Department of Computing

Imperial College of Science, Technology and Medicine

Image Registration

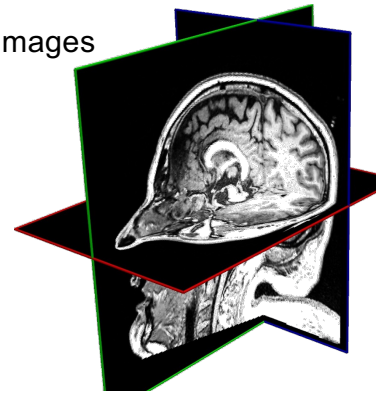
This week's lecture contains content about image registration and how we can 'combine' two images of the same object together with transformations. This week's paper recommendation is [1]

Author: Anton Zhitomirsky

Contents

1 Image Registration

- d-dimensional arrays with scalar/vector values
- Mathematical functions: $f: \mathbb{R}^d \rightarrow \mathbb{R}^n$
  - Often, we have three-dimensional, scalar-valued images $d = 3, n = 1, f(x, y, z) \in \mathbb{R}$
- Meta information
  - Scale: element spacing (e.g. in mm)
  - Orientation: directions of main axes
  - Position: image origin

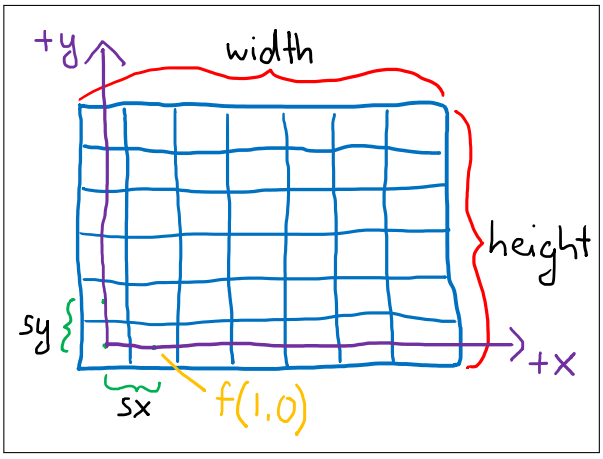


We typically don't know the analytical value of f, however, we have it in a sampled manner.

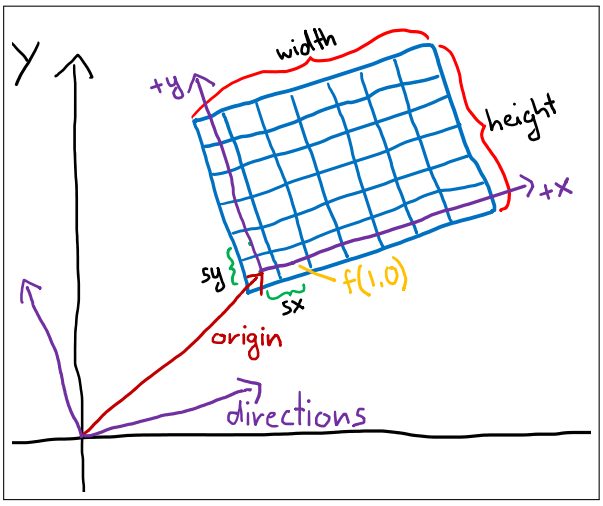
If we have an image of a patient, with two different imaging modalities to scan the same plane, we require image registration to see how does a pixel look like in another image. This requires a spatial transformation between the two images.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{where } T \text{ is the transform operation from one image to another}$$

1.1 Coordinate systems and transformations



(a) Image coordinate system



(b) Word coordinate system

In an image coordinate system in Figure 1(a) we have a pixel grid of a set width and height. Each pixel has a certain physical dimension, which is obtained form the measurement device. This assumes that the x and y axis is aligned with the pixel grid lines. Typically this is not the case.

Therefore, we introduce the real-world coordinate system in Figure 1(b). Here, it is not necessarily aligned with the x and y, and the origin is not necessarily at (0,0). (We don't really worry when it comes to segmentation but when we relate images together then it becomes as concern because they

may be scanned at different transformations). In cartesian space, you worry about the orthogonal unit vectors defining the directions and the origin.

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \underbrace{\overbrace{\begin{bmatrix} 1 & 0 & ox \\ 0 & 1 & oy \\ 0 & 0 & 1 \end{bmatrix}}^{\text{translation vectors}} \overbrace{\begin{bmatrix} dxx & dyx & 0 \\ dxy & dyy & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{direction vectors}} \overbrace{\begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{scaling matrix}}}_{T_{ItW}} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

This **affine** transformation equation uses homogeneous coordinates[1]. We write them in a 3D form, because we can model interesting geometric rtasnformations such as translations.

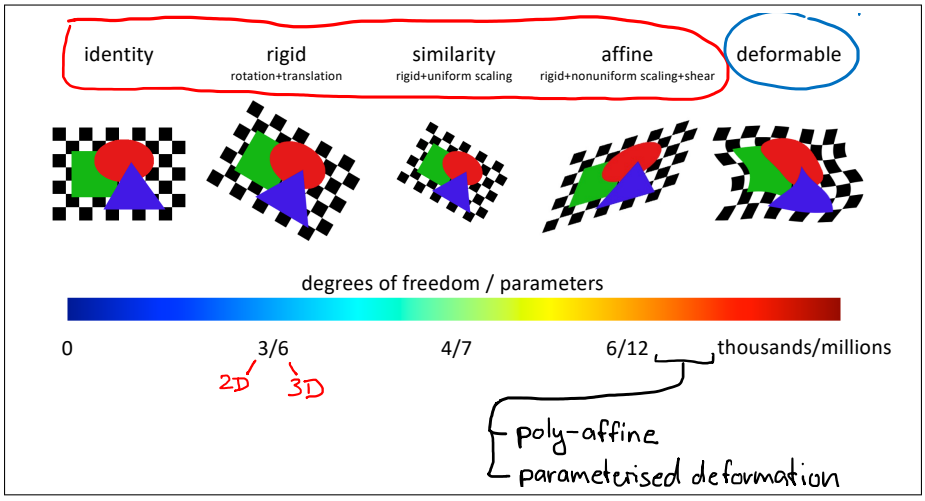
- **translation vectors**: translation from the origin of the world coordinates to the origin of the image coordinates
- **direction vectors**: define the directions on the axis of the image
- **scaling matrix**: defines the pixel spacing to produce a scaled version of the image or the image coordinates in the X and Y direction.

Therefore, if we have more than one world, A and B and we wish to align them, first by aligning the world coordinates then doing the remaining transformations.

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = T^A_{WtI} T_{BtA} T^B_{ItW} \begin{pmatrix} x_B \\ y_B \end{pmatrix}$$

Image transformation is about finding T_{BtA}, since the other transformations can be obtained from the measurement devices.

2 Transformation Models



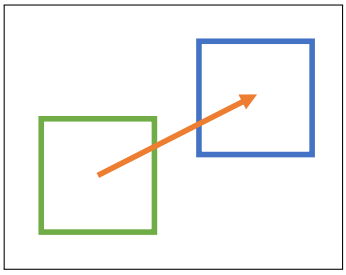
2.1 Linear Transformations

2.1.1 Rigid

Allows for rotation and translation. It is useful in the real world because it can't scale, mimmiking real-world; distances are always maintained.

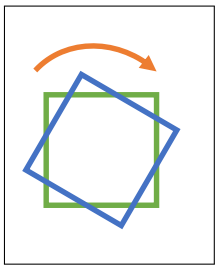
[1] TODO

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Translation

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

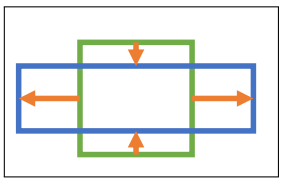


Rotation

2.1.2 Similarity

Allows for scaling, which is a similarity transform

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



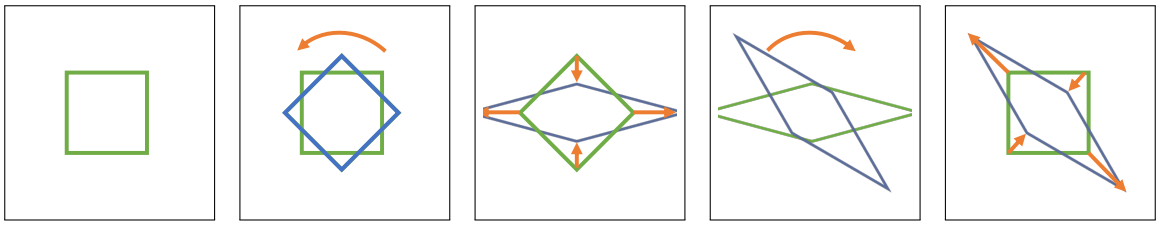
Shearing

2.1.3 Affine

Allows for scaling in different dimensions, and shears. They allow for scaling in different directions. No matter what affine transformation you do, lines will always be parallel.

Shearing is achieved by doing a combination of different transformation matricies, specifically RSR^{-1}; a rotation, scaling and rotation.

$$S = \begin{bmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Shearing

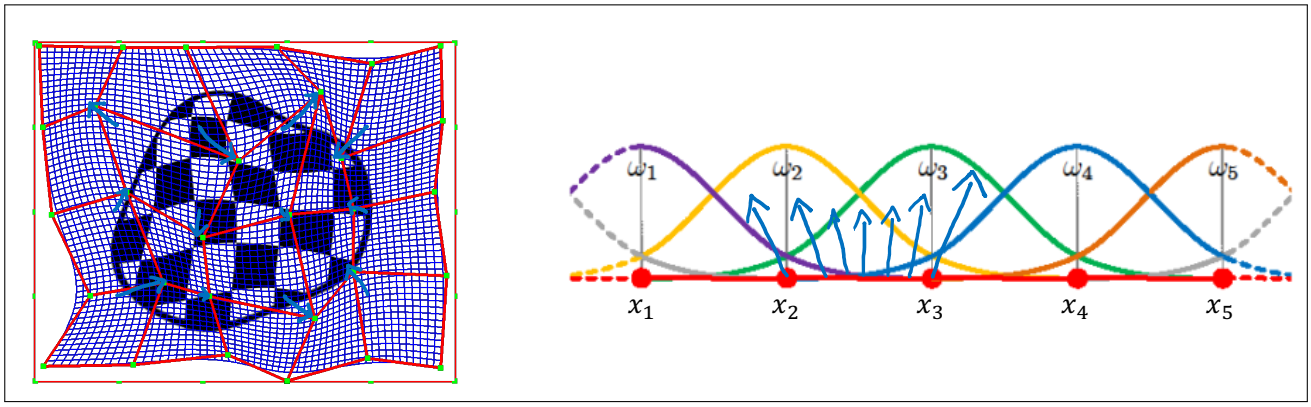
2.1.4 Combinations of Linear Transformations

The nice thing about linear transformations is that you can combine them all together into one big matrix and then directly operate on the matrix coeffiecients.

2.2 Non-Linear (Free-form) Transformations

2.2.1 Control Point Based Model

Free-form transformations; lines can cross (in topology preserving transformations this cannot happen)



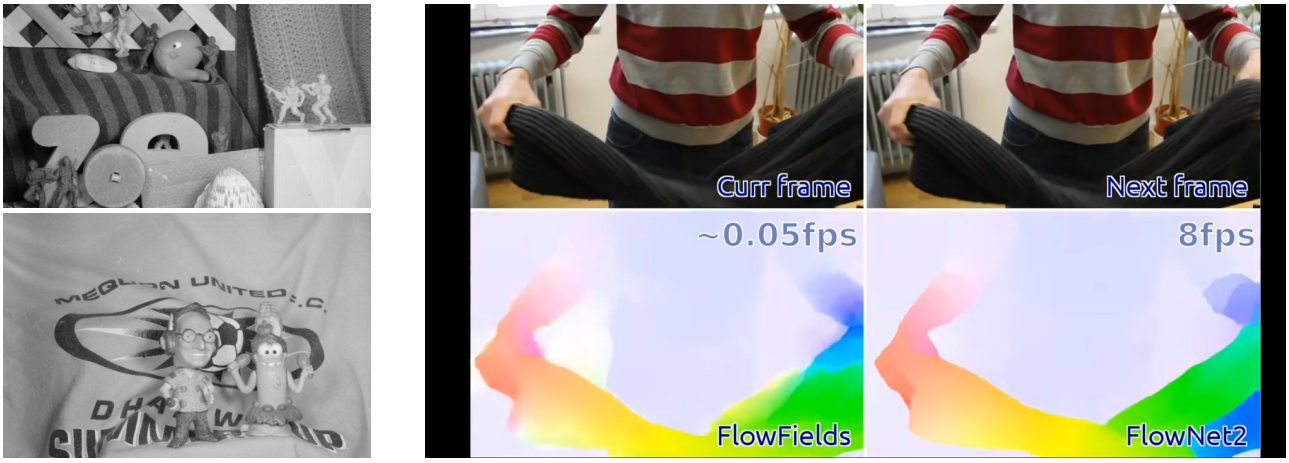
1. We can embed an image onto a grid (not a pixel grid)
2. We prescribe the motion/deformation at each grid point. These are control points. The blue arrows prescribe how the trainsformation at that point should behave in a smooth fashion.
3. The underlying blue grid comes out smooth despite ugly red grid.
4. This is because we start interpolating between the deformation values to produce a smooth deformation.
5. This model is based on linear-interpolation, where the (triangle) functions have the directions applied to them, and the deformations are applied linearly.

There are also finite-element methods, and dense displacement fields (we assign a direction for each pixel).

2.3 Applications

(left) Satelite Imaging (right) Point Correspondences

Panoramic Image Stitching



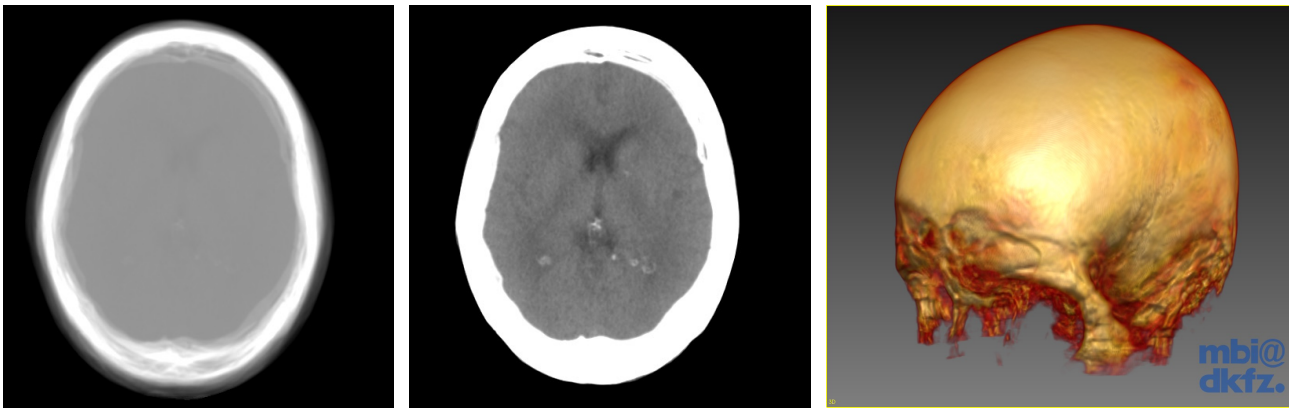
Optical Flow (3D reconstructions)

2.4 Medical Applications

- Cardiac Motion tracking
- Respiration Motion tracking
- Multi-modal Image Fusion
- Pre- and Post-op comparison

2.4.1 Intra-subject Registration

Perform Atlas construction: assuming you have different picutres of the human brain, you would like to find out what the average brain looks like, by integrating and averaging all images together.

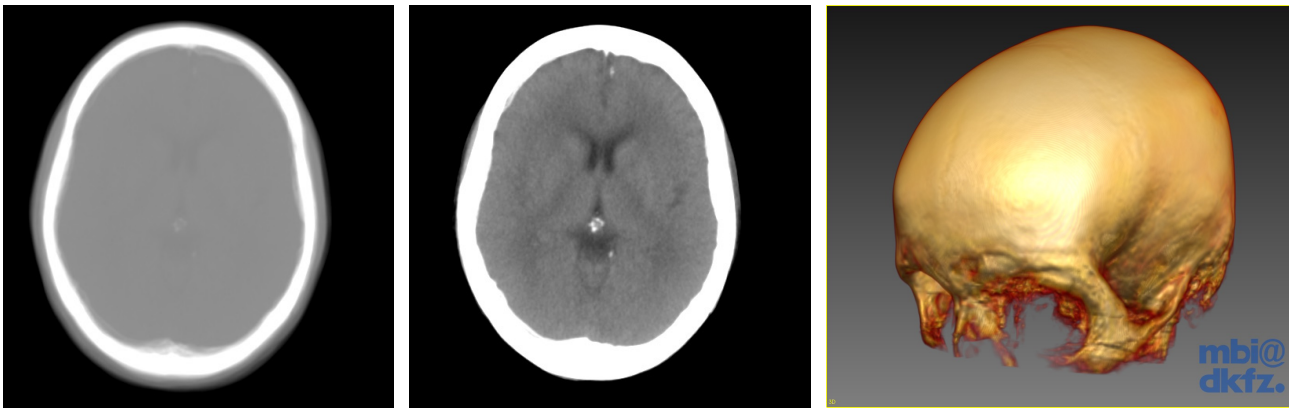


average image
full range

average image
bone window

average image
volume rendering

Atlas Iterations: Step 1 — Rigid

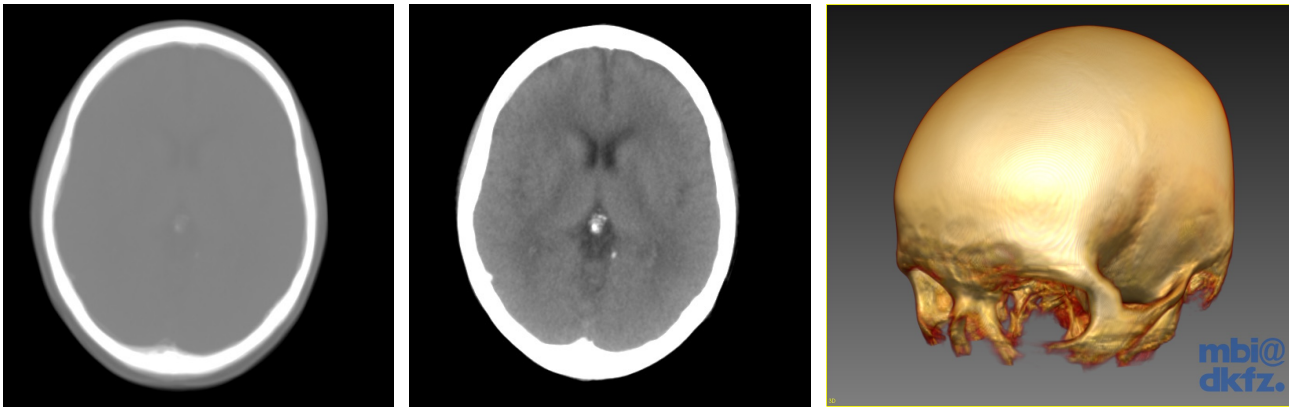


average image
full range

average image
bone window

average image
volume rendering

Atlas Iterations: Step 2 — Affine



average image
full range

average image
bone window

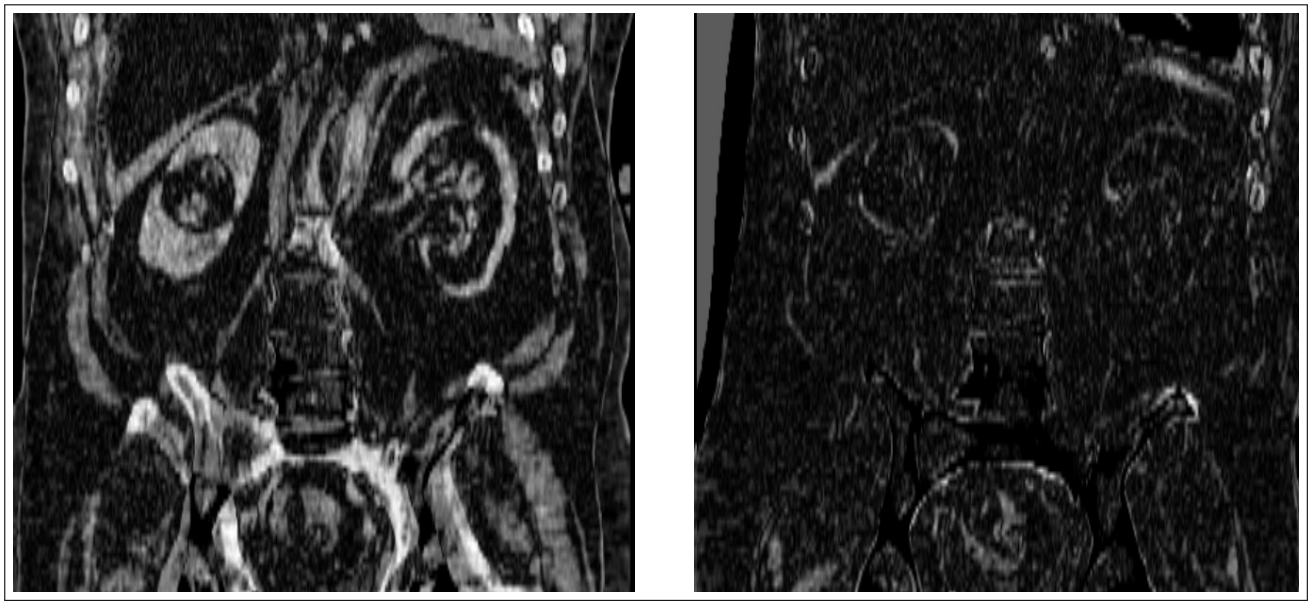
average image
volume rendering

Atlas Iterations: Step 3 — Nonrigid

3 Intensity-based image registration

Assumption: When images are correctly aligned they should look very similar.

The estimation of transformation parameters is driven by the appearance of the images, and images are registered when they appear similar.



Pixel differences before registration (left) and after (right)

3.1 Components

3.1.1 Objective Function

$$C(T) = D(I \circ T, J) \tag{3.1.1}$$

- T: Transformation
- D: Dissimilarity measure
- $I \circ T$: Moving Image
- J: Fixed Image

3.1.2 Optimisation Problem

$$\hat{T} = \arg\min_{T} C(T) \tag{3.1.2}$$

- $\arg$: return the argument (not the value)
- $\min_T$: search T with minimum cost value
- $C(T)$: cost function where $C : \mathbb{R}^d \to \mathbb{R}$ where d is the d.o.f/ parameters of transform.

3.2 Mono-modal Registration

Image intensities are related by a (simple) function. Here, a simple difference may be sufficient.

Assumption: the identity relationship between intensity distributions ("images should be ideally matched after registraition").

3.2.1 Sum of squared differences

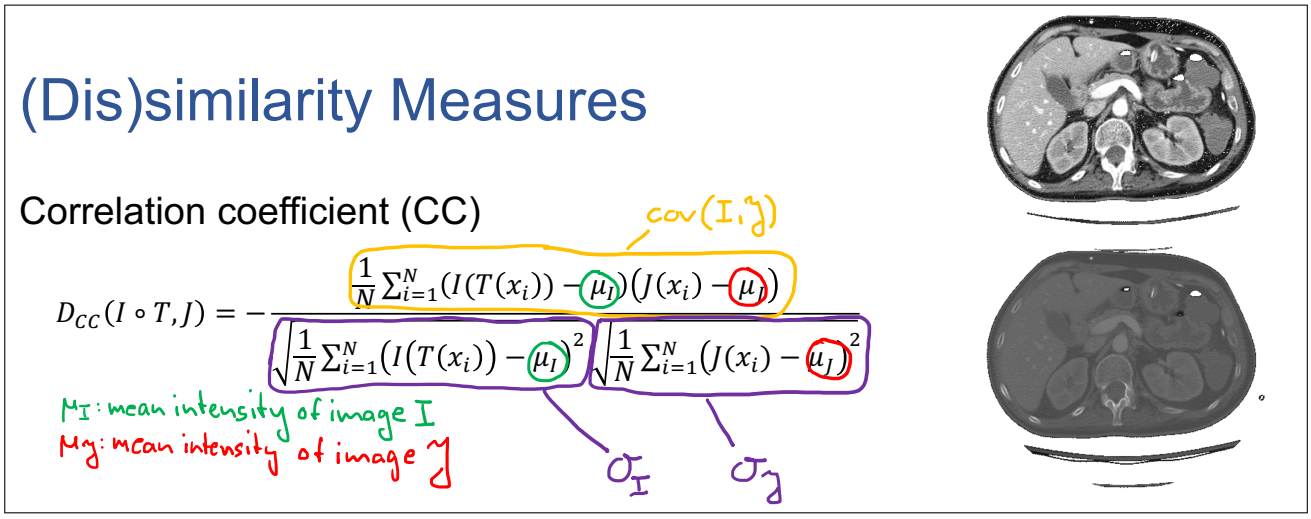
$$D_{SSD}(I \circ T, J) = \frac{1}{N}\sum_{i=1}^{N}(I(T(x_i)) - J(x_i))^2 \tag{3.2.1}$$

3.2.2 Sum of absolute differences

$$D_{SAD}(I \circ T, J) = \frac{1}{N}\sum_{i=1}^{N}|I(T(x_i)) - J(x_i)| \tag{3.2.2}$$

3.2.3 Correlation coefficient (CC)

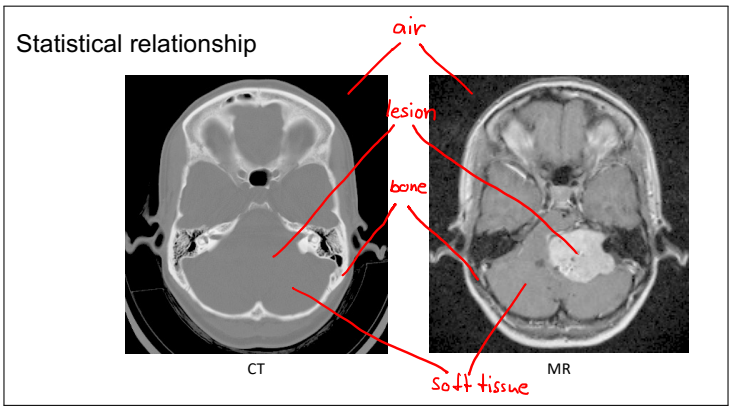
With the above dissimilarity measure, there are some scenarios where that's not quite the case. For example, if the brightness changes, then subtraction is not a good metric. We can instead compute the normalised corss correlation betwen the pixel values or the pixel intensities in one image and in the second image. We get values between 0 (no correlation) and 1 (perfect correlation).



Assumption: Linear relaionship betwene intensity distribution.

3.3 Multi-modal Registration

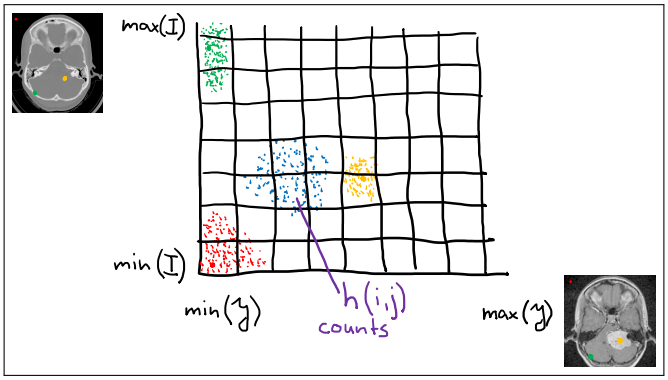
Image intensities are related by a complex funciton or statistical relationship



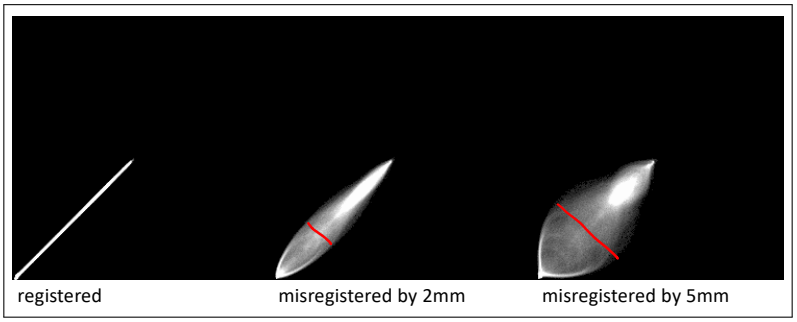
Dissimiarity in the case where images look different (posisbly different modalities) to do registration we may find correspondences (red arrows).

However, it may be that the images have very different behaviours in the two images (tumor isn't visible in one of the imaging modalities).

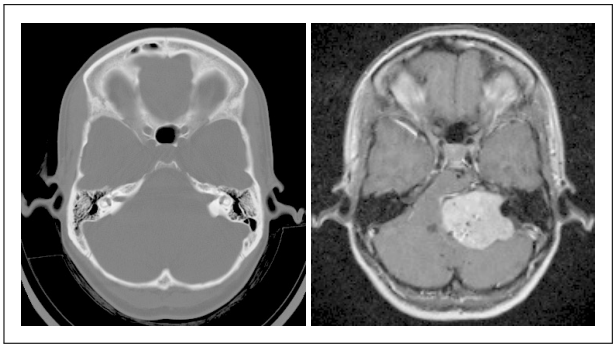
3.3.1 Statistical Relationship — Intesntiy Histograms



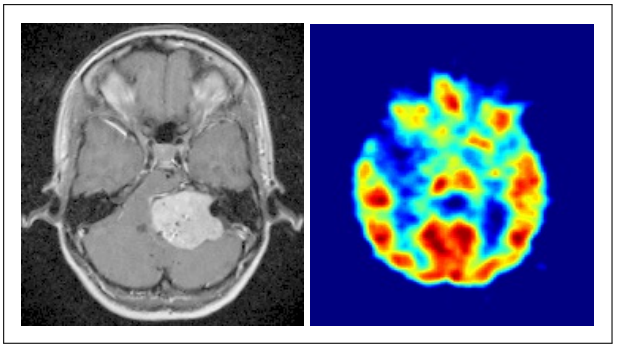
Cluster all pixels in a histograms. If two images are perfectly aligned, then this is fairly clustered, otherwise, the two images are completely misaligned.



Theoretical Example: (left) perfectly registered, the entire histogram is aligned along the diagonal (pixel values are exactly equivalent between the two), (middle) by shifting, we often have the same value but not all the time.

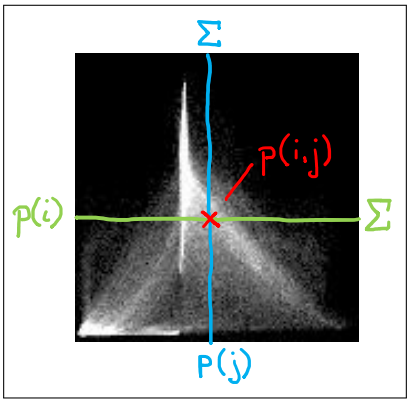


(k) Comparison between CT and MR scan



(l) Comparison between MR and PET scan

3.3.2 Intensity Distribution



$$p(i,j) = \frac{h(i,j)}{N} \tag{3.3.1}$$

joint probability of an image point having a value i in an image I and value j in image J, where $h(\cdot,\cdot)$ is the counts in histogram.

$$p(i) = \sum_j p(i,j) \tag{3.3.2}$$

marginal probability of an image point having a value i in image I.

$$p(j) = \sum_i p(i,j) \tag{3.3.3}$$

marginal probability of an image point having a value j in image J.

3.3.3 Shannon entropy

$$H(I) = -\sum_i p(i) \log p(i) \tag{3.3.4}$$

amount of arbitraty information contained in image I. This value is low if every pixel has the same pixel intensity value or high if there is a lot of infromation.

3.3.4 Joint entropy

$$H(I,J) = -\sum_i \sum_j p(i,j) \log p(i,j) \tag{3.3.5}$$

amount of information contained in the combined image I, J.

This could be used for image registration: “it measures how clustered a space is, and minimising that entropy is a good criterium is good for image registration” $D_{JE}(I \circ T, J) = H(I \circ T, J)$.

3.3.5 Mutual infromation

$$MI(I,J) = H(I) + H(J) - H(I,J) \tag{3.3.6}$$

describes how well one image can be explined by another image. This can be rewritten in terms of a marginal and join probabilities:

$$MI(I,J) = \sum_i \sum_j p(i,j) \log \frac{p(i,j)}{p(i)p(j)} \tag{3.3.7}$$

Where the dissimilaity measure is redefined as

$$D_{MI}(I \circ T, J) = -MI(I \circ T, J) \tag{3.3.8}$$

3.3.6 Normalised mutual information

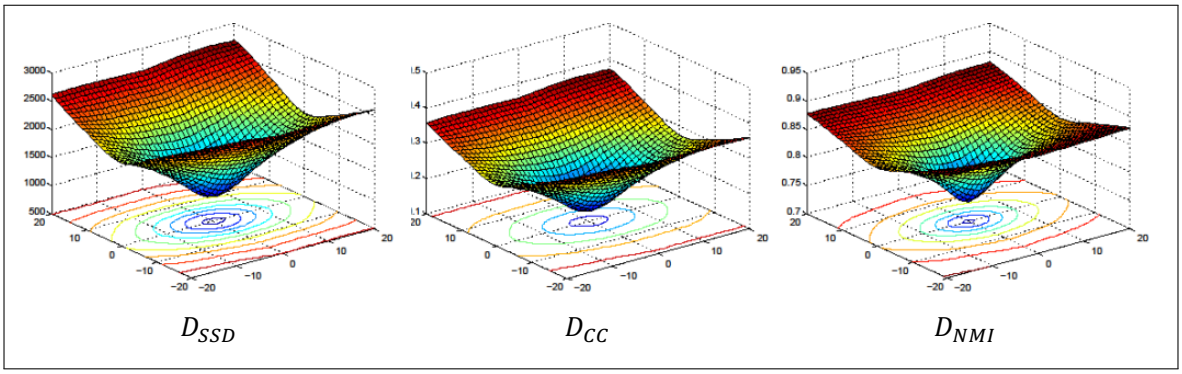
$$NMI(I,J) = \frac{H(I) + H(J)}{H(I,J)} \tag{3.3.9}$$

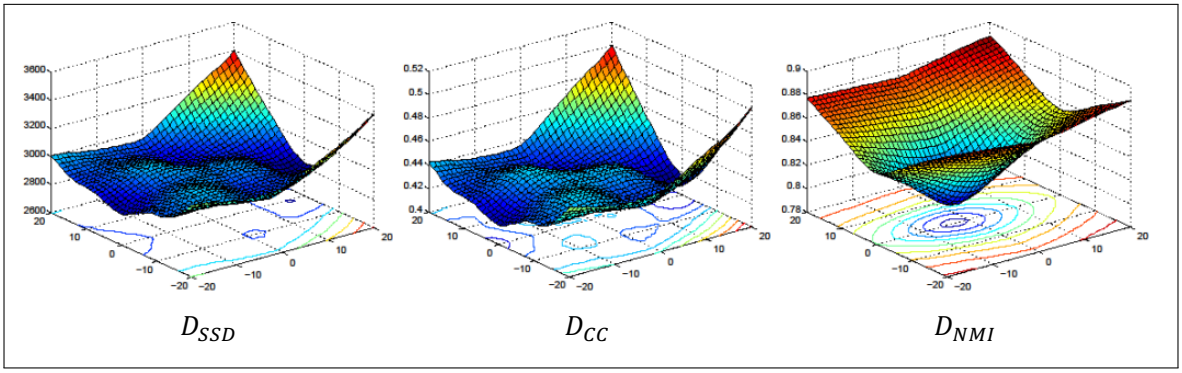
is independent of the amount of overlap bewteen images. The dissimilarity measure is redefined as

$$D_{MI}(I \circ T, J) = -NMI(I \circ T, J) \tag{3.3.10}$$

Assumption: statistical relationship between intensity distributions

3.4 Conclusion

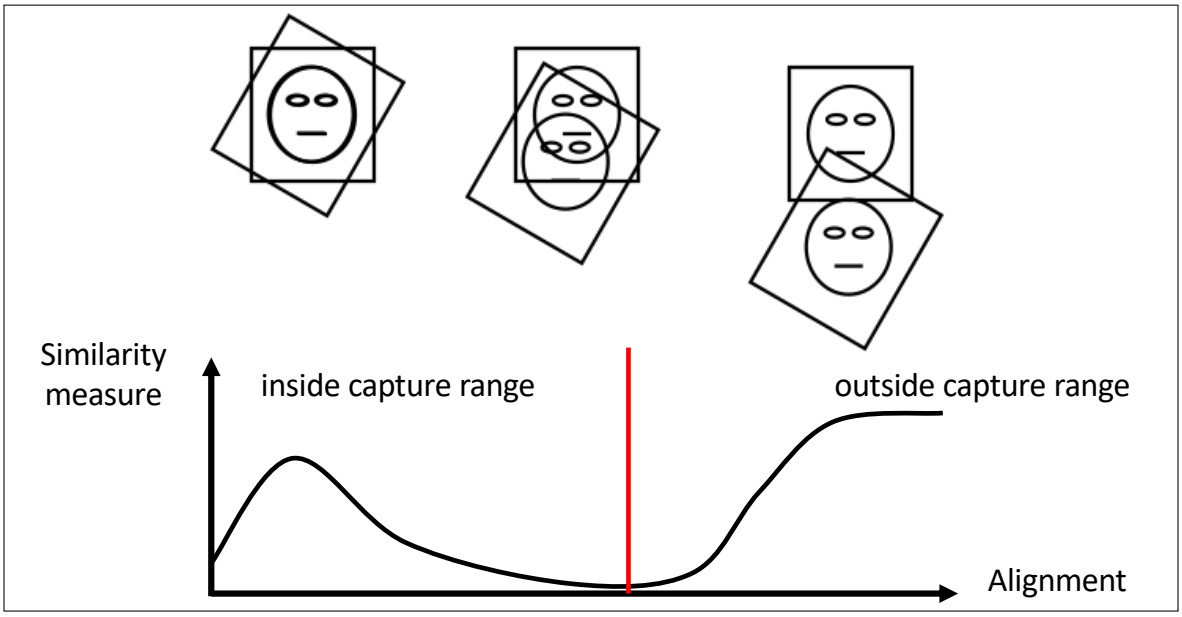




Here we see that NMI is the best for multi-modal. It doesn't assume that one image is brighter, but measures when are these two images maximally statistically related.

3.4.1 Image Overlap

(Dis)similarity mesaures are evaluated in the overalpping region of two images, which therefore requires a capture range evaluation.

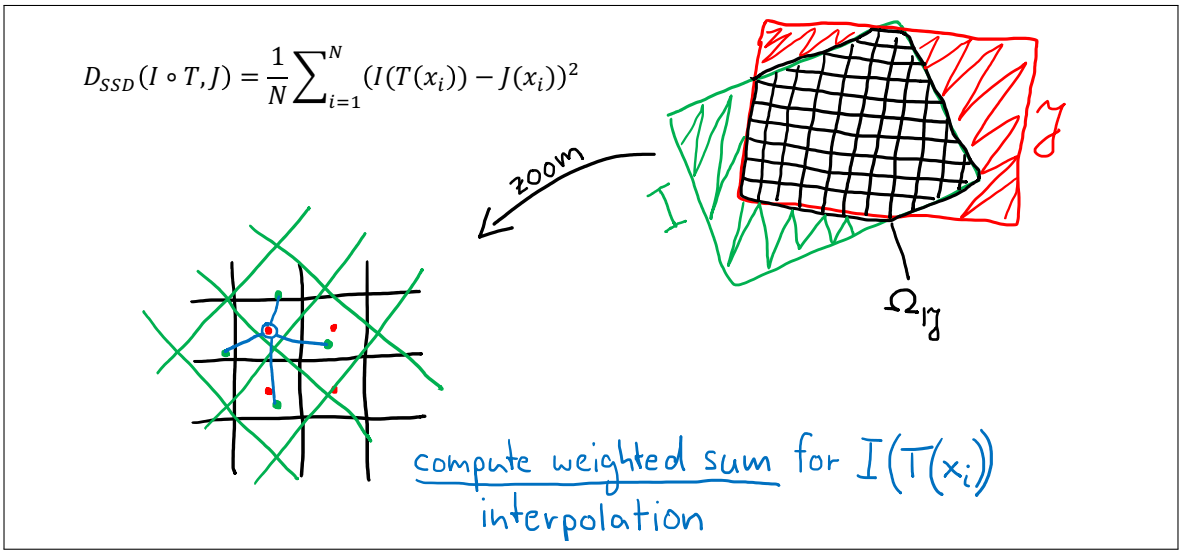


Optimisation strategy needed: need to minimise cost function, you may find local minimums, which may be a problem.

You therefore try to combat this by using:

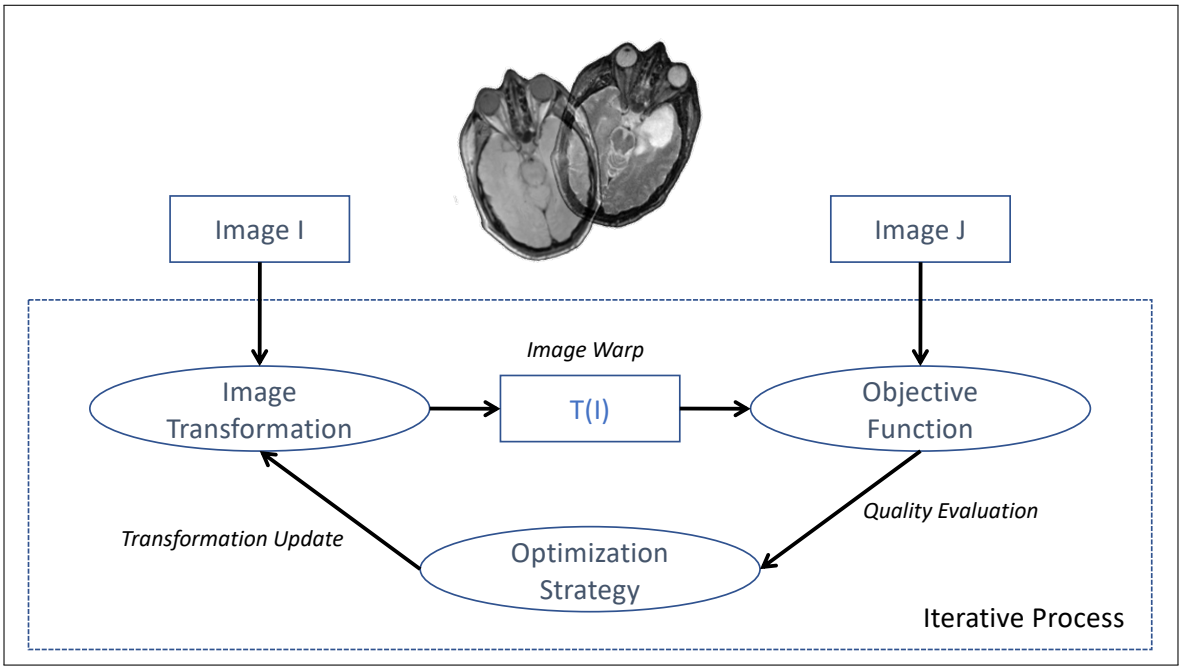
- Multi-resolution strategies — Successsively increase degrees of freedom
- smoothing — Gaussian image pyramids

3.4.2 Interpolation



Quite often, interpolation is needed, because pixel grids aren't aligned. But still, you need to compare intensities, therefore use linear interpolations for pixel comparisons.

3.4.3 Registration as an Iterative Process



Optimisation stragecy: gradient based, e.g. stochastic or complete gradient descent (without subsampling)

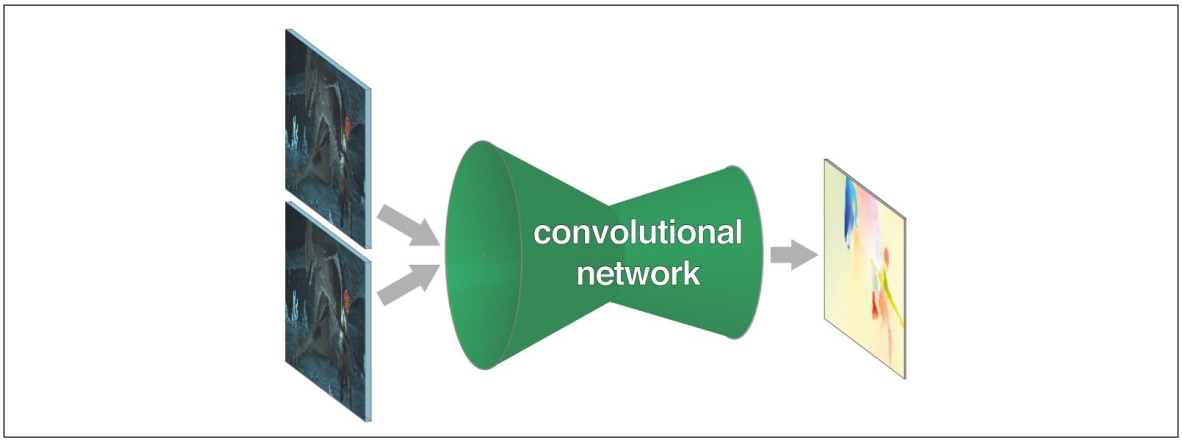
3.4.4 Optimisation strategies

- Gradient-descent
- stochastic Optimisation
- bayesian Optimisation
- discrete Optimisation
- convex Optimisation
- downhill-simplex
- ...

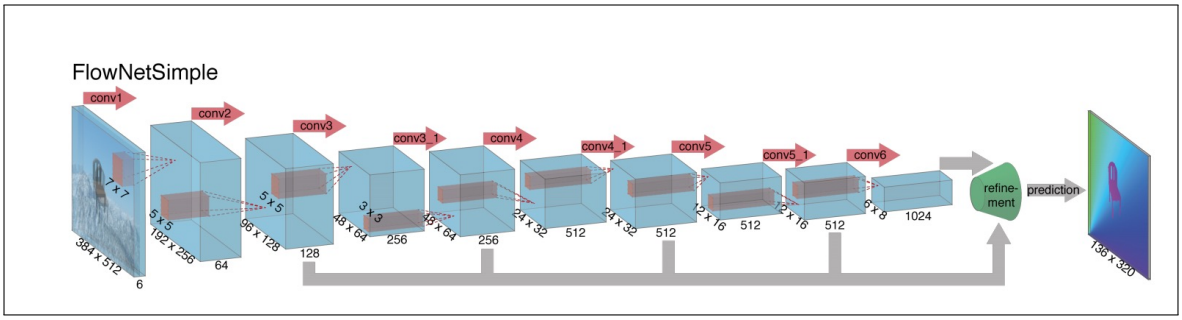
4 Neural networks for image registration — Supervised

4.1 FlowNet

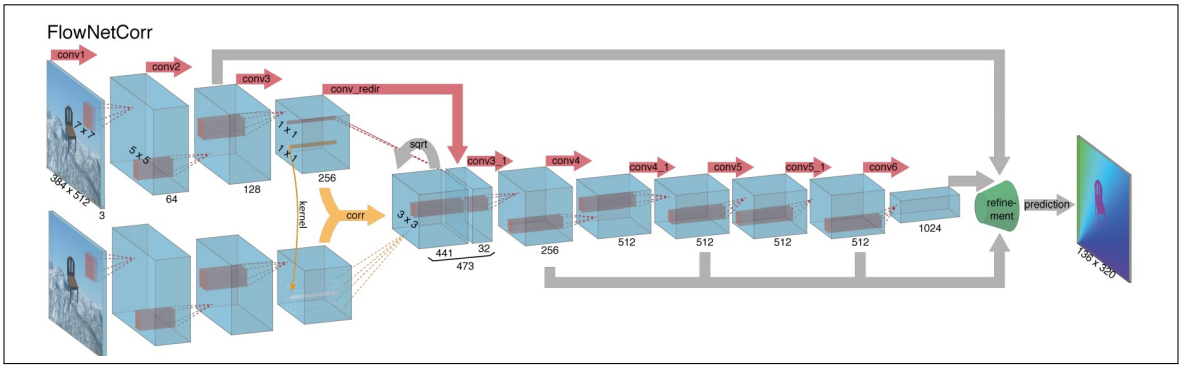
Supervised Learning approach.



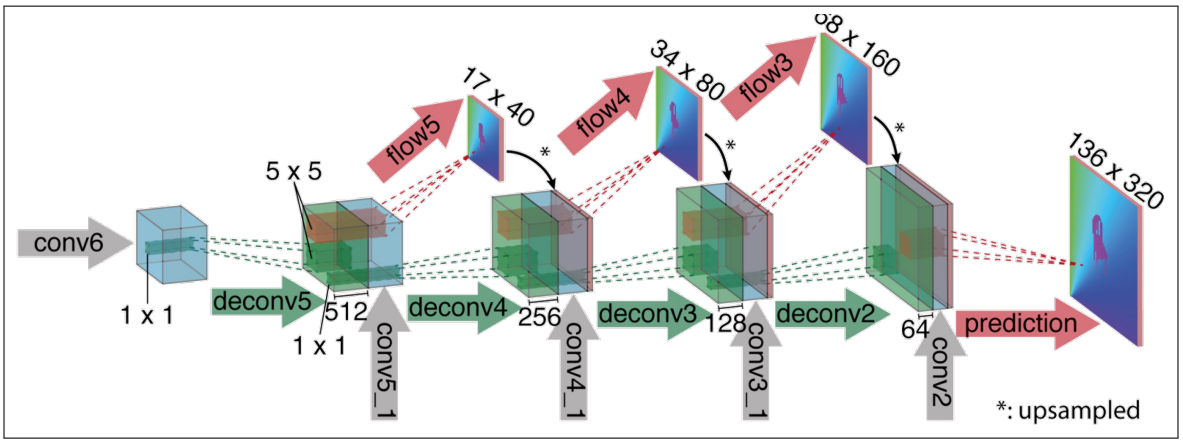
Take two frames of a video and put them into a CNN architecture. The idea is to predict the dense displacement fields between the two video frames. How do you train this?



The network proposed is the first paper to do registration in neural networks. There are 6 channels because 2×3 RGB channels, then you contract them in the network and predict the output.



Alternative architecture — network has two separate CNNs for both frames. Then do a pixel-wise correlation betwen the features then pass them through the network predicing the output. Here, we explicitly calculate a correlation between the features in both the video frames.



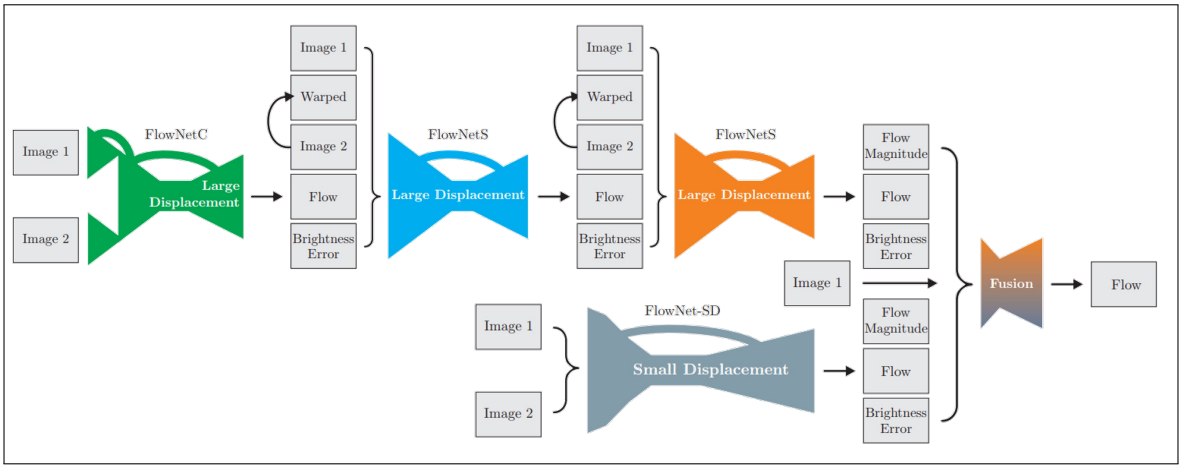
Lastly, there is an upsampling to estimate pixel-wise dense displacement.

4.1.1 How to train such a network?

Key requirement, is that you need to know the ground-truth of the optical flow. In the real-world scenario, you need to take real videos, then they renderred flying chairs in different viewpoints into these videos. Because these are rendered, you know the geometric trasnformation.

4.2 FlowNet 2.0

Evolution of Optical Flow Estimation with Deep networks



Take the flow correlation network (not good for large displacements), build a new architecture:

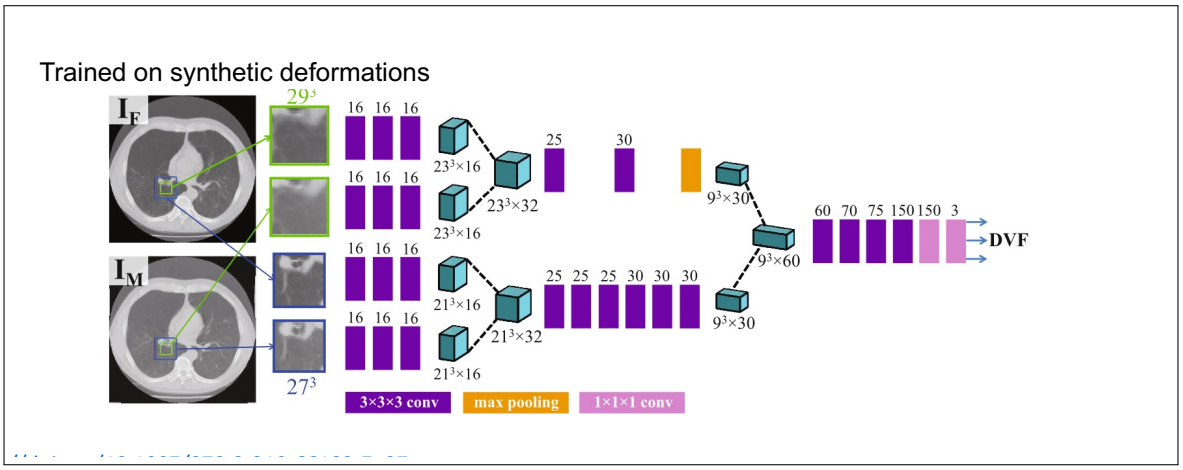
1. flow correlation network
2. take the flow as output and compute the error in terms of intensities. Take the second image and apply the warping found in the first layer to try and find the remaining residual error remaining.
3. Have a parallel branch that tries to do very detailed matching
4. A final network trying to integrate everything

4.3 Optimcal Flow with Semantic Segemntation and Localized Layers

text

Have two separate components that segment different objects. We can use information from the segmentation in the optical flow, and at the same time use this information for estimating the optical flow.

4.4 Nonrigid Image Registration Using Multi-scale 3D CNNs.

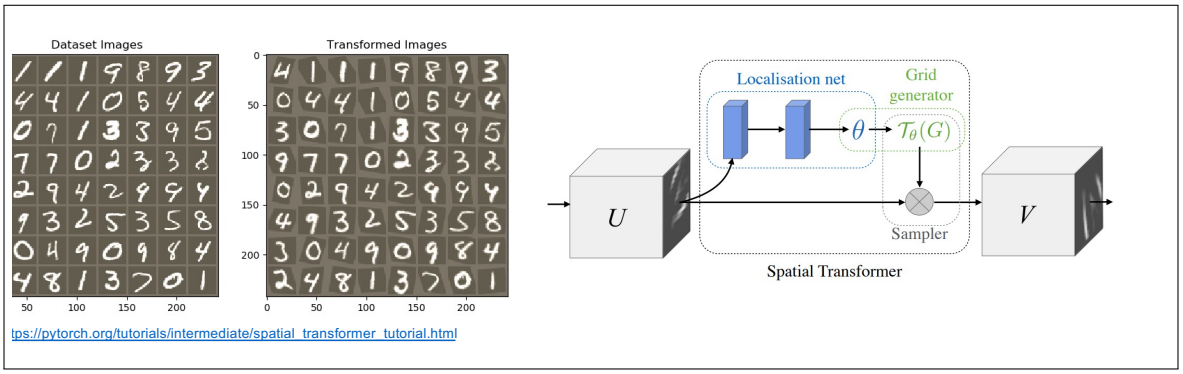


Take an image, and randomly deform it. You now know how you've deformed it so therefore, extract patches from the image (at different levels of detail) and try to predict for each patch how the center pixel moves. To use this network, you need to slide this network accross the image and generate for each pixel a displacement vector.

Here you generate the synthetic data (this may not be quite realistic)

5 Neural networks for image registration — Unsupervised

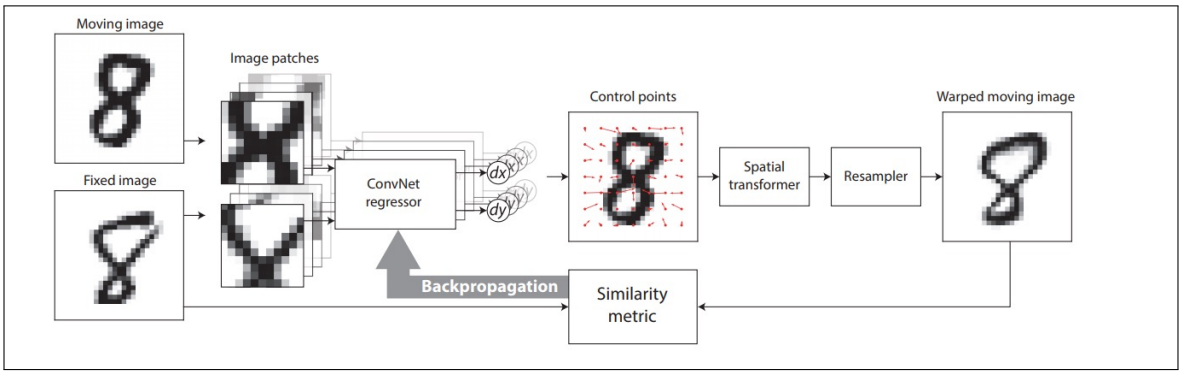
5.1 Spatial Transformer Networks



MNIST dataset — learnt to align numebrs as much as possible

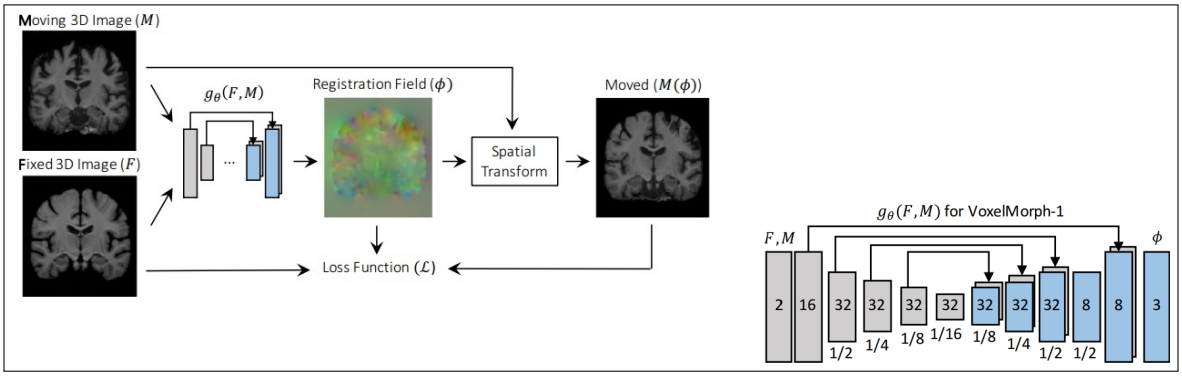
A model that allows you to take a feature map (either original or processed feature map) and predict transformation parameters from the feature map and transform the image according to this transformation map. There is a localisation net which trains θ to then deform the grid.

5.2 Unsupervised Deformable Image Registration



Take two images (you don't know the ground truth deformation field) but you have an NN which predicts the deformation. Then this is fed into a spatial transforme which transforms the input and resampler and calculate the similarity metric. It is unsupervised because you don't need to know the ground truth.

5.3 VoxelMorph



It has a u-net architecture which produces a dense displacement field. Then it uses the spacial transformer to warp the image to the fixed image then minimise the loss to the network. This only happens during training. During use, only do a forward pass.

References

[1] Max Jaderberg et al. *Spatial Transformer Networks*. 2016. arXiv: 1506.02025 [cs.CV].