

CHAPTER 2: THEORETICAL BACKGROUND

Before getting on with the proposed methods, it seems elementary to mention and explain some concepts which are to be used extensively in the coming chapters in detail. This chapter presents the building blocks which are necessary to understand the concepts presented and discussed in the forthcoming chapters. Here we elaborate the images classifications based on which they are up for different registration methodologies depending on their morphological properties. This would lead on to a discussion on geometrical transformation models inspired from physical models. A standalone description about the registration algorithms used in the methods proposed, the feature detection/description method(s) used, the database employed, and registration accuracy and similarity measures would be discussed. This chapter would help in setting just the right tone for upcoming content.

2.1 Introduction

The need for establishing this distinction between rigid and deformable images before talking about transformations in both these kinds of images is imperative. The application and use of registration algorithms to these images varies and differs both on the basis of viability (time and space complexity) and the nature of transformations which suit best for the image(s) it is being applied for. In the coming chapters, there will be detailed propositions of registration methodologies for different image sets. To keep the discussion of these algorithms relevant, it was necessary to detail the morphological properties of these varieties of images (with real-life examples). The geometric transformation models proposed in the past for deformable images are discussed; they take inspiration from physical models such as Navier-Cauchy's theories of dynamics in elastic and near-elastic

objects. It is quite interesting to see the similarities between these elasticity models and deformable images of various human body organs.

2.2 Morphological classification of Images

Image registration methods in general have been applied to a variety of kinds of images. These images have been classified into two types Rigid and Deformable images depending on their temporal behavior. These methods and their application depend on the morphological properties of the images it is being used for. For instance, an image registration technique which provides optimal results for rigid images both in terms of registration error and run time/space complexity might fail on all aspects when applied to deformable images. Similarly, an IR algorithm giving optimal results for deformable images in aforementioned terms, might be a waste of run time /space complexity & resources when applied to rigid images, as the same results could have been obtained using smaller resources had rigid image-specific IR algorithm been used.

2.2.1 Rigid Images

Rigid images are most commonly those of structures with rigid morphological properties e.g. bones, buildings, geographical structures etc. Images which do not exhibit morphological changes such as warps etc. over a period of time can be classified as rigid images. Rigid images can be modeled after real life real objects with least elasticity. To understand the behavior of rigid images, imagine an image as a collection of innumerable small miniscule points, the image will be a rigid image if there is no (ideally) relative motion between the points under deformations as can be seen in fig 2.1.

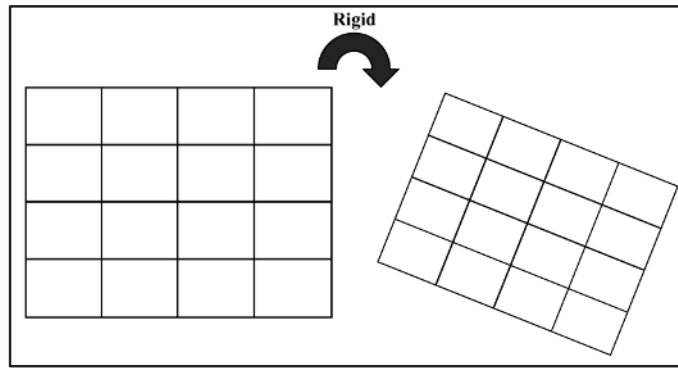


Figure 2.1: Rigid image morphological behaviour under deformation

Below in fig. 2.2 is an example from the ‘Möbius Transformations Revealed’ page illustrating the rigid motion of a Riemann sphere through different transformations.

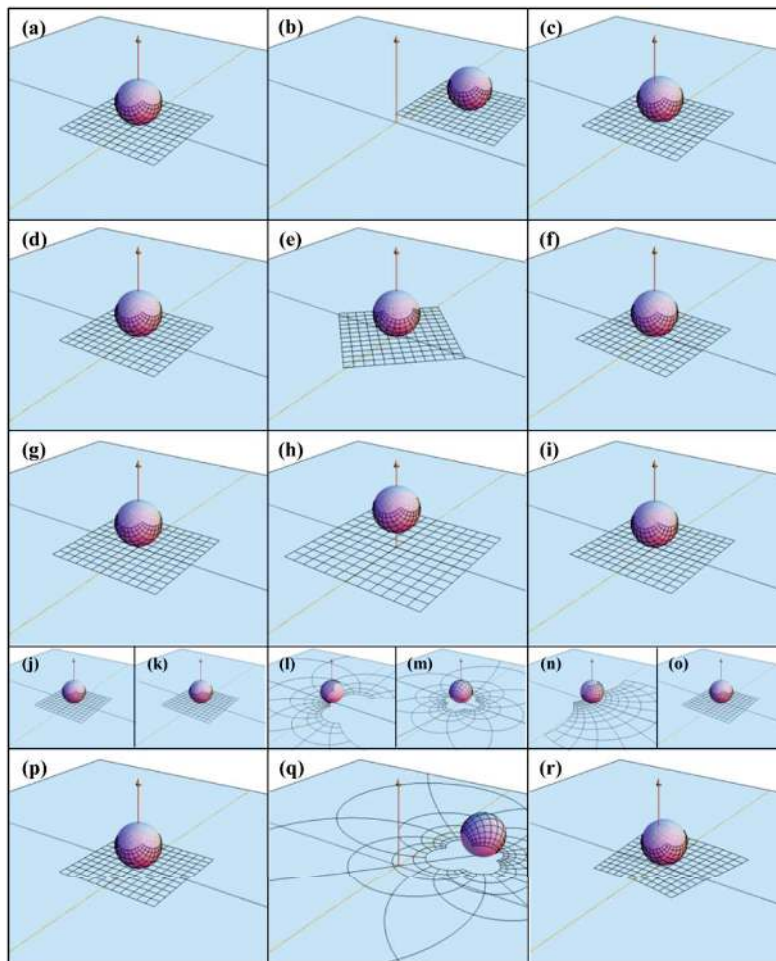


Figure 2.2: Rigid image motions in stages as shown using a Riemann Sphere; (a) to (c) is the translation motion, (d) to (f) is the rotation motion, (g) to (i) is the zoom or dilation, (j) to (o) is the inversion and (p) to (r) is a combination of them all.

The graticule of lines that we see on the plane and on the sphere form an image of a square in the complex plane. It is a projection of the sphere on the two dimensional plane. The first row depicts the translation transformation; fig. 2.2 (a) and (c) are the last initial and final stages of the transformation while (b) is an intermediary stage. These transformations can be formulated in terms of matrices, so as to make its theoretical understanding, development and applications easy. The elegance of formulating these transformations in terms of matrices is that several of them can be combined, simply by multiplying the matrices together to form a single matrix. This means that repeated re-sampling of data can be avoided when re-orienting an image.

As for translation (fig. 2.2 (a) to (c)); suppose a point x in an image is to be translated by q units, then the transformation would be simply:

$$y = x + q$$

which in matrix terms would be considered as:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \tag{2.1}$$

Looking into rotation (fig. 2.2 (d) to (f)), Consider a point at co-ordinate (x_1, x_2) on an image. A rotation of this point to new co-ordinates (y_1, y_2) , by θ radians around the origin, can be generated by the transformation:

$$y_1 = \cos(\theta)x_1 + \sin(\theta)x_2$$

$$y_2 = \cos(\theta)x_2 - \sin(\theta)x_1$$

This can be loosely related to the affine transformation in non-rigid images. For the three dimensional case, there are three orthogonal planes that an object can be rotated in. These planes of rotation are normally expressed as being around the

axes. A rotation of q_1 radians about the first (x) axis is normally called pitch, and is performed by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_1) & \sin(q_1) & 0 \\ 0 & -\sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \quad (2.2)$$

Rotations are combined by multiplying this matrix together with the other planar matrices in appropriate order. The order of the operations is important.

The transformations described so far will generate purely rigid-body mappings. Zooms/dilations (fig. 2.2 (g) to (i)) are needed to change the size of an image, or to work with images whose pixel sizes are not isotropic, or differ between images. These represent scaling along the orthogonal axes, and can be represented via:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \quad (2.3)$$

Rigid image transformation has a variety of obvious applications in registering images of solid and still structures etc., acquired from different viewing angles. It is also applicable in case of medical images of anatomical parts that do not deform significantly with over time duration. Such anatomical parts may be bones, human brain structure etc. The shape of a human brain changes very little with head movement, so rigid image transformations can be used to model different head positions of the same subject. Matching of two brain images whether it's an MR or CT image is performed by finding the rotations and translations that optimize some mutual function of the images. Rigid-image registration techniques have employed to monitor changes in the brain in

individual subjects who underwent serial MRI examinations. This approach allows disease progression and response to treatment to be monitored with great sensitivity. It fits naturally with the noninvasive nature of MRI.

2.2.2 Deformable Images

Deformable images are those of structures, shape and size of which can be modeled after tangible physically deformable models. Images that exhibit morphological changes such as warp, shape changes etc. when subjected to transformations and/or external forces over a period of time are categorized as deformable/non-rigid/elastic images. These images may or may not return to an original state with time. Deformable images can be modeled after real life objects with elastic morphological properties. To understand the behavior of deformable images, imagine an image as a collection of innumerable small miniscule points, the image will be a deformable image if there is independent relative motion of the constituent points under deformation as can be seen in fig 2.3. This essentially means that however the constituent points of a deformable image be connected/related to each other, under an external deformation, they might tend to lose that connection/association.

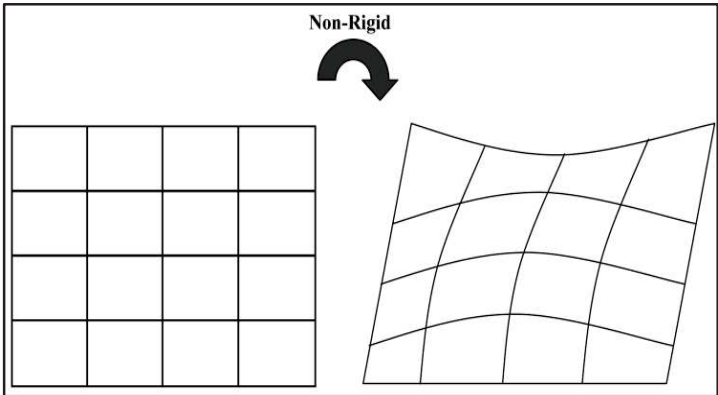


Figure 2.3: Deformable image morphological behaviour under deformation

A common example of deformable images is a temporal image sequence of an amoeba in motion. This is shown as visualization in the fig 2.4 how amoeba, an inherently shapeless microorganism moves and assimilates its food (assimilation process is not a part of image) using its pseudopods (false limbs).

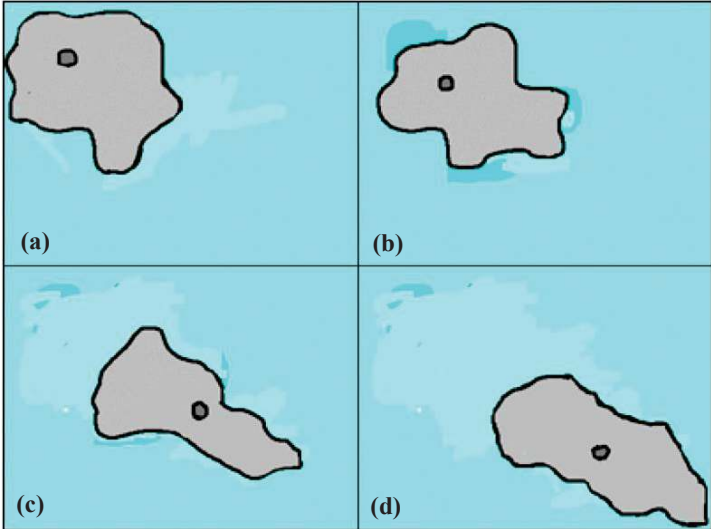


Figure 2.4: Amoeboid motion realized as a deformable image motion

There are many transformations both geometric and physical which are applicable only to deformable images and provide best emulations of real life deformations. These are radial basis functions (thin-plate or surface splines, multiquadrics etc.), physical continuum models (viscous fluids), and large deformation models (diffeomorphisms).

A quadratic transformation model is defined by second order polynomials:

$$T(x, y, z) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & \dots & a_{09} \\ a_{10} & \dots & a_{19} \\ a_{20} & \dots & a_{29} \\ 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ \dots \\ 1 \end{bmatrix} \tag{2.4}$$

Radial basis functions however like splines be it thin plate or b-splines use a linear combination of basis functions θ_i to describe the deformation field instead of using a polynomial as a linear combination of higher order terms.

$$T(x, y, z) = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & \dots & a_{0n} \\ a_{10} & \dots & a_{1n} \\ a_{20} & \dots & a_{2n} \\ 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \theta_1(x, y, z) \\ \dots \\ \theta_n(x, y, z) \\ 1 \end{bmatrix} \quad (2.5)$$

A common choice is to represent the deformation field using a set of (orthonormal) basis functions such as Fourier (trigonometric) basis functions or wavelet basis functions. In the case of trigonometric basis functions this corresponds to a spectral representation of the deformation field where each basis function describes a particular frequency of the deformation. The term spline originally referred to the use of long flexible strips of wood or metal to model the surfaces of ships and planes. These splines were bent by attaching different weights along their length. A similar concept is used to model spatial transformations. Many registration techniques using splines are based on the assumption that a set of corresponding points or landmarks can be identified in the source and target images. This is analogous to the use of point landmarks for rigid or affine registration using the Procrustes method. Thin plate splines are part of a family of splines that are based on radial basis functions. Radial basis function splines can be defined as a linear combination of n radial basis functions $\theta(s)$.

$$T(x, y, z) = a_1 + a_2x + a_3y + a_4z + \sum_{j=1}^n b_j \theta(|\varphi_j - (x, y, z)|) \quad (2.6)$$

As for the B-splines:

$$u(x, y, z) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 \theta_l(u) \theta_m(v) \theta_n(w) \varphi_{i+l, j+m, k+n} \quad (2.7)$$

Where $i = \lfloor x/\delta \rfloor - 1, j = \lfloor y/\delta \rfloor - 1, k = \lfloor z/\delta \rfloor - 1, u = x/\delta - \lfloor x/\delta \rfloor, v = y/\delta - \lfloor y/\delta \rfloor, w = z/\delta - \lfloor z/\delta \rfloor$, and θ_l represents the l -th function of the B-splines.

In case of elastic deformations (Navier's elastic PDE (partial differential equation)):

$$\mu \nabla^2 u(x, y, z) + (\lambda + \mu) \nabla(\nabla \cdot u(x, y, z)) + f(x, y, z) = 0 \quad (2.8)$$

Here u describes the displacement field, f is the external force acting on the elastic body, ∇ denotes the gradient operator, and ∇^2 denotes the Laplace operator. The parameters μ and λ are Lamé's elasticity constants which describe the behavior of the elastic body.

There are a large number of applications for deformable image registration. Since almost all anatomical parts or organs of the human body are deformable structures, they come across as a most common application for deformable images registration, for example lung motion during breathing is shown in fig. 2.5. Areas of considerable interest for deformable image registration are the applications in which the geometry during image acquisition is unknown or distorted, and include correction for scaling, gantry tilt and magnetic field inhomogeneity. Other areas of deformable image registration can be classified into either the registration of deformable structures of the same individual (intra-subject registration) or the registration across individuals (inter-subject registration). Due to the different nature of these image registration tasks, the algorithms developed to solve them have quite different characteristics.

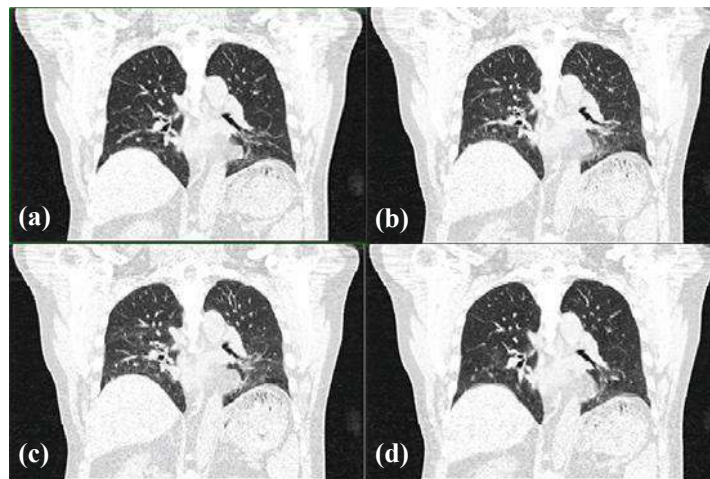


Figure 2.5: Motion of lungs during breathing

2.3 Geometric Deformation Models: A survey

An image registration algorithm can be divided into three main components, a deformation model, an objective function and an optimization method. A registration algorithm's result naturally depends on the deformation model and the objective function. The registration result's dependency on an optimization strategy follows from the fact that image registration is inherently an ill-posed problem according to Hadamard's definition of well-posed problems [Hadamard, J., 2014]. For example in a rigid setting, let us consider a scenario where two images of a disk (white background, gray foreground) are registered. Despite the fact that the number of parameters is only 6, the problem is ill-posed. The problem has no unique solution since a translation that aligns the centers of the disks followed by any rotation results in a meaningful solution [Sotiras, A. *et al.*, 2013]. However, since the subject of this work is deformable images that too body organs, in these situations in general in general, no closed-form solutions exist to estimate the registration parameters. In this setting, the search methods reach only a local minimum in the parameter space. The approach that one should take depends on the anatomical properties of the organ (for example, the heart and liver do not adhere to the same degree of deformation), the nature of observations to be registered (same modality versus multi-modal fusion), the clinical setting in which registration is to be used (e.g., offline interpretation versus computer assisted surgery). The primary interest of this work lies in deformable registration hence problems with relatively higher-degree-of-freedom setting have been discussed particularly.

The main scope of this work is focused on applications that seek to establish spatial correspondences between medical images and thus the organ state

of which the images are recorded with respect to time. The scope of this work has been extended to cover applications where the interest is to recover the apparent motion of objects between sequences of successive images (optical flow estimation) [Fleet, D., & Weiss, Y., 2006], [Baker, S. *et al.*, 2011]. Deformable registration and optical flow estimation are closely related problems. Both problems aim to establish correspondences between images. In the deformable registration case, spatial correspondences are sought, while in the optical flow case, spatial correspondences, that are associated with different time points, are looked for. Given data with a good temporal resolution, one may assume that the magnitude of the motion is limited and that image intensity is preserved in time, optical flow estimation can be regarded as a small deformation mono-modal deformable registration problem.

The parameters that registration estimates through the optimization strategy correspond to the degrees of freedom of the deformation model (these are variational approaches in general attempt to determine a function, not just a set of parameters). There is a great variation in this number, from six in the case of global rigid transformations, to millions when nonparametric dense transformations are considered. Increasing the dimensionality of the state space almost always results in enriching the descriptive power of the model. This model enrichment also brings along an increase in the model's complexity which, in turn, results in a more challenging and computationally demanding inference. Furthermore, the choice of the deformation model implies an assumption regarding the nature of the deformation to be recovered.

The geometric transformation models inspired by physical models suggested by Modersitzki in 2004 [Modersitzki, J., 2004] and currently being

employed can be separated into three/five basic categories i.e. elastic body models, viscous fluid flow models, diffusion models, (curvature registration and flows of diffeomorphisms).

Elastic Body Models can be further subdivided into Linear and Non-linear models. In the case of linear models, images under deformation are modeled as an elastic body. The Navier-Cauchy Partial Differential Equation (PDE) describes this deformation.

$$\mu \nabla^2 u + (\mu + \lambda) \nabla(\nabla \cdot u) + F = 0$$

Where, F is the force field that drives the registration based on an image matching criterion, μ refers to the rigidity that quantifies the stiffness of the material and λ is Lamé's first coefficient.

The image grid was modeled after an elastic membrane that is deformed under the influence of internal and external competing forces until a state of equilibrium is reached. The external force influences deformation in the image to achieve matching and the internal force exercises the elastic properties of the material [Broit, C., 1981].

This approach was extended in a hierarchical fashion by Bajcsy and Kovacic where the coarsest scale solution was up-sampled and was used to initialize the finer one when linear registration was used at lowest resolution [Bajcsy, R., & Kovačič, S., 1989]. Linear elastic models have also been found useful when registering brain images based on sparse correspondences. They were used for the first time by Davatzikos [Davatzikos, C., 1997] based on geometric characteristics to establish mapping between the cortical surfaces. Modeling the images as inhomogeneous elastic objects led to the estimation of a global

transformation function. Spatially-varying elasticity parameters were used to emulate the fact that certain structures tend to deform more than others.

An important drawback of image registration in general is that if deformed image is used as input to an inverse process of the previously used transformation (forward), the output obtained will not be the same as original input image for the forward transformation. The idea of parallel estimation of both forward and backward transformations, while compensating for inconsistent transformations by adding a constraint to the objective function was introduced later. Linear elasticity was used as a regularization constraint and Fourier series' were used to parameterize the transformation [Christensen, G. E., & Johnson, H. J., 2001]. A unidirectional approach was also introduced by Leow et al. that coupled the forward and backward transformations and provided an inverse consistent transformation by construction, thus diminishing the idea of a constraint addition to penalize the inconsistency error [Leow, A. *et al.*, 2005].

An important drawback of linear elastic models is their inability to cope with larger deformations. Nonlinear elastic models were proposed so as to account for large deformations. These models ensure the preservation of topology of deformable images emulating hyper-elastic materials and their properties. The use of the Finite Element method provided a solution for the nonlinear equations and local linearization [Rabbitt, R. D. *et al.*, 1995]. Two of the modeling processes for deformation were proposed, they were based on the concept of St. Venant-Kirchhoff elasticity energy [Pennec, X. *et al.*, 2005; Yanovsky, I. *et al.*, 2008].

Viscous Fluid flow models: Image under deformation is modeled as a viscous fluid; these models do not assume small deformations hence can cope with the larger ones [Christensen, G. E. *et al.*, 1996]. This transformation is

governed by the Navier-Stokes equation that is simplified by assuming a very low Reynold's number flow

$$\mu_f \nabla^2 v + (\mu_f + \lambda_f) \nabla(\nabla \cdot v) + F = 0$$

Where, v is the velocity field, while μ_f and λ_f are the viscosity co-efficients.

Christensen et al. extended their earlier work to recover transformations for brain anatomy; fluid transformation preceded by elastic registration step was used to refine the result obtained [Christensen, G. E. *et al.*, 1997]. The processes in use till then had an important drawback in the form of computational inefficiency. To circumvent this shortcoming a new fast algorithm based on a convolution filter in scale space was proposed [Bro-Nielsen, M., & Gramkow, C., 1996]. Fluid deformation models were used in an atlas-enhanced registration setting [Wang, Y., & Staib, L. H., 2000] while same models were used to tackle multi-modal registration [D'Agostino, E. *et al.*, 2003]. More recently, an inverse consistent variant of fluid registration to register Diffusion Tensor images was proposed [Chiang, M. C. *et al.*, 2008].

Diffusion models: The deformation in this case is modeled by the diffusion equation

$$\Delta u + F = 0$$

Thirion, inspired by Maxwell's Demons [Thomson, W., 1874], proposed to perform image matching as a diffusion process, his work in turn inspired most of the work done in image registration using diffusion models [Thirion, J. P., 1998]. The most suitable version for medical image analysis involved selecting all image elements as demons, calculating demon forces by considering the optical flow constraint, assuming a nonparametric deformation model that was regularized by applying a Gaussian filter after each iteration, and a tri-linear

interpolation scheme. The use of Demons, was able to provide dense correspondences but lacked sound theoretical justification [Sotiras, A. *et al.*, 2013]. However, this did not stop it from being an immediate success and soon enough a fast algorithm based on demons [Thirion, J. P., 1998] for image registration was proposed by Fischer and Modersitzki [Fischer, B., & Modersitzki, J., 2002] which provided theoretical insights into its workings. Vercauteren *et al.* [Vercauteren, T. *et al.*, 2007] adopted the alternate optimization framework that Cachier *et al.* proposed [Cachier, P. *et al.*, 2003], to relate symmetric Demons forces with the efficient second-order minimization (ESM) [Malis, E., 2004]. In this methodology, an auxiliary variable was used to separate the matching and regularization terms. ESM optimization was used to perform matching by minimizing the data term whereas regularization was achieved by Gaussian smoothing.

A variation of Thirion's Demon Algorithm was proposed by Vercauteren *et al.* endowed with the diffeomorphic property [Vercauteren, T. *et al.*, 2007]. In this approach, opposite to classical Demons approaches, an update field is estimated in all the iterations of the algorithm. A compositional rule is used between the previous estimate and the exponential map of the update field to estimate the running transformation. This exponential map is calculated by using the composition of displacement fields and the 'scaling and squaring' method [Higham, N. J., 2005; Moler, C., & Van Loan, C., 2003]. Diffeomorphism of the mapping is ensured by exponentiation of the displacement field. As an application of the model, Stefanescu *et al.* proposed a way of performing adaptive smoothing by taking into account the knowledge regarding the elasticity of tissues [Stefanescu, R. *et al.*, 2004].

The Demons algorithm has found use not only in study of scalar images but its application has been extended to multi-channel images [Peyrat, J. M. et al., 2008], diffusion tensor ones [Yeo, B. T. et al., 2009], as well as different geometries [Yeo, B. T. et al., 2010]. Peyrat et al. used multi-channel Demons to register time-series of cardiac images by enforcing trajectory constraints. Each time instance was considered as a different channel while the estimated transformation between successive channels was considered as constraint [Peyrat, J. M. et al., 2008]. Yeo & associates [Yeo, B. T. et al., 2010] derived Demons forces from the squared difference between each element of the Log-Euclidean transformed tensors while taking into account the reorientation introduced by the transformation.

Curvature Registration: These image registration methodologies don't necessarily need an extra affine linear pre-registration step, since the regularization scheme associated with it does not affect the affine linear transformations. This constraint has been used by Fischer and Modersitzki in [Fischer, B., & Modersitzki, J., 2003; 2004]. Despite several attempts to solve the underlying transformation function using the Gâteaux derivatives with Neumann boundary conditions, Henn [Henn, S., 2006] pointed out that the resulting underlying function space still penalized the affine linear displacements. Henn, further proposed including second-order terms as boundary conditions in the energy and applying a semi-implicit time discretization scheme to solve the full curvature registration problem. Beuthien and associates [Beuthien, B. *et al.*, 2010], proposed another way to solve the curvature based registration problem based on the approach presented in [Bro-Nielsen, M., & Gramkow, C., 1996] for the viscous fluid registration scenario. Instead of devising a numerical scheme to

solve the PDE that resulted from the curvature registration equilibrium equation, recursive convolutions with an appropriate Green's function were used.

Flows of Diffeomorphisms have also been one of the propositions for deformation modeling. In this case, the deformation is modeled by considering its velocity over time according to the Lagrange transport equation [Christensen, G. E. *et al.*, 1996; Dupuis, P. *et al.*, 1998; Trouvé, A., 1998]. This framework is also known as large deformation diffeomorphic metric mapping (LDDMM). It allows for the definition of a distance between images or sets of points [Joshi, S. C., & Miller, M. I., 2000; Marsland, S., & Twining, C. J., 2004]. The mathematical rigor of the LDDMM framework comes at an important cost. The fact that the velocity field has to be integrated over time results in high computational and memory demands. Moreover, the gradient descent scheme that is usually employed to solve the optimization problem of the geodesic path estimation converges slowly [Ashburner, J., & Friston, K. J., 2011]. More efficient optimization techniques for the LDDMM have been investigated in [Ashburner, J., & Friston, K. J., 2011; Marsland, S., & McLachlan, R., 2007; Cotter, C. J., & Holm, D. D., 2006]. For a tabular comparison of these methods, table A.1-A.5 from Appendix A can be referenced.

2.4 Classification of Registration Methodology Used

There is a plethora of registration algorithms being proposed by researchers and scientists being used in a multitude of applications. The classification taken up in this thesis (fig. 2.6) is the most basic classification of Image Registration algorithms roughly based on the work of Barbara Zitova and Jan Flusser [Zitova, B., & Flusser, J., 2003].

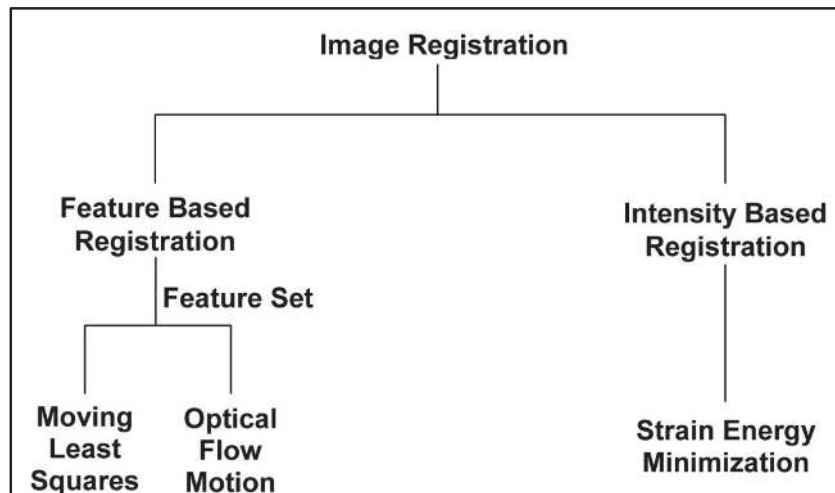


Figure 2.6: Image registration methodology classification

2.4.1 Feature Based Registration

This approach for image registration is based on detection and extraction of salient structures i.e. features in the images. Significant regions of interest in an image and lines (region boundaries, coastlines, roads, rivers) or points (region corners, line intersections, points on curves with high curvature) are understood as features ready to be used for image registration. They should be distinct, spread all over the image and efficiently detectable in both images. Each feature point has a certain set of attributes, for the features to be distinct; there should be at least one attribute of two even similar features. There should be an unbiased equally weighted uniform spread of the feature points all across the image region to be registered. These features are usually detected for being present in both source and the target images simultaneously and they are expected to be stable in time to stay at fixed positions in the target image during the whole experiment while the source image feature points may be moving. The comparability of feature sets in the source and target images is assured by the invariance and accuracy of the feature detector and by the overlap criterion. In other words, the

number of common elements of the detected sets of features should be sufficiently high, regardless of the change of image geometry, radiometric conditions, presence of additive noise, and of changes in the scanned scene. The ‘remarkableness’ of the features is implied by their definition.

Feature-based registration methods are concerned with finding the transformation that minimizes the distances between features, extracted from the pre-interventional image, or a model, and corresponding 2D features. Extraction of these geometrical features greatly reduces the amount of data, which in turn makes such registrations fast [Markelj, P. *et al.*, 2012]. The core algorithms of feature detectors in most cases follow the definitions of the ‘point’ as line intersection, centroid of closed-boundary region or local modulus maxima of the wavelet transform. Corners form specific class of features, because ‘to-be-a-corner’ property is hard to define mathematically (intuitively, corners are understood as points of high curvature on the region boundaries).

Feature-based matching methods are typically applied when the local structural information is more significant than the information carried by the image intensities. They allow registering images of completely different nature (like aerial photograph and map) and can handle complex between-image distortions. The common drawback of the feature-based methods is that the respective features might be hard to detect and/or unstable in time. The crucial point of all feature-based matching methods is to have discriminative and robust feature descriptors that are invariant to all assumed differences between the images.

2.4.1.1 Moving Least Squares

Moving least squares is a method of reconstructing continuous functions from a set of unorganized point samples via the calculation of a weighted least squares measure biased towards the region around the point at which the reconstructed value is requested. In computer graphics, the moving least squares method is useful for reconstructing a surface from a set of points. Often it is used to create a 3D surface from a point cloud through either down-sampling or up-sampling. Moving Least Squares (MLS) methods are linear systems of equations for the global least squares, and the weighted, local least squares approximation of function values from scattered data. By scattered data it should be understood as an arbitrary set of points in \mathbb{R}^d which carry scalar quantities (i.e. a scalar field in d dimensional parameter space). This scattered point cloud is the feature point set common in both target and source image pairs. The point cloud of source image is interpolated to the points in the target image using MLS, this registers surfaces in source to the target image.

The MLS method was proposed by Lancaster and Salkauskas [Lancaster, P. & Salkauskas, K., 1981] for smoothing and interpolating data. The idea was to start with a weighted least squares formulation for an arbitrary fixed point in \mathbb{R}^d and then move this point over the entire parameter domain, where a weighted least squares fit is computed and evaluated for each point individually. It can be shown that the global function $f(x)$, obtained from a set of local functions:

$$f(x) = f_x(x), \min_{f_x \in \Pi_m^d} \sum_i \theta(\|x - x_i\|) \|f_x(x_i) - f_i\|^2 \quad (2.9)$$

So instead of constructing a global approximation, it constructs and evaluates a local polynomial fit continuously over an entire domain Ω , resulting in the MLS

fit function. Moving least squares will be explained in detail with examples pertaining to this thesis in coming chapters.

2.4.1.2 Optical Flow Motion

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [Warren, D. H., & Strelow, E. R., 1985]. It is a dense field of displacement vectors which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames. Optical flow motion estimation is commonly used as a tool in motion-based segmentation and point tracking applications. Popular techniques for computing dense optical flow include methods by Horn and Schunck [Horn, B. K., & Schunck, B. G., 1981], Lucas and Kanade [Lucas, B. D., & Kanade, T., 1981], Black and Anandan [Black, M. J., & Anandan, P., 1996], and Szeliski and Coughlan [Szeliski, R., & Coughlan, J., 1997].

It is a well known registration technique which is equivalent to the equation of motion for incompressible flow in physics. The concept of optical flow was originally introduced in computer vision in order to recover the relative motion of an object and the viewer in between two successive frames of a temporal image sequence. Its fundamental assumption is that the image brightness of a particular point stays constant, i.e.

$$I(x, y, z, t) = I(x + \delta x, y + \delta y, z + \delta z, t + \delta t) \quad (2.10)$$

After a bit of mathematical interpolations it basically comes down to:

$$\Delta I + \nabla I \cdot u = 0 \quad (2.11)$$

Where ΔI the temporal difference between the images, ∇I is the spatial gradient of the image, and \mathbf{u} describes the motion between the two images. In general, additional smoothness constraints are imposed on the motion field \mathbf{u} in order to obtain a reliable estimate of the optical flow. It also helps in tracking common feature points across the sequence of images. Temporal image sequence registration and deformity estimation using optical flow motion is explained in detail in upcoming chapters.

2.4.2 Intensity Based Registration

Intensity based registration methods compare intensity patterns between images. The moving image is subjected to transformations such that the resulting transformed image exhibits minimum intensity differences with the fixed image. It has recently become the most widely used registration basis for several important applications. In this context, the term intensity is invariably used to refer to the scalar values in image pixels or voxels. The physical meaning of the pixel or voxel value depends on the modalities being registered and is very often not a direct measure of optical power (the strict definition of intensity).

Intensity-based registration involves calculating a transformation between two images using the pixel or voxel values alone. In its purest form, the registration transformation is determined by iteratively optimizing some “similarity measure” calculated from all pixel or voxel values. For deformable image registration, a major attraction of intensity-based algorithms is that the amount of preprocessing or user-interaction required is much less than for point-based methods. As a consequence, these methods are relatively easy to automate. Intensity-based registration algorithms can be used for a wide variety of applications: registering images with the same dimensionality, or different

dimensionality; both rigid transformations and registration incorporating deformation; and both inter-modality and intra-modality images.

2.4.2.1 Strain Energy Minimization

This registration technique gives best results for deformable/non-rigid/elastic images. There is a potential energy associated with an elastic system at a time. Since, the images involved in the study are of a human body organ, they can be categorized as non-rigid or deformable images and the energy principles of elastic systems are applicable to this set of images. Potential energy of an elastic two dimensional system at static equilibrium is supposed to be pure strain energy. The potential energy function although consists of tensile stress, shear modulus, shear strain and both of the Lamé's constants. This energy function is reduced to just strain energy variables and is equated to zero for static equilibrium conditions. This complete process is known as strain energy minimization. It is a transformation between the source and the target image pair. The strain energy function of the source image is minimized comparing intensity values iteratively to such a stage that no lesser intensity difference is found between the registered image and the target image. Obviously the intensity differences are never achieved to zero nor does the strain energy of the source-target image pair system ever reach zero in real life conditions.

It is fully automatic in its mode of operation and helps in faster and more accurate image registration in comparison to pure point based registration methods. This factor gives this method an upper hand when it comes to real-life medical image registration problems. The intensity based energy minimization methodology seems more practical, stable and cost efficient for deformable images in comparison to landmark based or segmentation based methodologies

for similar purposes. The method is simpler and faster than its contemporaries because the energy function is worked upon directly without solving large matrix system assemblies.

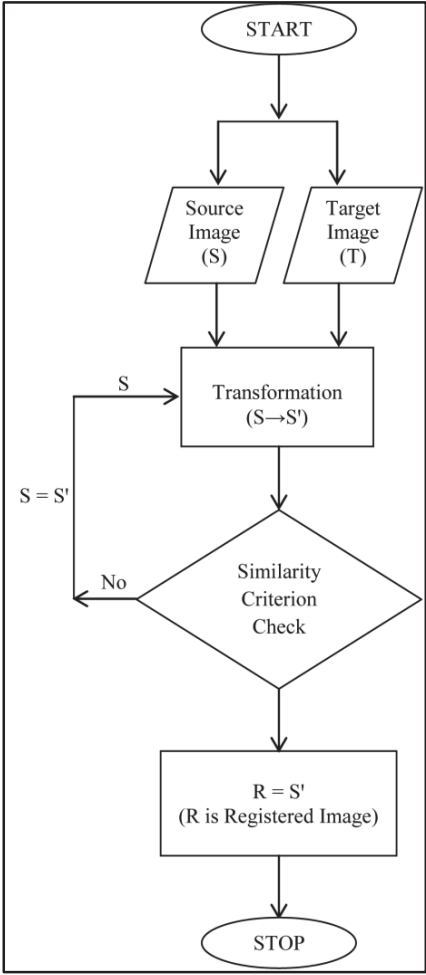


Figure 2.7: The iterative strain energy minimization process

2.5 Feature Detection/Description methods

Feature detection/description is an important precursor for implementing the feature based registration techniques for both deformable and rigid images. Feature detection is the process where we automatically examine an image to extract features that are unique to the objects in the image, in such a manner that

we are able to detect an object based on its features in different images. This detection should ideally be possible when the image shows the object with different transformations, mainly scale and rotation, or when parts of the object are occluded [Pedersen, J. T., 2011]. The feature keypoints obtained in the process have certain attributes on the basis of which their distinctiveness etc. is determined. These attributes are coordinate position $pt(x, y)$, the angle (orientation), the magnitude (response, strength), size (diameter), octave (pyramid octave in which keypoint is detected) and the object_id.

The SURF (speeded up robust features) algorithm has been explained here using details courtesy Tae-Koo Kang and associates [Kang, T. K., *et al.*, 2015]. It consists of two major parts: (1) detector and (2) descriptor. The detector uses a basic Hessian matrix approximation and an integral image, which significantly reduces computation time. The procedure it uses comprises four steps: (1) integral image; (2) Hessian matrix-based interest points; (3) scale space representation and (4) interest point localization. First, in order to speed up local feature extraction, an integral image $I_{\Sigma}(x)$ (as shown in fig. 2.8) is adapted to the SURF algorithm. The entry of an integral image $I_{\Sigma}(x)$ at a position $x = (x, y)^T$ can be represented as the summation of all the pixels in the input image I within a rectangular region generated by the origin and x as follows:

$$I_{\Sigma}(x) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (2.12)$$

Once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any upright, rectangular area. Therefore, computation is independent of the size of the rectangle. In Step 2, the Hessian matrix, $H(x, \sigma)$, is used to determine the interest points. The Hessian matrix, $H(x, \sigma)$, in x at scale σ is defined as follows:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.13)$$

Where, $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative $(\partial^2/\partial x^2)g(\sigma)$ with image I at a given point x , and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$.

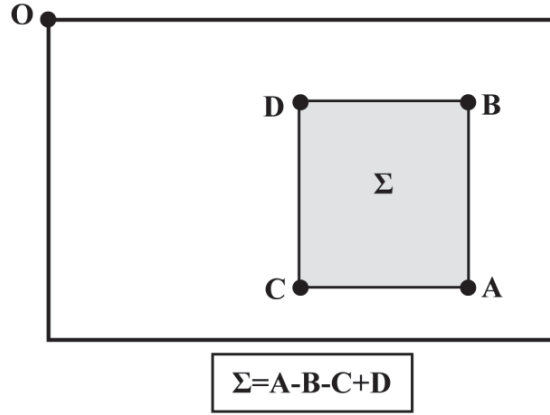


Figure 2.8: Integral image calculation

To reduce the computational cost, SURF uses the following approximation for $H(x, \sigma)$:

$$H_{app.} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.14)$$

Blob-like structures are then detected at the location where the determinant $d(H_{app.})$ is maximum using:

$$d(H_{app.}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (2.15)$$

Where the relative weight w is used to balance the expression for $d(H_{app.})$, which is needed for energy conservation between the Gaussian and the approximated Gaussian kernels. In Step 3, the scale-space representation step, Gaussian approximation filters are adapted to each level of filter size in the scale space to extract interest points from images. This scale-space representation concept has also been applied to the SIFT (Scale-invariant feature transform)

algorithm. However, the SIFT algorithm iteratively reduces the image size, whereas the SURF algorithm uses the integral images, allowing up-scaling of the filter at a constant cost. As a result, the SURF algorithm is computationally more efficient and conserves more high frequency components with no aliasing. In Step 4, the interest point localization step, interest point detection is performed using the non-maximum suppression (NMS) over three neighborhood scales ($3 \times 3 \times 3$ neighborhood pixels). The points that have the maxima of the determinant of the Hessian matrix are then regarded as the feature points by NMS.

In the descriptor, in order to assign invariability to the interest points, every interest point sought by the detector has to carry its own indicator. When deformations such as viewpoint angle changes, scale changes, increasing blur, image rotation, image blur, compression, and illumination changes occur, the interest point descriptors can be employed to look for correspondences between the original image and the transformed image. In SURF, the procedure used by the descriptor comprises of two steps: (1) orientation assignment and (2) descriptor based on the sum of Haar wavelet responses.

In Step 1, the orientation assignment step, image orientation is especially used to identify invariability of the interest point with respect to image rotation. The orientation is computed by detecting the dominant vector of the summation of the Gaussian weighted Haar wavelet responses under sliding window split circle region by $\pi/3$. Because the horizontal and vertical responses of the Haar wavelet include both the strength and directional property of interest points, image orientation efficiently represents the essentials of the image point with respect to image rotation.

In Step 2, the descriptor based on the sum of Haar wavelet response step, to discover the descriptor of interest point, the orientation selected in the orientation assignment step and the square region around the interest point are needed. Each of the square regions is split into smaller 4×4 sub-regions. For each sub-region, the horizontal Haar wavelet response d_x and the vertical Haar wavelet response d_y are computed at 5×5 regularly spaced sample points. The d_x and d_y from each sub-region are then utilized to form the 4D description vector $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$, which is called the descriptor. This method is more robust than that of SIFT.

Though there are many combined and standalone feature detector/descriptors are available in open source environment, SURF (Speeded Up Robust Feature) feature detector/descriptor has been employed for the same in this work. SURF is a unique scale- and rotation-invariant detector and descriptor, outperforming contemporary methods with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster. Focus is on scale and in-plane rotation-invariant detection and descriptions. These seem to offer a good compromise between feature complexity and robustness to commonly occurring photometric deformations in thoracic images. Skewing, anisotropic scaling, and perspective effects are assumed to be second order effects, that are covered to some degree by the overall robustness of the descriptor. For guaranteed invariance to any scale changes the input thoracic images are analyzed at different scales. The detected interest points are provided with a rotation and scale-invariant descriptor. These basic advantages that SURF provides relating to speed and relative accuracy in face of rotation, illumination changes and several other distortions led to its use for setting up a common

landmark point cloud set between the source and target image pair to assist and speed up the application of feature based image registration algorithms. Looking at the fact that these operations were being performed on CT images of real subjects (not fabricated phantoms etc.), the deformations in images were voluntary and spontaneous SURF came out to be the best choice of feature detector/descriptor for the job in terms of accuracy and speed.

2.6 Database Employed

The dataset used comprised of a total $(3 \times 10) \times 10$ i.e. 300 thoracic CT images across 10 subjects. The dataset was obtained from the publicly available database, <http://www.dir-lab.com>, with proper downloading permissions from the concerned administrator. All images were anonymized and all procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Declaration of Helsinki 1975, as revised in 2008 (5). Informed consent was obtained from all patients for being included in the study. All patients or legal representatives signed informed consent. The images lie between CT phases 0-6 i.e. end-inspiration to end-expiration in timestamp range $t_{00} \rightarrow t_{06}$. The image dimensions lie between 396×396 to 432×400 pixels. There were 6 frames from a temporal thoracic image sequence each for every Anatomical Plane (AP) i.e. Axial (supine), Coronal and Sagittal for all the 10 subjects acquired simultaneously with a gap of 0.1 second starting from time $t = 0.1$ to 0.6 seconds. The three anatomical planes are explained through graphical representations in figs. 2.9 – 2.12. All images were identified as $I_N^{AP}(x, y, t)$ where $\{N, t \in \mathbb{R}^+ \mid 1 \leq N \leq 10; 0.1 \leq t \leq 0.6\}$, (x, y) are the x & y coordinates in the Cartesian plane and AP signifies the three anatomical planes of view i.e. Axial (a), Coronal (c)

and Sagittal (s). Suppose the 3rd frame from coronal AP for subject ‘case 9’, would be notified as $I_9^c(x, y, 0.3)$. A view of the image database is shown in the tables 2.1 and 2.2 for representational purposes.

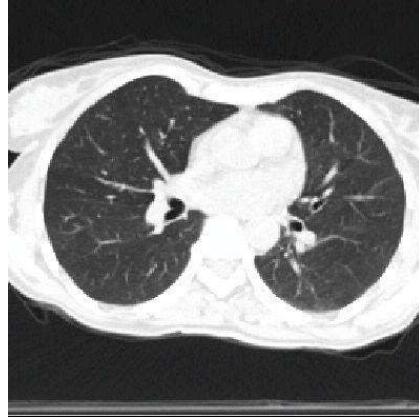


Figure 2.9: Axial anatomical position

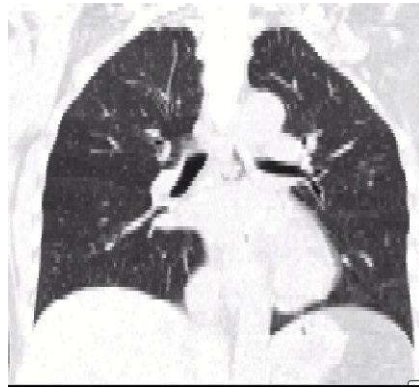


Figure 2.10: Coronal anatomical position

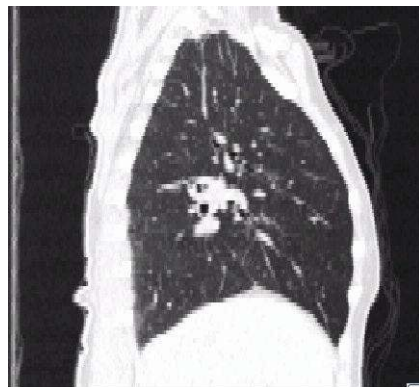


Figure 2.11: Sagittal anatomical position

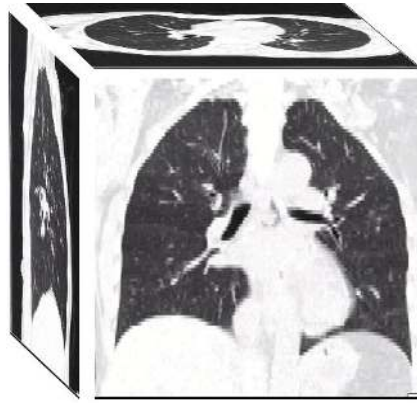
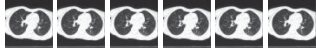

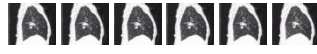
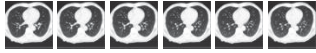

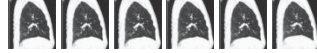
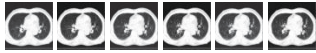
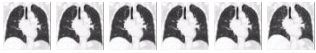

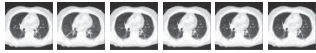


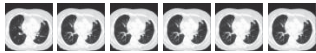
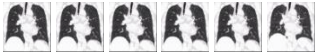
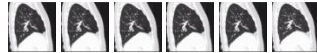


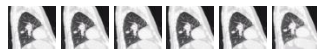


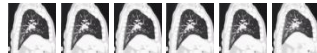


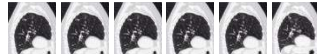
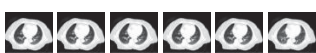
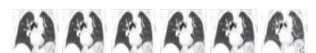
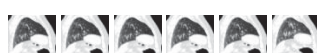


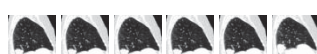


Figure 2.12: A 3-dimensional cubical representation of the APs

Table 2.1: CT images at $t=0.1$ & $t=0.6$ sec

Subjects	ANATOMICAL PLANES (T & S Images)		
	Axial	Coronal	Sagittal
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Table 2.2: Working database through all anatomical planes from t=0.1 to 0.6 sec

	Axial	Coronal	Sagittal
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

2.7 Accuracy & Similarity Measures Used

Choices of the quality of alignment as the measure of success follow directly from our definition of registration, which is the determination of a transformation that aligns points in one view of an object with corresponding points in another view of an object. Most of the work and much of the literature on the subject of registration inevitably focuses on the quest for registration methods that produce a better alignment for some combination of modalities. The success of the registration, which we are relating monotonically to the quality of the alignment, has been estimated in published work by visual inspection, by comparison with a gold standard, or by means of some self-consistency measure. Although the great

majority of studies of registration quality have been carried out for rigid-body registration algorithms, the same concepts are also applicable for non-rigid registration. The measurement of registration success will be some statistical estimate of some geometrical measure of alignment error. Many such measurements have been used to measure the quality of registration, but not all are of equal value [Silva, E. A. *et al.*, 2007]. An understanding of their meanings is crucial to understanding and evaluating claims of registration accuracy.

2.7.1 Target Registration Error

It is a common geometrical measure to assess alignment errors. It can be understood as the displacement between two corresponding points after registration, i.e., after one of the points has been subjected to the registering transformation. The word “target” in the name of this error measure is meant to suggest that the error is being measured at an anatomical position that is the target of some intervention or diagnosis. Such errors would be expected to be more meaningful than errors measured at points with no intrinsic clinical significance. Suppose p represents a point in the first image of a pair to be registered, and q a point in the second image. A registration method applied to this pair leads to a transformation T that, without loss of generality, registers the first image to the second. The difference between the two vectors representing the transformed point and the corresponding point gives the target registration error. Thus,

$$T_{RE} = T(p) - q$$

However, it's the magnitude of the target registration error that is usually reported and documented.

2.7.2 Signal to Noise Ratio (SNR)

The signal-to-noise ratio (SNR) is used in imaging as a physical measure of the sensitivity of a (digital or film) imaging system. It has been used as a metric to demonstrate enhanced similarities in a pair of images later on in comparison to the pair in its former state. The initial image pair is the source and target image pair and the later one is transformed source and target image pair. Transformed source image is the source image we get post the registration process. An increased SNR value for the post registration image pair helps in indicating a better performance by the registration algorithm.

$$SNR = 10 \cdot \log_{10} \left[\frac{\sum_0^{a-1} \sum_0^{b-1} [t(x, y)]^2}{\sum_0^{a-1} \sum_0^{b-1} [t(x, y) - s'(x, y)]^2} \right]$$

Where, $s'(x, y)$ is the transformed image post registration and $t(x, y)$ is the target image in question.

2.7.3 Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is most commonly used to measure the quality of reconstruction of a lossy compression. In terms of assessing an image registration algorithm, it has been used as an image similarity metric as well. It would compare the transformed source image $s'(x, y)$ against the target image $t(x, y)$ and get a value; this value is compared against the same for original source-target image pair. An increased value in PSNR for the transformed-target image pair would indicate a better transformation and thus a better image registration process.

$$PSNR = 10 \cdot \log_{10} \left[\frac{\max(t(x, y))^2}{\frac{1}{a \cdot b} \sum_0^{a-1} \sum_0^{b-1} [t(x, y) - s'(x, y)]^2} \right]$$

2.7.4 Structural Similarity Index (SSIM)

The structural similarity index is a method for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human visual perception. The difference with respect to other techniques mentioned previously such as SNR or PSNR is that these approaches estimate absolute errors; on the other hand, SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. SSIM has been used as an increasing factor for the transformed-target image pair in comparison to the original source-target image pair; indicating a better image transformation process and thus a better image registration process.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where, μ_x, μ_y are the mean intensities of the respective signals; σ the respective standard deviation and C_1, C_2 constants.

In practice however, it is usually required to have a single overall quality measure of the entire image. In those scenarios, a mean SSIM (MSSIM) index is used to evaluate the overall image quality. For image dimensions [a, b]:

$$MSSIM(x, y) = \frac{1}{a \cdot b} \sum_{i=0}^{a \cdot b} SSIM(x_i, y_i)$$

2.7.5 Normalized Cross Correlation (NCC)

Normalized cross correlation (NCC) has been commonly used as a metric to evaluate the degree of similarity (or dissimilarity) between two compared images. The main advantage of the normalized cross correlation over the ordinary cross correlation is that it is less sensitive to linear changes in the amplitude of illumination in the two compared images. Furthermore, the Normalized Cross Correlation is confined in the range between -1 and 1 . The setting of detection threshold value is much simpler than the cross correlation. The Normalized Cross Correlation does not have a minimal frequency domain expression. It cannot be directly computed using the more efficient FFT (Fast Fourier Transform) in the spectral domain. Its computation time increases dramatically as the window size of the template gets corpulent.

$$NCC = \frac{\sum_{i=1}^a \sum_{j=1}^b t(i, j) s'(i, j)}{\sum_{i=1}^a \sum_{j=1}^b [t(i, j)]^2}$$

Where, $t(i, j), s'(i, j)$ are the target and transformed source image frames respectively, both of dimensions $[a, b]$.