BIOMATH 208 Week 1

Aidan Jan

January 8, 2025

Logistics

Office Hours: Tuesdays 1-3pm

No midterm/final

Grading: 80% homework, 20% final project

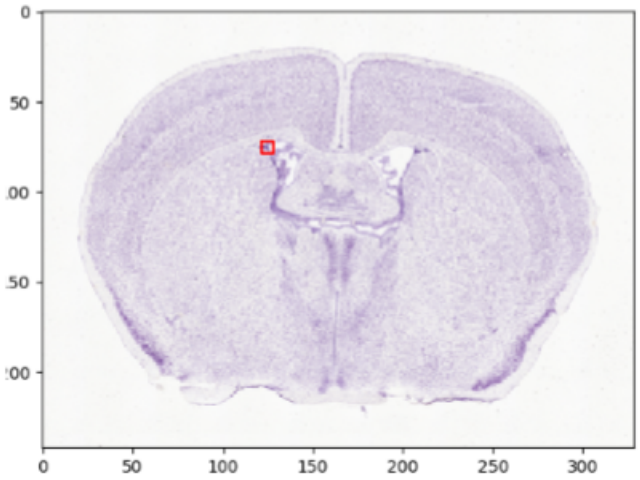
Overview

1. Representing and visualizing imaging data
    - Pixels, vectors, etc.
2. Multilinear Algebra
    - Linear algebra with stuff added on top
    - Two types of square matrices
3. Curves and surfaces
    - Comparing curves and surfaces using multilinear algebra
4. Manifolds
5. Transformation Groups
    - Understanding rotation, linear transformations
6. Tangent spaces
7. Optimization and image registration
    - Aligning multiple images of the same object
8. Metric manifolds
    - Finding distance between two rotation matrices, ellipsoids, probabilities, etc.
    - Using distances to compute averages, make predictions, etc.
9. Averaging filtering and regression

Representing images

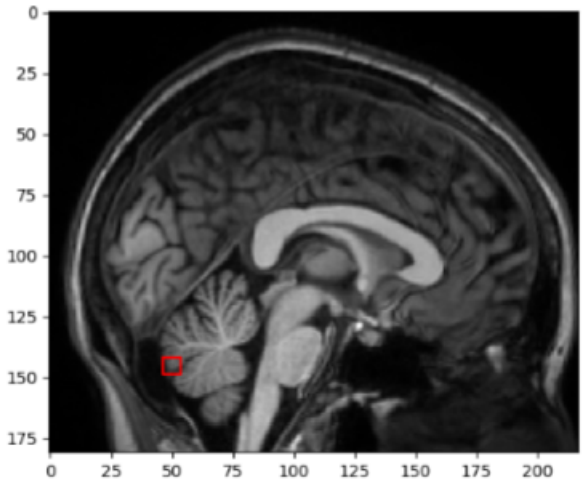
- Definition (images): We generally consider images as functions from some set $X \subseteq \mathbb{R}^d$, to some other set, S.
  - X describes space; usually, $d = 2$ or $d = 3$, depending on the dimension of the image.
- What this means:
  - Let X represent a picture, some 2D set of pixels in this example. Let $x \subset X$, where x is a pixel in the picture.
  - We can define the image, I, as a function, such that $X : X \to S$, where S is a pixel value.
    - $S \in \mathbb{R}$ if the image is grayscale, and $S \in R^3$ if the image is colored.

Example 1



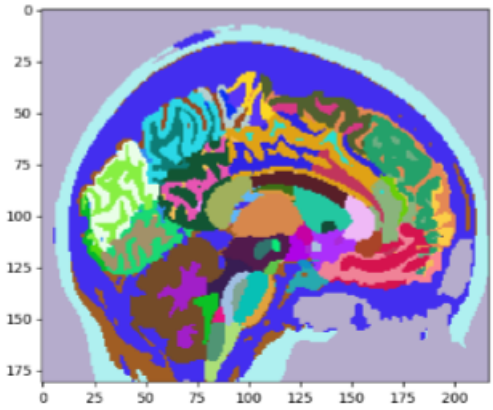
- Above is a microscopy image of a Nissl stained mouse brain.
- Here, $d = 2$ with X some rectangle, and $S = [0, 1]^3$ containing 3 RGB values, each value between 0 and 1.

Example 2



- Above is a part of a human brain MRI. The set X is shown on the axes.
- Here, $d = 3$, since this is a 3D image. $S = \mathbb{R}$ since it is grayscale, therefore only one value.
  - The S value is a value between $[0, 1]$ representing the brightness of the pixel.

Example 3 - Label Images

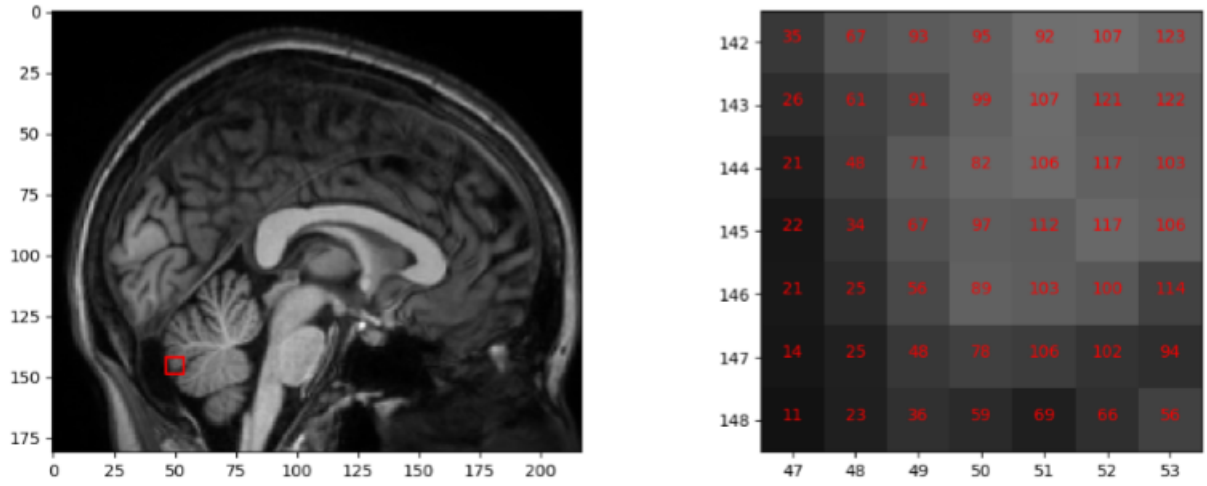


- Here, $d = 3$ with X some rectangular prism, and $S = \mathbb{N}$ containing 1 integer, `id`. Each `id` represents some particular type of structure.
- Part of a labeled brain image is shown. Integers are shown as colors, and the set X is shown on the axes for one slice of the 3D volume.

Discrete Images

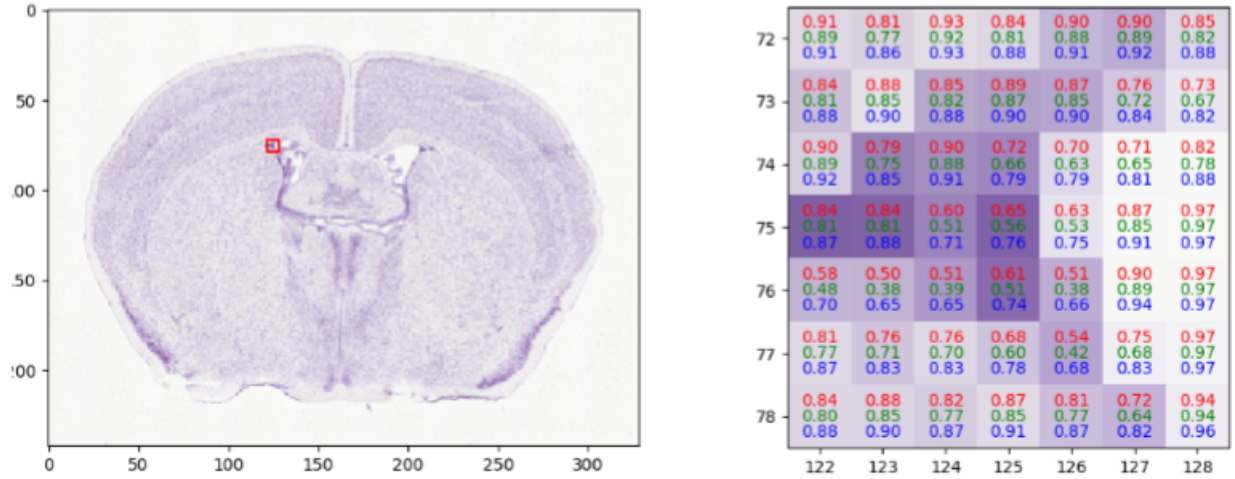
- Treating images as functions is nice, because they have basically infinite resolution. (You can plug in any coordinate value and receive the corresponding values.)
- However, we can't store the functions in a computer. As a result, we usually work with **discrete images**.
- In this case, images are d-dimensional arrays, storing values in S.
  - Geometry is stored with pixel size $\Delta \in \mathbb{R}^d$, and origin $O \in \mathbb{R}^d$.
  - Square brackets are used to index starting from 0. `img[0, 0]` would represent the top left corner.
  - In 3D, we would use rows, columns, and slices.
  - We may also specify the location of the origin, O, to scale in real life. Origin is `img[0, 0]`, but may be assigned some real world location, for example, [10ft, 20ft]. Δ is the pixel size, and therefore the physical location for any pixel can be calculated. (Real location of pixel `img[i,j]` would be $O + \Delta \cdot \binom{i}{j}$).

Grayscale images



- Here we see a sagittal view of a brain MR image, and a zoom in showing pixel values.
- Here, the origin is (0, 0), and the spacing is (1, 1) (in mm).

Color Histology



- Here we see an image, and a zoom in showing pixel values as RGB triples.
- Here the origin is (0, 0), and the spacing is (58.8, 58.8) (in microns).

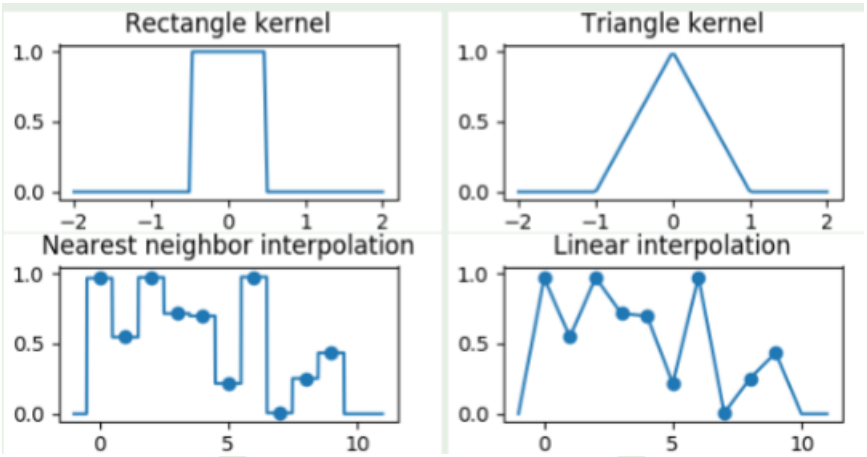
Interpolation

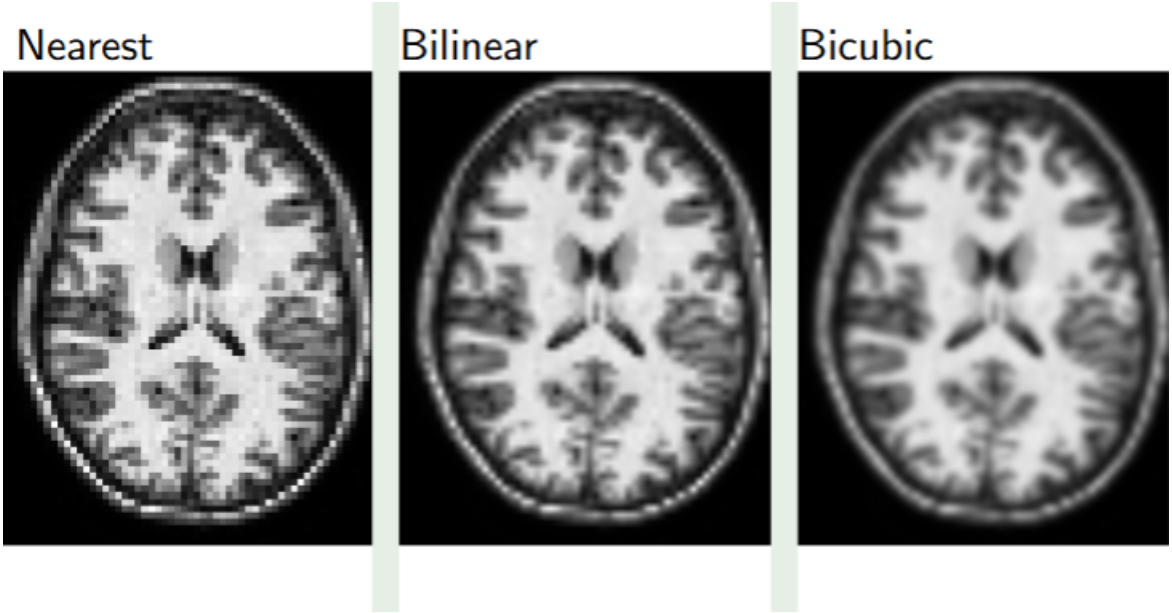
Interpolation links images as functions to images as discrete arrays. If S is a vector space, then we can specify an interpolation kernel h and write

$$I(x) = \sum_{i,j,k} I[i,j,k]h(x - x[i,j,k])$$

where $x[i,j,k] = O + \text{diag}(\Delta) \begin{pmatrix} i \\ j \\ k \end{pmatrix}$, the coordinate of the i,j,k-th pixel.

- $I(x)$ is the image function, and $I[i, j, k]$ is the array of pixels. h is often a function that is close to 1 near the origin and decays down to zero as you get far away.
  - $h()$ is also known as a point-spread function (psf), or an impulse response function.
- What the function basically does is that it multiplies the pixel value by the impulse response function at that real-world location.
- We are convolving the kernels with the discrete pixel dataset. Triangle kernel gives linear interpolation, and rectangle kernel gives nearest neighbor interpolation.



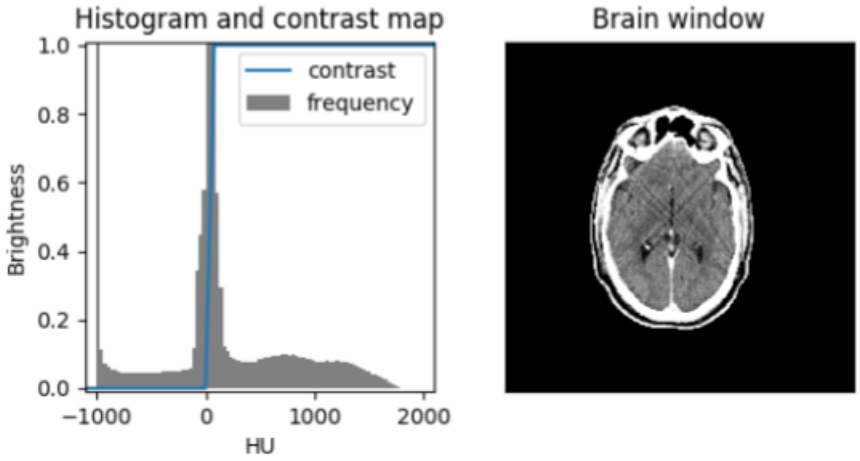


- The image on the left is a brain interpolated (convolved) using the rectangle kernel
- The middle image is one interpolated with the triangle kernel
- The right image is a brain interpolated with a piecewise cubic function (aka. third degree spline).

Importantly, the range of values the function takes is the same as the values. Since the kernel function is always between values 0-1, the minimum value of the interpolated image is 0, and the maximum is the highest original pixel value.

2D grayscale images

- To display images, intensity values at each pixel need to be mapped to a brightness value on the screen (a number between 0 and 1)
- A piecewise linear function is used to do this, by selecting a lower value for black, and an upper value for white.
- A histogram of pixel values can be used to choose these parameters.



- The histogram is one that shows pixel values. The center should be set at the highest peak of the contrast map, e.g., setting brightness and contrast.
- The brightness value is essentially "which pixel value should be mapped to black", and contrast is the "difference between white pixel and black pixel" or, "what pixel value is white relative to black".
- The above image sets black to 0 HU, and white to 80 HU. Anything below the range is black, anything above is white.

Summary so far:

- Why study medical images from a geometric point of view?
  - Different ways to look at images of the same thing
  - Looking at the pixel values as a list of numbers is not ideal
  - Lists of numbers will be called "coordinates"; we want to study in a way that is independent of choice of coordinates
  - Differential geometry studies this problem.
- Images as functions: $I : X \to S$, where
  - X is some subset of space, e.g., inside the scanner
  - S are the values that pixels might take
    - $\mathbb{R} \to$ grayscale images
    - $[0, 1]^2 \to$ colored images
- Discrete images
  - Two parts:
    - An array that stores a set of values ($I[i, j, k]$, if $X \subseteq \mathbb{R}^3$)

    - An origin and pixel size, O, Δ. $x[i,j,k] = \begin{pmatrix} O^0 + i\Delta^0 \\ O^1 + i\Delta^1 \\ O^2 + i\Delta^2 \end{pmatrix}$

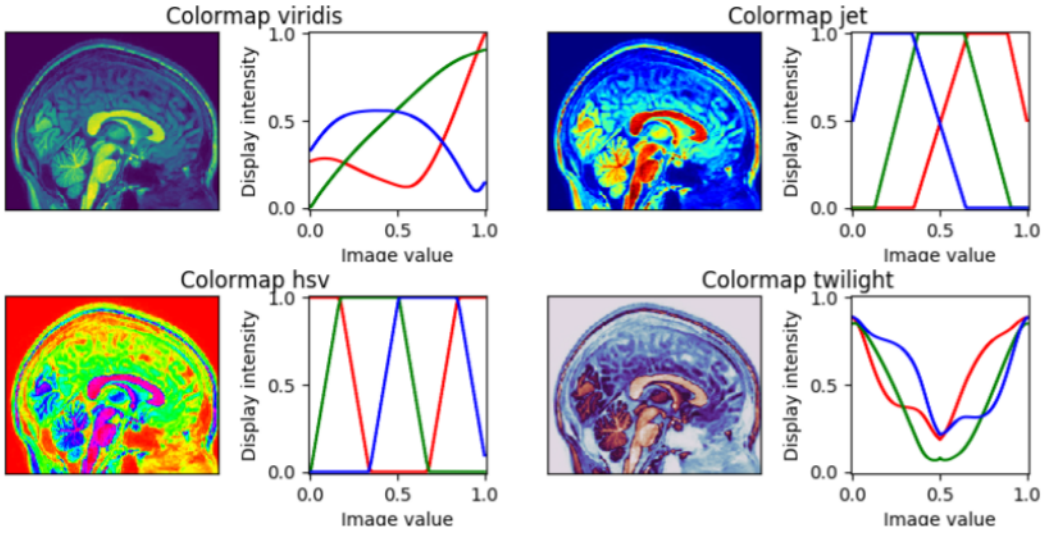
- Interpolation

$$I(a) = \sum_{i,j,k} \cdot h(x - x[i,j,k])$$

  - h is the interpolation kernel.
  - If h is a rectangle $\rightarrow$ nearest neighbbor interpolation
  - If h is a triangle $\rightarrow$ linear interpolation

- Contrast Curve
  - aka. contrast and brightness
  - aka. window and level
  - A function that maps pixel values to the power we give the LED on the screen
    - Function used usually ranges from 0 to 1, and is monotonically increasing
    - v_{min} is the pixel value such that anything less will be black
    - v_{max} is the pixel value such that anything more will be white
  - In CT scans, we can use a "brain window", which is very narrow
    - v_{min} will be slightly lower than the attenuation for water
    - v_{max} will be slightly higher than the attenuation for water
  - We also have "bone window", where v_{min} is closer to the value of air and v_{max} is a value larger than the attenuation for bone.

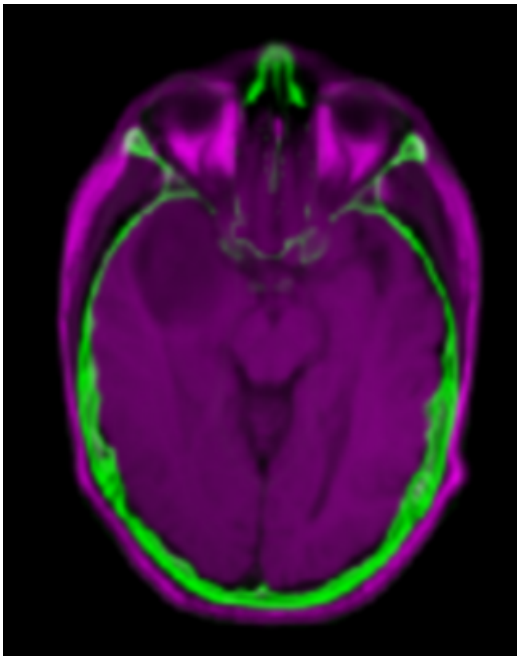
Color Mapping



- Grayscale images can be mapped to colors to emphasize certain details.
- We chose 3 functions from [0, 1] $\rightarrow$ [0, 1], one for red, green, and blue.
- Colormap Jet is not perpectually uniform, and it is used for highlighting. Often, Red = Bad, Blue = Good/Normal

- Color HSV is a "cyclic" colormap. For this one, only Hue changes, while Saturation and Value stay constant.
- Colormap Twilight is a "diverging" colormap. For this one, the "normal" is at 0.5, and the colors at that point are not altered. Since lower values are mapped to red and higher values to blue, it shows the direction and magnitude of differences compared to the median.

Images with a signal value that is more than one dimension can be rendered as a color image by mapping different dimensions to the red, green, or blue channel. The most common such type is a RGB image, which stores color values directly, but there are other possibilities. For example, it is sometimes useful to display multiple images on top of one another, to see how signals co-localize in different images. One image can be set to green, and one to magenta, as shown in the figure below



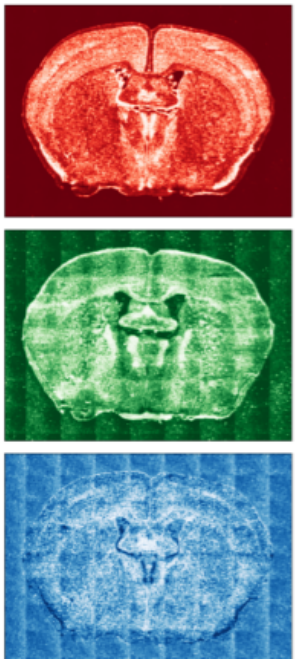
- This type of image is used to show complementary data.
- This image in particular maps the MRI scan to red + blue, and CT scan is green.

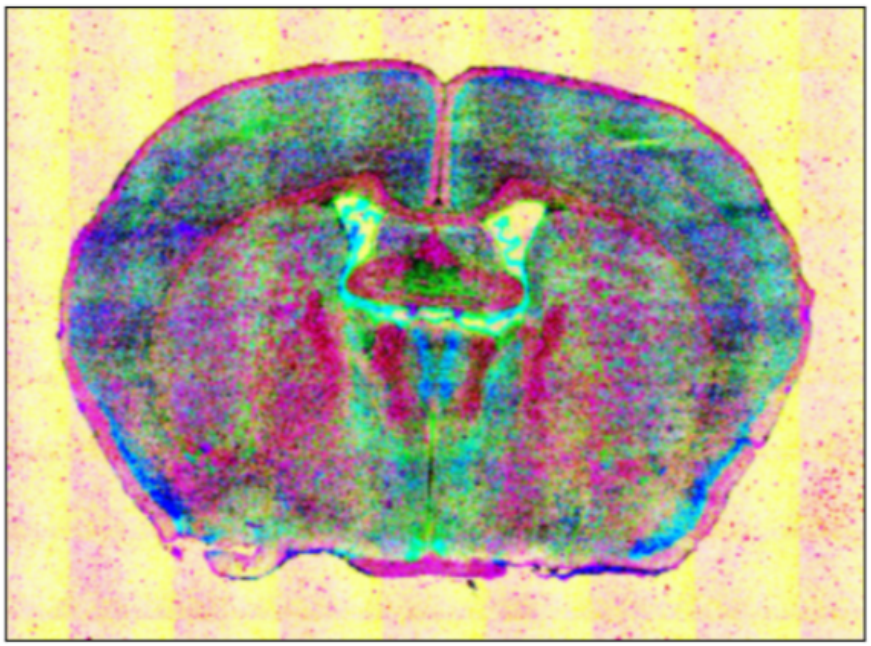
Code for Multimodal Images

```
import numpy as np
import matplotlib.pyplot as plt
# ...
# I is a 2D array storing grayscale MR data,
# normalized to the range [0,1]
# J is a 2D array storing grayscale CT data,
# normalized to the range [0,1]
f,ax = plt.subplots()
RGB = np.concatenate((I,J,I), axis=-1)
ax.imshow(RGB)
```

Feature Maps

- Feature maps such as texture classifiers can be mapped to RGB channels to visualize high dimensional data.

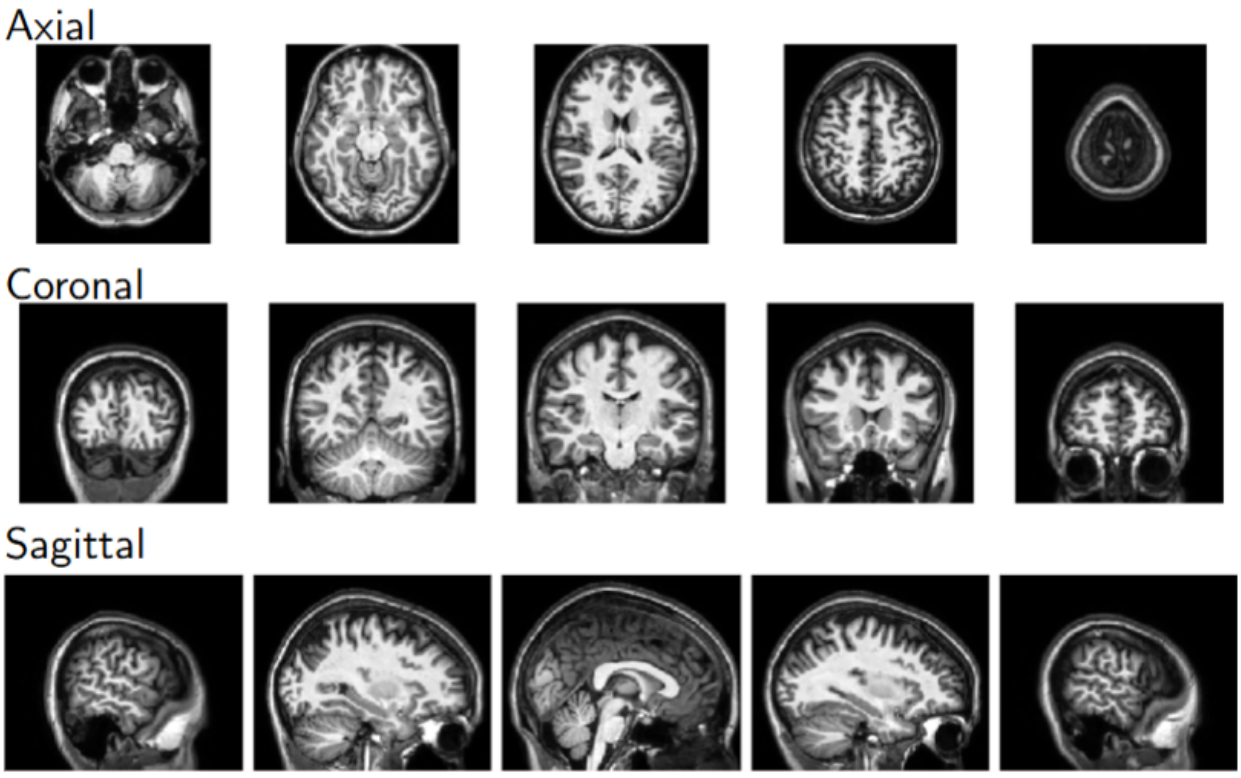




3D Images

- Unlike natural images, most medical imaging data is 3D.
- In the absense of virtual reality platforms, these must be represented in 2D.

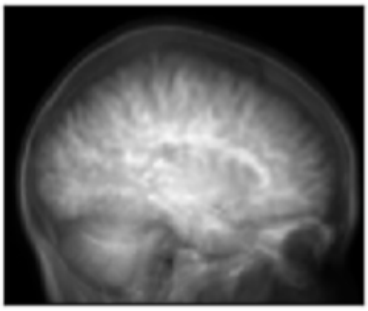
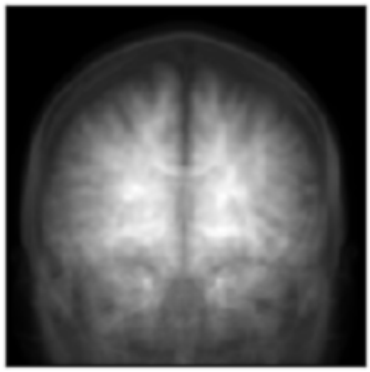
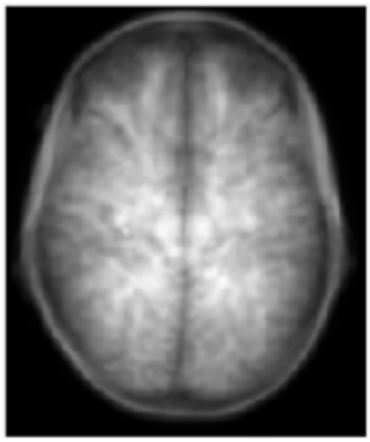
Slice directions



- Coronal plane is an image taken from the front/back.
- Sagittal plane is an image taken from the left/right side.
- Axial plane is an image taken from the top/bottom.

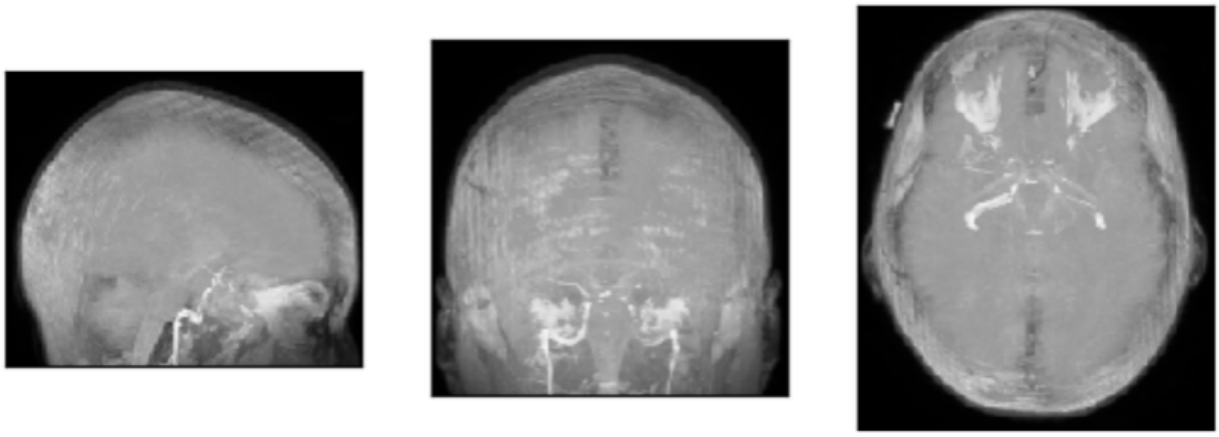
Linear Projections

- Let $J(x, y) = \int I(x, y, z)\mathrm{d}z$
- In Python, this can be implemented as `J = np.sum(I, axis=2)`.
- We are essentially summing all the pixels on one axis. We can generate three projections since there are three axes we can sum over.
- Linear projections can simulate a radiograph

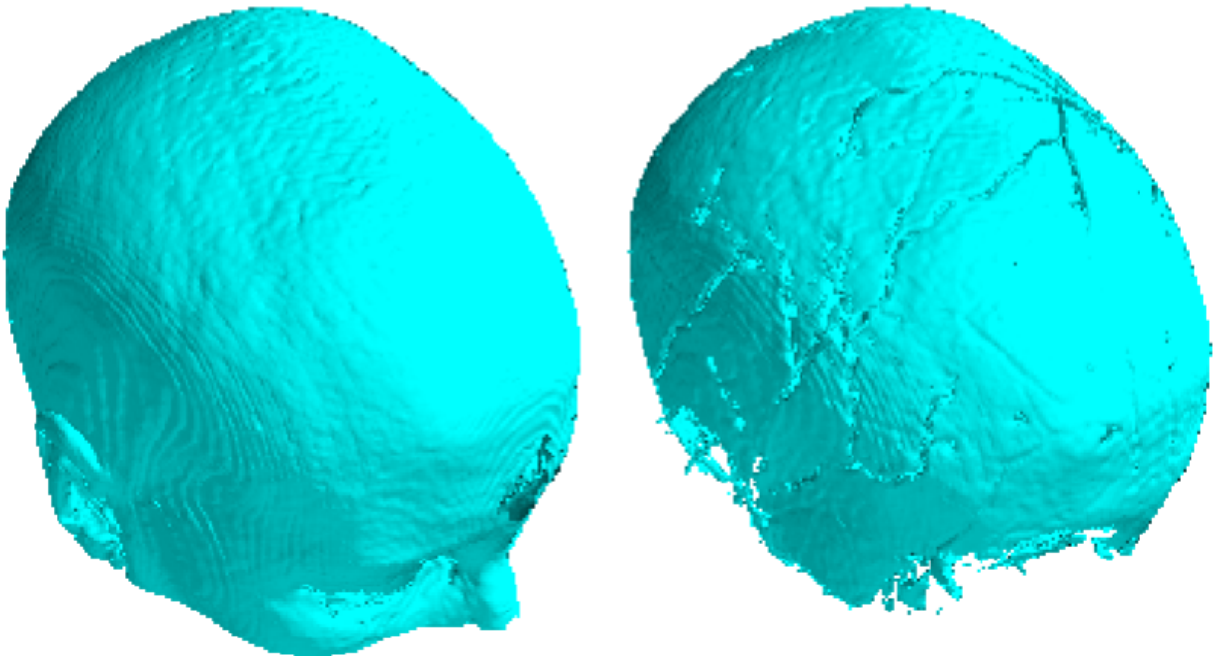
Maximum Intensity Projection

- Let $J(x, y) = \mathrm{Max}I(x, y, z)$
- In Python, `J = np.max(I, axis=2)`.
- Here, we take the maximum of each pixel over an axis. Like linear projections, we can create three projections since there are three axes we can sum over.
- MIPs are often useful for thin structures like vessels.
  - In a slice, vessels look like a dot.
  - In a linear projection, vessels are obscured by everything else.
  - Same goes with neuron axons, etc.



Isosurfaces

- By selecting a threshold, we can draw a surface that separates bright regions from dark regions. Below, two thresholds are shown.



- This is like a topography image. In topography, changes in altitude on a mountain is viewed as circles. In an isosurface, we are doing this in 3D. The boundary created is a difference in brightness. As such, one side of the boundary is brighter than the threshold, the other side is darker.

Isosurface Code

- Matpoltlib is not good at rendering 3D data, like isosurfaces. One alternative in Python is mayavi.

```
from mayavi import mlab
# ...
# I is a 3D array storing grayscale imaging data
mlab.init_notebook()
f = mlab.figure(bgcolor=(1.0,1.0,1.0),fgcolor=(0.0,0.0,0.0))
mlab.contour3d(I,contours=[20.0], color=(0.0,1.0,1.0))
```

- The best way to explore 3D data is with interactive software.
- Examples:
  - ITKsnap
  - Paraview
  - Imaris