment