hhmi | janelia
Research Campus

John Bogovic

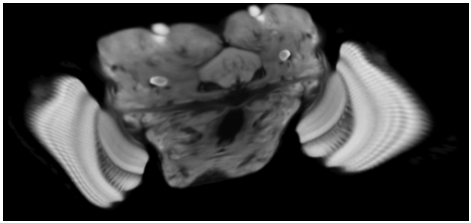
Advanced registration of big image data

Outline

Image Registration

- Transformations
- Image Geometry
- Practical algorithmic details
  - choosing parameters
  - common pitfalls
  - big data strategies

Image registration



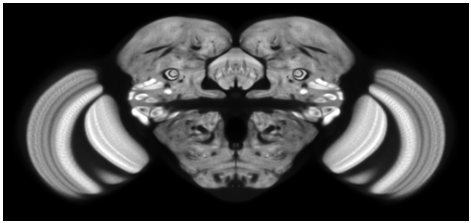
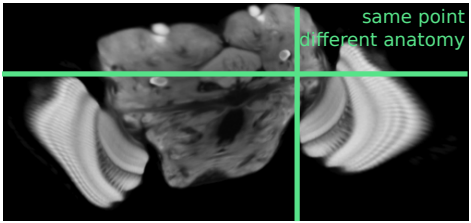


Image registration



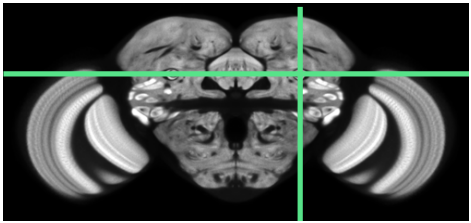
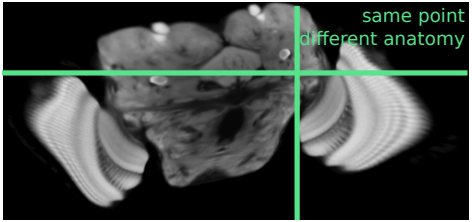
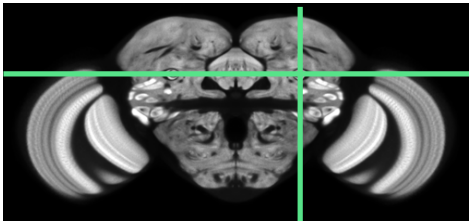


Image registration





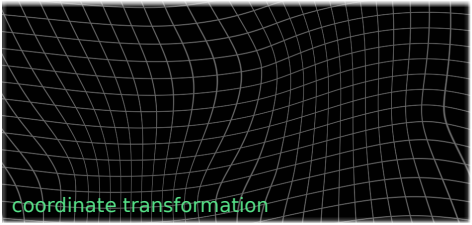
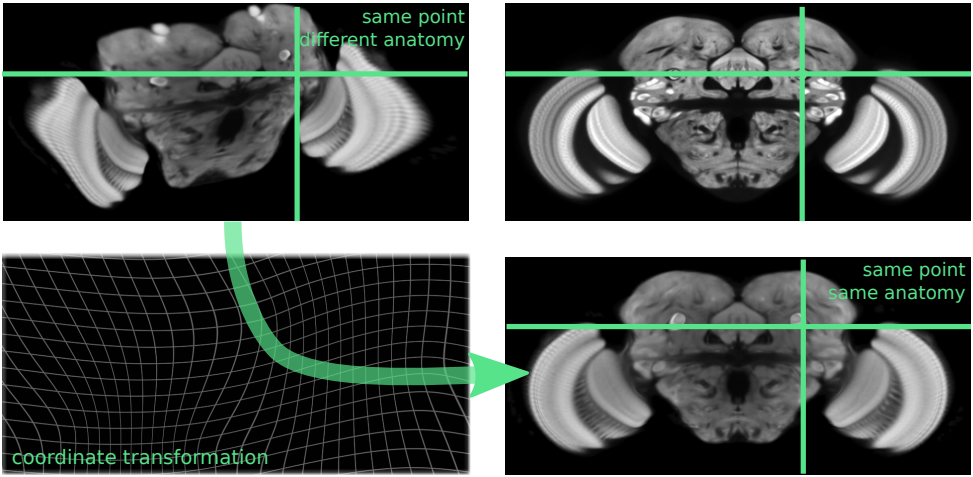


Image registration

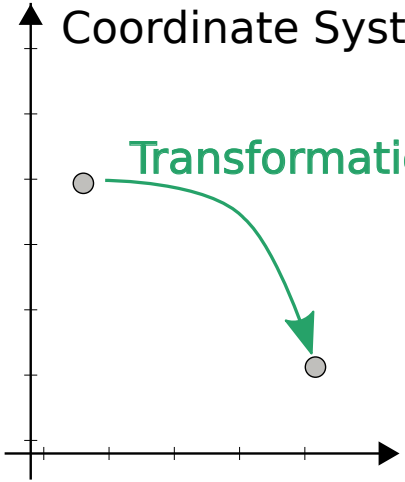


Key idea

The primary output of an image registration algorithm is the *transformation*.

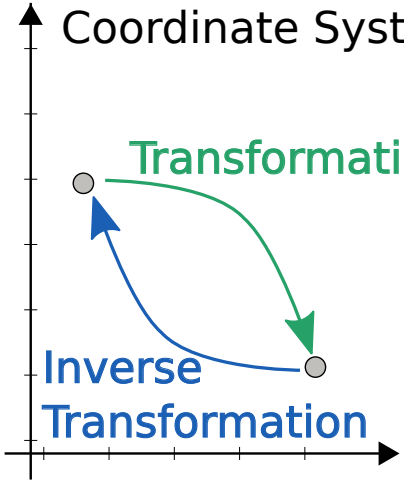
A secondary output of an image registration algorithm is the *transformed moving image*.

What is a transformation



Transformations are functions that:
take points (coordinates) as inputs
produce points (coordinates) as outputs.

What is a transformation



Transformations are functions that:
take points (coordinates) as inputs
produce points (coordinates) as outputs.

The inverse is a transformation that:
given the output of a transformation,
produces the original input point
(un-does the original transformation).

Quiz

If we have the transformation:

$$T(x, y) = (x + 5, y - 12)$$

what is the inverse transformation?

Quiz

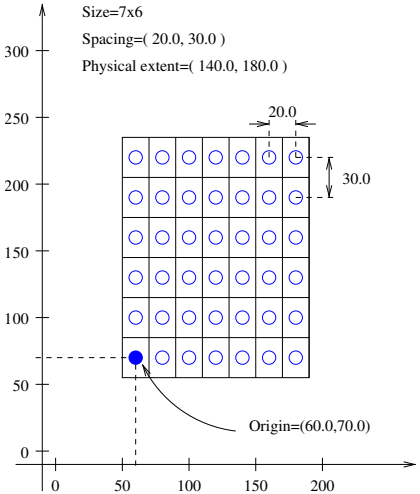
If we have the transformation:

$$T(x, y) = (x + 5, y - 12)$$

what is the inverse transformation?

$$T^{-1}(x, y) = (x - 5, y + 12)$$

Image geometry



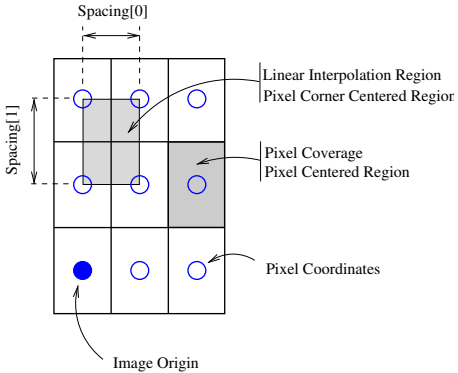
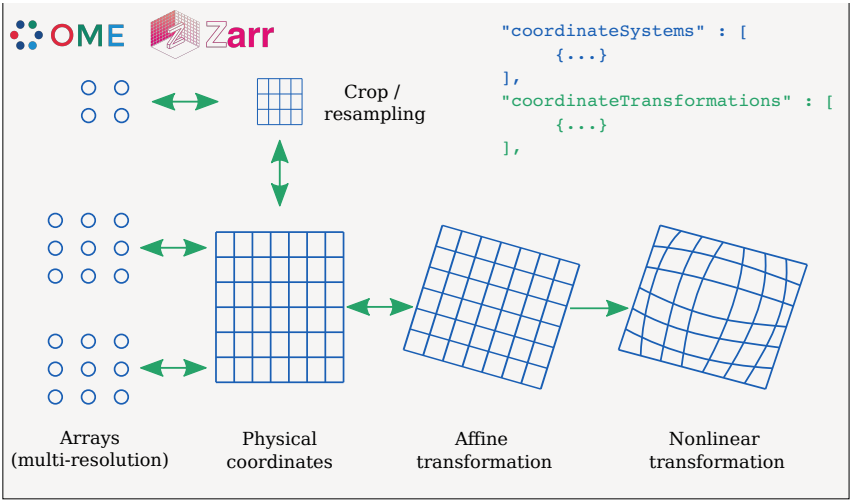


image from ITK Software Guide

Many relationships are transformations



Join the conversation at `github.com/ome/ngff`

Key ideas

Pixels (entries in the image array) are *point samples*.

Pixels are located in *physical coordinates*

Reinforce these ideas in `ResampleImages.ipynb`

point and image transformations

Define:

- The *forward* transformation maps *points* from moving to target space.
- The *inverse* transformation maps *points* from target to moving space.

point and image transformations

Define:

- The *forward* transformation maps *points* from moving to target space.
- The *inverse* transformation maps *points* from target to moving space.

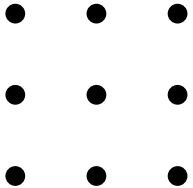
Then:

- We need the *inverse* transformation to map *images* from moving to target space
- We need the *forward* transformation to map *images* from target to moving space

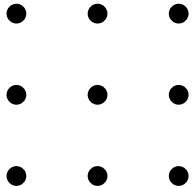
The following will explain why this is the case.

Transforms on a grid

Moving image grid

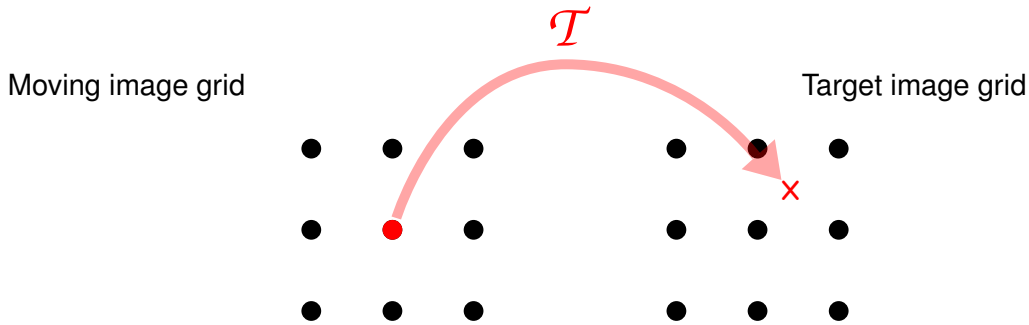


Target image grid



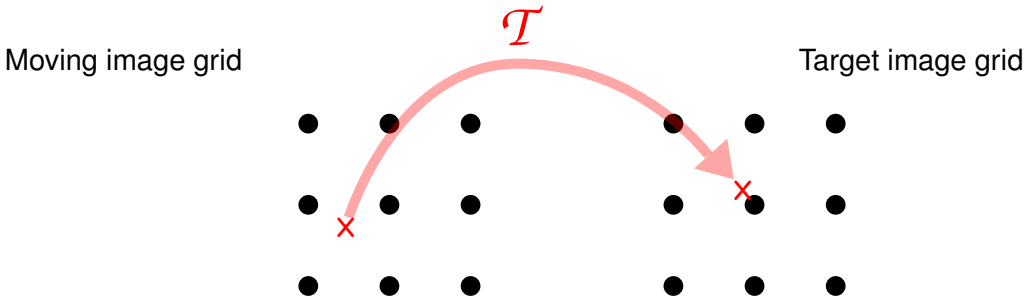
We have a moving image that we want to transform, and render on a new, target image grid.

Transforms on a grid



A transformation maps points from the moving space to the target space. But moving grid points may not land on the target grid.

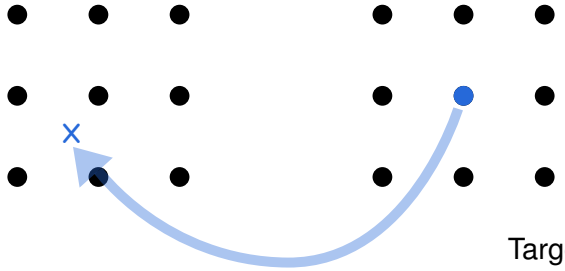
Transforms on a grid



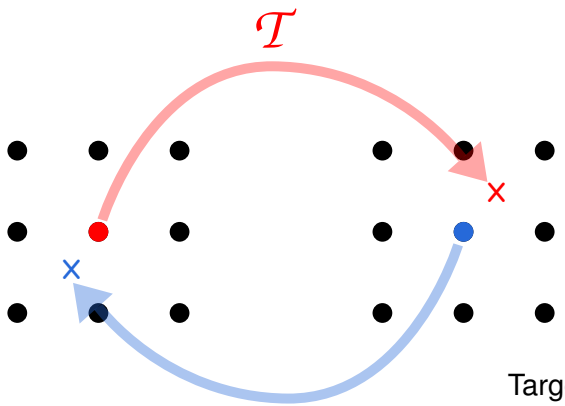
How do we find a point in moving space (possibly between grid points) that maps to a grid point?

Transforms on a grid

Applying the inverse of the transformation to grid points in target space tells us where to sample from in moving space.



Transforms on a grid



Moving image grid

Target image grid

point and image transformations

Define:

- The *forward* transformation maps *points* from moving to target space.
- The *inverse* transformation maps *points* from target to moving space.

Then:

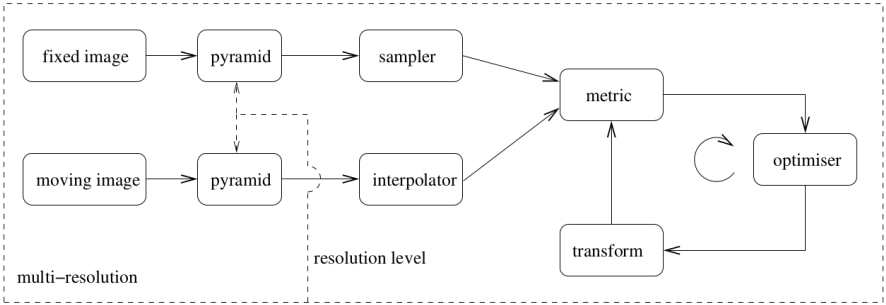
- We need the *inverse* transformation to map *images* from moving to target space
- We need the *forward* transformation to map *images* from target to moving space

Remember this when working through `RecoverOffset.ipynb`

“Pixel-based” Image Registration Algorithms

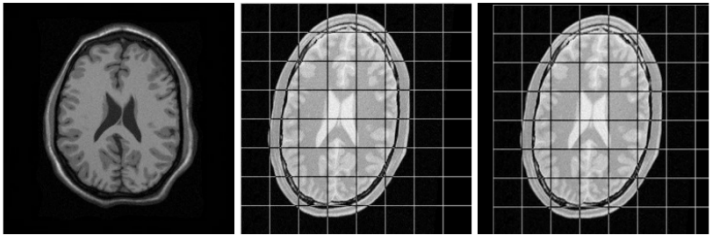
“Feature-based” registration methods will not be covered in this course.

Image registration pipeline



basic registration components

Transformation types



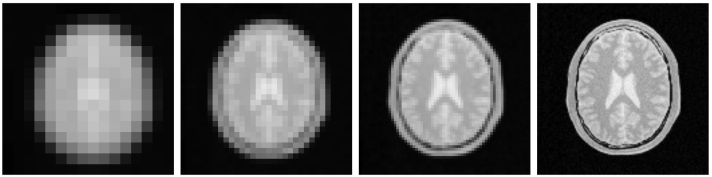
(a) fixed (b) moving (c) translation

(d) rigid (e) affine (f) B-spline

image from elastix manual

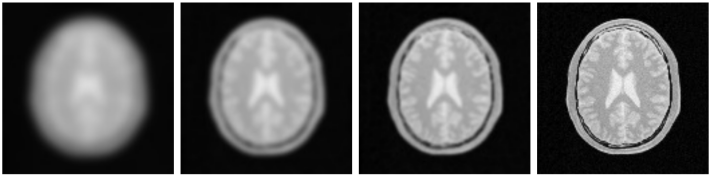
Multi-resolution registration

Smooth and downsample



(a) resolution 0 (b) resolution 1 (c) resolution 2 (d) original

Smooth only



(e) resolution 0 (f) resolution 1 (g) resolution 2 (h) original

image from elastix manual

Optimization

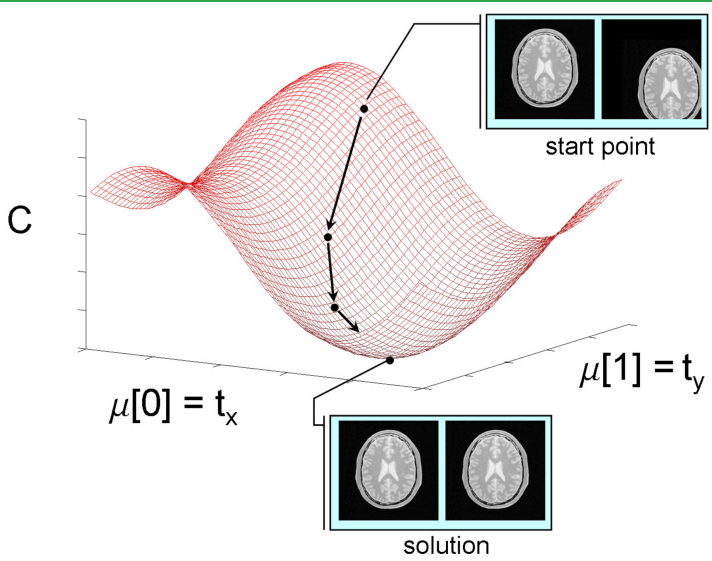


image from elastix manual

Elastix parameters

```
// ********  Image properties ********

(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(FixedImageDimension 3)
(MovingImageDimension 3)

// ************* Multiresolution ***************

(Registration "MultiResolutionRegistration")
(NumberOfResolutions 4)
(FixedImagePyramid "FixedRecursiveImagePyramid")
(MovingImagePyramid "MovingRecursiveImagePyramid")

// The downsampling/blurring factors for the image pyramids.
// There are defaults, but can be provided manually
// (ImagePyramidSchedule 8 8 8  4 4 4  2 2 2  1 1 1 )

// ******** Interpolation and Resampling ***********

(Interpolator "BSplineInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")

(BSplineInterpolationOrder 1)
(FinalBSplineInterpolationOrder 3)
(DefaultPixelValue 0)
```

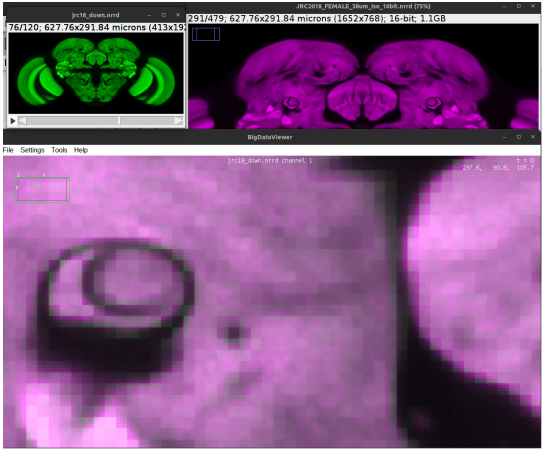
```
// ************* Transformation ****************

(Transform "AffineTransform")

// ************ Similarity metric **************

(Metric "AdvancedMattesMutualInformation")

// Some metrics have additional options
(NumberOfHistogramBins 32)
(NumberOfSpatialSamples 2048)
(NewSamplesEveryIteration "true")
(ImageSampler "Random")

// ************ Optimization **************

(Optimizer "AdaptiveStochasticGradientDescent")
(MaximumStepLength 0.5)

(MaximumNumberOfIterations 1000)

// or can have number of iterations per resolution
// (MaximumNumberOfIterations 500 250 100 50 )

// ************* Output options ****************

(WriteResultImage "false")
(ResultImagePixelType "float")
(ResultImageFormat "nrrd")
```

We will learn

- Reinforce fundamental concepts
- How to choose an appropriate metric
- Some properties registration algorithms
  - the importance of initialization
  - non-linear registration
- Strategies for big data
  - run registration at low resolution
  - apply transformation at full resolution

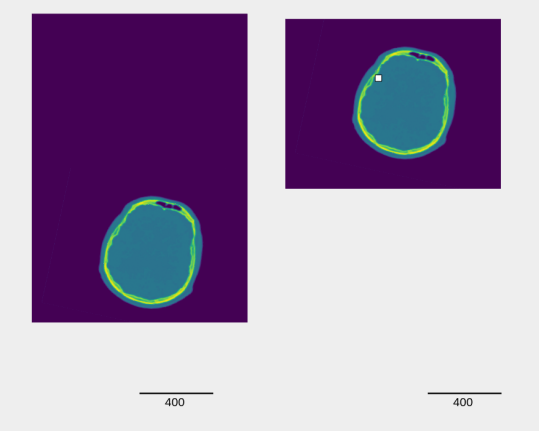
Practical outline

- `resample_images.ipynb`
  - Images have a physical coordinate system
  - Downsampling strategies might affect physical coordinates
- `translation_initialization.ipynb`
- `similarity_metrics.ipynb`
- `multi_resolution.ipynb`
- `nonlinear.ipynb`



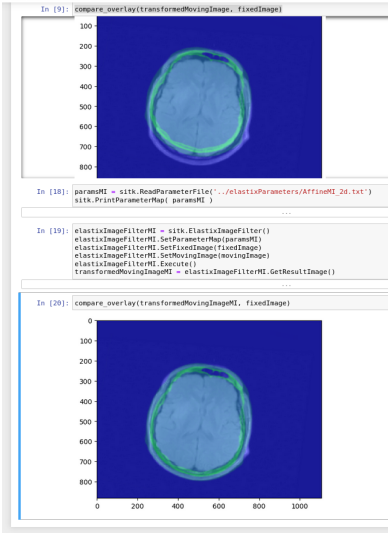
Practical outline

- `resample_images.ipynb`
- `translation_initialization.ipynb`
  - Simple image registration (translation)
  - Initialization can be important
- `similarity_metrics.ipynb`
- `multi_resolution.ipynb`
- `nonlinear.ipynb`



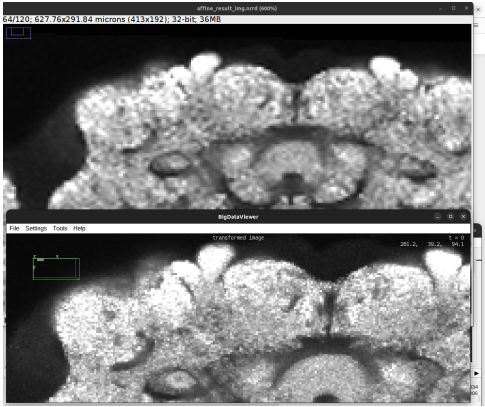
Practical outline

- `resample_images.ipynb`
- `translation_initialization.ipynb`
- `similarity_metrics.ipynb`
  - Choosing an image similarity metric
- `multi_resolution.ipynb`
- `nonlinear.ipynb`



Practical outline

- `resample_images.ipynb`
- `translation_initialization.ipynb`
- `similarity_metrics.ipynb`
- `multi_resolution.ipynb`
  - Big data strategies
- `nonlinear.ipynb`



Practical outline

- `resample_images.ipynb`
- `translation_initialization.ipynb`
- `similarity_metrics.ipynb`
- `multi_resolution.ipynb`
- `nonlinear.ipynb`
  - Advanced, non-linear image registration
  - Measuring smoothness