POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione

Polo regionale di Como
Laurea Specialistica in Ingegneria Informatica

Master Thesis

Implementation of divergence and curl operators embedded cost function in deformable image registration for adaptive radiotherapy

Supervisor: Marco Riboldi
Assistant Supervisor: Marta Peroni
Author: Mohamad Sadegh Riyahi Alam

Date: 20th December 2011

Table of Contents

I. Abstract

In adaptive radiotherapy optimal deformable image registration to preserve the anatomical consistency of the transformation is a key issue. The main aim of this thesis is to supply a framework in order to design and implement an image registration method which contains vector spline weighted cost function using ITK libraries. The framework is exploited to embed the similarity of the images and the anatomical consistency of the transformation in a single cost function.

For this purpose we implemented a multi-resolution image registration application containing mutual information similarity metric and a regularization term based on divergence and curl of the deformation field. This technique is based on BSpline transformations, including a coarse deformable stage to cover global dissimilarities and for a finer stage to adjust local deformations. In order to maintain the trade-off between the mutual information and regularization functions, we defined four types of weights which are assigned to div/curl parameters in both coarse and fine stages. These weights supply the users with a high flexibility in preserving the smoothness of expansion/compression irregularities of the final deformation field.
For the evaluation we profited of two datasets comprising synthetically deformed phantom images and real clinical patient images. Overall we tested up to 80 cases using design of experiments method base on observational sensitivity assessment by regulating the optimal weights of div/curl to obtain the best alignment and a smooth deformation field. For the clinical cases we also justified inverse consistency of the transformation and clinical features landmark errors.

The statistical results show that for synthetic dataset comparing to only-mutual information technique we have 4.8% improvements in term of both image similarities and smoothness of the final deformation field. We also reached to totally 16% of better recovery comparing to only-sum of squared difference similarity metric. For clinical dataset in comparison with only-MI registration, we have obtained 4% of better result considering the same metrics and 12% better persistency in evaluation of the inverse consistency.

Future work will consider the implemented framework as a mathematical platform to quantify experimentally optimal registration parameters for adaptive radiotherapy purposes.

II. Sommario

In radioterapia adattativa, l'ottenimento di una registrazione non rigida ottima che contemporaneamente preservi la consistenza anatomica della trasformazione è di importanza fondamentale. L'obiettivo fondamentale di questa tesi è la progettazione e l'implementazione di un algoritmo di registrazione non rigida di immagini la cui funzione di costo si basi su splines pesate, completamente integrato nella libreria ITK. La funzione di costo sia un termine di somiglianza tra le immagini sia la consistenza anatomica della trasformazione.

A questo scopo è stato implementato uno schema di registrazione di immagini multi-risoluzione avente la Mutual Informazione quale indice di similarità tra le immagini e un termine di regolarizzazione basato sul calcolo della divergenza e del rotore del campo di deformazione. La trasformazione è modellata con BSplines e l'algoritmo utilizza uno stage a bassa risoluzione per ricostruire le deformazioni globali, mentre quello più fine per le modificazioni più localizzate. Per mantenere il trade-off tra la Mutua Informazione e la funzione di regolarizzazioni, vengono definiti quattro tipi di pesi che sono assegnati ai parametri di divergenza e rotore sia negli stage a bassa e alta risoluzione. Questi pesi concedono all'utente finale la massima flessibilità nel conciliare la continuità e le irregolarità del tipo espansione/compressione nel campo di deformazione finale.
Per la validazione, sono stati utilizzati due diversi dataset, che comprendono rispettivamente immagini di fantocci antropomorfi deformati artificialmente e volumi di pazienti reali. In totale sono stati esaminati 80 casi, ottimizzando i pesi di divergenza e rotore sperimentalmente al fine di ottenere il miglior allineamento possibile e un campo di deformazione continuo.

Confrontando il metodo sviluppato con un classico metodo basato solo su Mutua Informazione, si nota un miglioramento del 7% in termini di somiglianza tra le immagini e una diminuzione del 15% delle irregolarità nel campo di deformazione finale. Inoltre, si nota un miglioramento complessivo del 17% rispetto ad un metodo basato solo su sum-of-squared-differences tra i voxel dell'immagine.

Sviluppi futuri verteranno nell'estenzione di questo framework a piattaforma per la definizione sperimentale dei parametri ottimi per la registrazione in applicazioni di radioterapia adattativa.

List of Figures

List of Tables

List of Graphs

Chapter 1
Introduction

1-1- Radiotherapy and Image registration

Radiation therapy is primarily used to treat cancer by locally targeting radiation to the diseased tissue. Radiation beams are produced by medical linear accelerators (Fig. 1). These devices are mounted on a gantry with a rotating couch to allow for many beam directions to be focused on the target volume [1]. It uses high-energy radiation to shrink tumors and kill cancer cells. X-rays, gamma rays, and charged particles are types of radiation used for cancer treatment. (Figure 1-1)

The way radiation therapy kills cancer cells is by damaging DNA of the tumor tissue. The radiation can either damage DNA directly or create charged particles (free radicals) within the cells that can in turn damage the DNA. Cancer cells whose DNA is damaged beyond repair stop dividing or die. When the damaged cells die, they are broken down and eliminated by the body's natural processes.
Treatment performed using radiotherapy can also hurt normal cells, which has side effects. Oncologists take potential damages to normal cells into account when they plan the treatment. The amount of radiation which does not harm the normal cells is known for each part of the body. Hence radiologists use that information to spare as much as possible those healthy structures that are more sensitive to radiation damage.
Sparing of normal tissues is accomplished in two fundamental ways: geometric avoidance of normal tissues is achieved by directing multiple beams at the target, thus delivering a high dose where the beams intersect, and a relatively lower dose outside of the intersection. Biological sparing of normal tissue is accomplished by fractionating the therapy over several weeks, i.e. irradiating daily. [1]

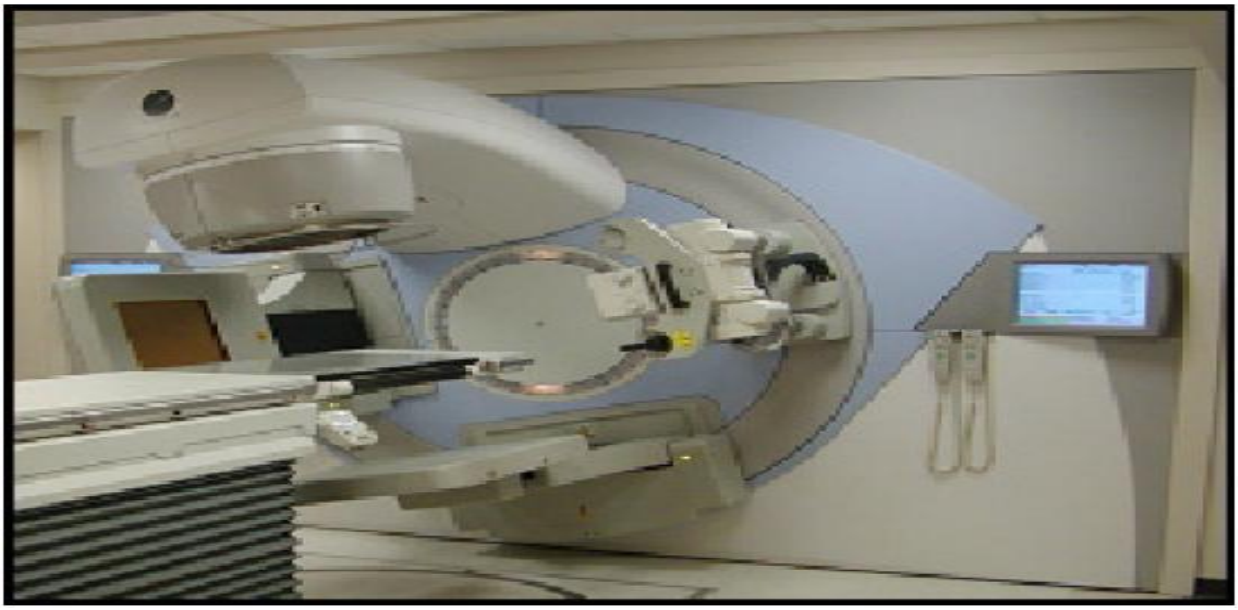


Figure 1-1- Linear accelerator with on-board kV cone-beam CT imaging unit. The device enables therapeutic irradiation and soft-tissue imaging while the patient is on the treatment unit.[1]

The amount of radiation per unit mass used in photon radiation therapy is measured in gray (Gy), and varies depending on the type and stage of cancer being treated. For curative cases, the typical dose for a solid epithelial tumor ranges from 60 to 80 Gy, while lymph nodes are treated with 20 to 40 Gy. (Figure 1-2)
Preventative doses are typically around 45 - 60 Gy in 1.8 - 2 Gy fractions (for Breast, Head, and Neck cancers). Many other factors are considered by radiation oncologists when selecting a dose, including whether the patient is receiving chemotherapy, whether radiation therapy is being administered before or after surgery, and the degree of success of surgery. Delivery parameters of a prescribed dose are determined during the treatment planning phase.

Treatment planning is generally performed on dedicated computers using specialized treatment planning software. Depending on the radiation delivery method, several angles or sources may be used to sum to the total necessary dose. The planner will try to design a plan that delivers a uniform prescription dose to the tumor and minimizes dose to surrounding healthy tissues.
The main procedure starts from prior to treatment: the patient is imaged using high resolution of Computer Tomography (CT) scan to determine the size, shape and location of the tumor. Following the CT scan, the images are digitally transferred to the treatment planning system to identify the tumor to be targeted and surrounding vital structures to be avoided and also to design the desired radiation dose to the tumor location. During treatment in-room imaging devices are used to check any modification with respect to the original CT scan employed for planning.

There are several applications of radiotherapy where image registration is applied to tackle specific issues. The concepts of adaptive radiotherapy (ART) and image-guided radiotherapy (IGRT) provide methods to monitor and adjust the treatment to accommodate changes occurring in the patient.

- Adaptive Radiotherapy (AR) is an offline approach where the anatomical and biological changes are monitored during the course of treatment by means of repeated imaging and the treatment is modified if significant changes are observed.
- Image Guided Radiotherapy is an online approach where acquired in-room images are processed on the fly to deliver the intended treatment.

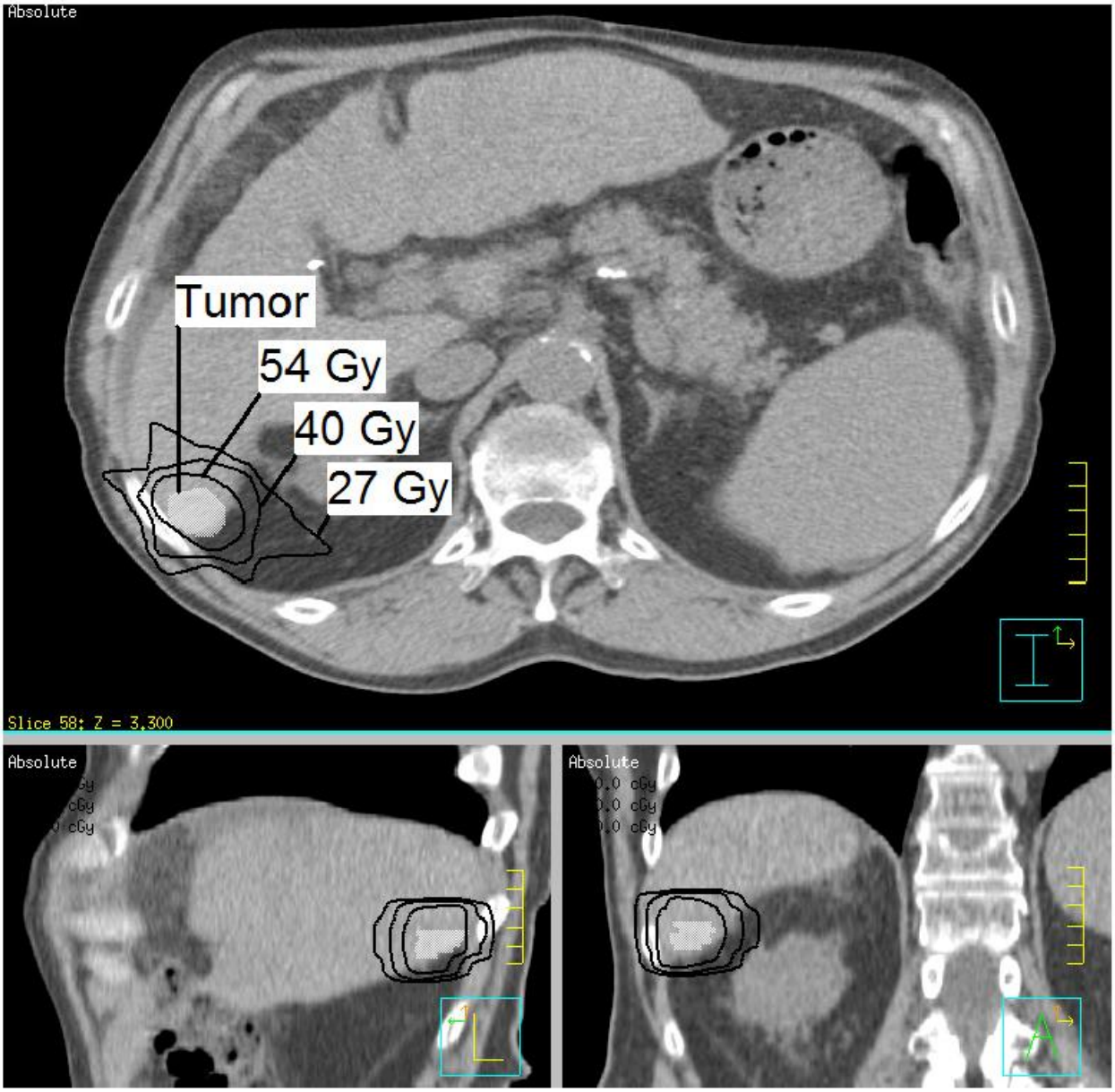


Figure 1-2- Example of a radiation treatment plan of a patient with liver cancer treated in 6 fractions with 36 Gy. Displayed are orthogonal cuts through the 3D CT, and target contours, and iso-dose lines. [1]

1-1-1- Problems and motivations

There are several serious issues that need to be considered during the radiotherapy treatment in order to optimize the correct delivery of the radiation dose.

In practice, patient set-up errors, relative changes of patient anatomy due to breathing and organ movements occur during the radiotherapy treatment. Also, since the patient model is updated sporadically, the treatment plan may not be optimal and it is difficult to update the patient model since it typically requires extra CT scan. Such errors lead radiotherapy to the unnecessary irradiation of normal healthy tissues.

The following factors are considered to be the major difficulties which influence the correct radiation delivery.

1- Movement of the body

- Intra-Fractional geometrical motion occurs at different anatomical sites due to breathing, motion or bladder filling on a short time scale (during each treatment fraction). Such motion may cause a geometrical miss of the target volume, thus leading to overdosage of the healthy tissues.
- Inter-fractional (day-to-day) geometric change occurs over the weeks of therapy, due to digestive processes, change of breathing patterns, differences in patient setup. These changes are taken into account by population-based “uncertainty” margins around the target area, which may be excessive or conservative and are applied to the structures identified before the therapy begins.

2- Deformation of the organ

Occasionally tumor volumes have shrinkage in response to treatment. Tumor shape deformation is due to filling state change of neighboring organs or relative position changes between tumor and normal organs, such as treatment response of growth or shrinkage of the tumor or nearby organs (e.g., the parotids in head and neck treatment). This could be a problem during the radiotherapy treatment. Image registration can be used to model the movement of the tumor and track it down.

In order to tackle the abovementioned problems, Deformable Image Registration (DIR) can be used in order to solve and compensate motion and deformation during the radiotherapy treatment.

Deformable registration is a fundamental image tool that is widely used for the analysis of medical images. Its use in the field of radiation therapy, either Image-Guided Radiotherapy or Adaptive Radiotherapy is relatively recent and in constant progression.

Image registration is the process of defining the transformation between two images so that the coordinates in one image correspond to those in the other. When the mapping includes deformation, this is referred to as deformable image registration. (Figure 1-3)
The major use of image registration is for estimating patient set up and anatomical changes in radiotherapy. While the patient set up estimation is accomplished by rigid image registration, patient anatomical changes require a deformable image registration method.

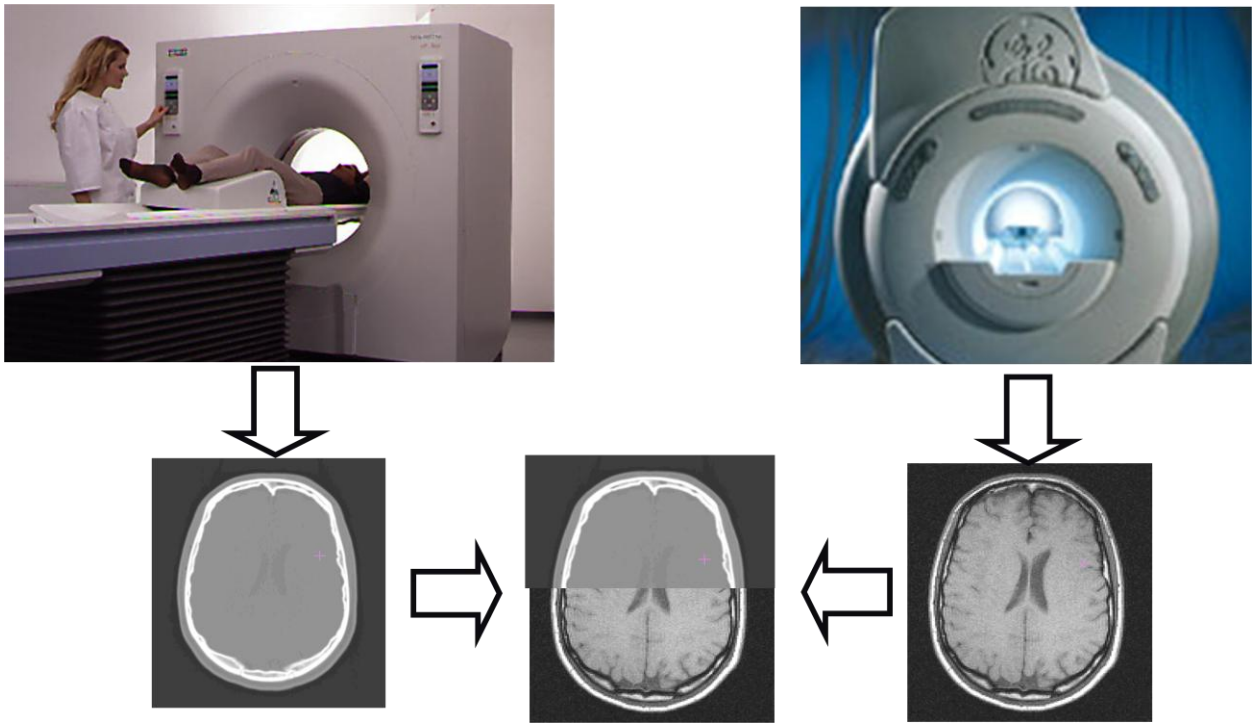


Figure 1-3

Figure 1-30. An example of image registration [2]

The application of deformable registration in radiotherapy is not only restricted to movement of the body and deformation of the organs (inter/intra fraction). Deformable registration has been vastly used in radiotherapy and it is considered as a critical operation during treatment planning, delivery and the patient follow up. It turns out to be useful in performing contour propagation and dose distribution adaptation as well as for tumor tracking and for patient follow-up procedures after the therapy. The major motivations for registering medical images in radiotherapy are described in the next part of this chapter in detail.

Registration of two images of the same part of the body is essential for many applications where the correspondence between the two images conveys the desired information. These two images can be produced by different modalities, for example CT and MRI, can be taken from the same patient with the same instrument at different times, or can belong to two different subjects. This perception leads to the definition of intra-patient and inter-patient registration.

- Intra-Patient Registration: this is the strategy for registering two images which belong to one patient either in the same or different modality.

- Inter-Patient Registration: this strategy is used to help transferring contours from one patient to another in order to study variability of anatomical structure across different patients, or to use anatomical atlases to contour incoming new patients.

The goal of deformable image registration in radiotherapy is to resolve differences in geometry while maintaining modality-specific differences in information content by means of estimating the spatial relationship between the volume elements (i.e., the image voxels) of corresponding structures across image data sets.

The main aim of this thesis is to supply a framework in order to design and implement an image registration method which contains vector spline weighted cost function using ITK libraries. The framework is exploited to empirically obtain the optimal registration parameters for parametric registration applied for radiotherapy application purposes. The optimization process will consider both the similarity of the images and the anatomical consistency of the transformation, as detailed in the following paragraphs.

1-1-2- Outlines

In order to better understand our method we have listed the robustness of this thesis as follows:

1- In the literature it has been shown that for registering 3D/3D multimodality images with deformation, mutual information (MI) as the measure of similarity and BSpline transformation function work reasonably. Also the vector splines of the divergence and curl of the deformation field in order to penalize the misregistration have proven good results. However there are very few designs and implementations that have been done in this regard. We have successfully developed this application of image registration using ITK VC++ framework. The application is portable and is designed to be multi-platform.

2- We have specially concentrated on the part of regularization to form the final deformation field. Consequently we have defined four types of weights to be assigned to the divergence and curl during the registration. Two weights for the Bspline coarse grid stage and two weights for the fine grid stage.

3- In comparison to the cost function implemented by ITK which contains only-MI, our cost function penalizes irregularities in the final deformation field. The Normalized Mutual Information (NMI) between the fixed and final warped image is improved by approximately 7%.

4- Although the application has been implemented for academic research and optimization is needed in terms of design and development, it can be used as a plug-in in other imaging analysis applications.

5- In terms of computational time, by using 256x256x61 resolution 3D meta-images as fixed and moving, in 3GB RAM and 1.3 GHZ intel-CPU with 256 MB dedicated memory RAM, it takes almost 2 hour and half on average, where most of the time is spent on calculation of the deformation field at each iteration of the optimizer to calculate the regularization function, disregard it from the similarity between the images. This computation time may seem quite big, but who have worked with the registration

programs using MATLAB knows it takes days to perform a trivial cost function contains two or three terms. Therefore the computation time is also improved very dramatically.
This thesis is organized as follows:
In this chapter we introduced basic concepts regarding the radiotherapy applications and the use of deformable image registration in radiotherapy. Also the main components of the image registration have been described along with the current state of the art regarding the divergence and curl operators. Finally an introduction is presented regarding the registration performance evaluation methods.

In chapter 2, our method and implementation techniques are described. The definition of the cost function contains MI and div-curl, problems and solutions have been described. The methodologies for the implementation of our cost function are described in detail as well. Also the methods of evaluation of the final warped image resulting from the deformable registration are presented.

In chapter 3, all the results are shown with tables and graphs obtained through the evaluation method described in chapter 2. We have tested two types of database of images. The synthetically deformed CT images of a head and neck phantom and also the exhale/inhale lung CT images of real-patient cases. Also in this chapter we have obtained three types of optimal registration parameters which are optimal BSpline grid spacing, subsampling resolution and number of iterations of the optimizer. These registration parameters are obtained empirically using our cost function.

In the final chapter we discuss about the results we have obtained in our synthetic cases and the real patient cases. We have also compared our results with other methods such as MI and Sum of Squared Differences cost function. Eventually summary, conclusion and future work make this thesis over.

Since the aforementioned applications in radiotherapy are considered as the major motivation to perform deformable image registration, hence in the next part each of the application is described separately in detail

1-1-3- Aligning to a common system

Directly combine the images from multiple studies, to enhance specific features of different modalities. The integration of the useful data obtained from multiple imaging modalities for radiotherapy planning is achieved by image registration software.

In this kind of study Functional Images (PET, SPECT,…) which provide physiological information, biological activities as well as metabolism are combined with Anatomical Images (X-ray, ultrasound, CT or MRI), which provide anatomy structure of the body. Fusing both capabilities gives us a better treatment planning dataset in a very high resolution output.
PET scans are increasingly utilized along with CT scans, the combination ("co-registration") giving both anatomic and metabolic information (i.e., what the structure is, and what it is doing biochemically). Because PET imaging is most useful in combination with anatomical imaging, such as CT, modern PET scanners are now available with

integrated high-end multi-detector-row CT scanners. So the areas of abnormality on the PET imaging can be more perfectly correlated with anatomy on the CT images. This is very useful in showing detailed views of moving organs or structures with higher anatomical variation, which is more common outside the brain.
Another example is represented by MRI/CT fusion: the hard bone features of a CT imaging study can be combined with the soft tissue features of an MRI study by adding the bone extracted from the CT to the MR dataset.

1-1-4- Perform contour propagation

To perform contour propagation across multiple imaging studies, which consists in warping the contours drawn by the physician as a function of the deformation field computed by image registration. This strategy allows reliable treatment planning and verification, because it accounts for physiological and radiation-induced movements and shape/size modification in the tumor area. Radiotherapy contours drawn on a reference phase are then propagated to the target phase. Additional non-rigid registration further reduces the propagation error and negates errors caused by small observer variations.

As an example Ming Chao et al [2] developed an effective technique to automatically propagate contours from planning CT to cone beam CT (CBCT) to facilitate CBCT-guided prostate adaptive radiation therapy. It was found that the approach was able to reliably warp the constructed narrow band with accuracy better than 1.3 mm.
Likewise it has been cited that an effective technique to automatically propagate contours from planning CT to cone beam CT (CBCT) to facilitate CBCT-guided prostate adaptive radiotherapy is being probed [2].

1-1-5- Dose distribution propagation

To propagate the dose distribution computed in one imaging study to another one in order to verify/accumulate the dose delivered to the target and organ at risk. This allows physicians to compute the cumulative dose delivered to a patient even when deformation occurs. This factor is crucial, as the dose delivered to normal critical structures is the main factor that regulates side effects in a radiotherapy treatment. It is therefore very important to track the dose delivered to each organ very carefully, and to include such information in the patient chart for future evaluation of his/her clinical conditions.

1-1-6- Follow-up treatment

Registering the baseline patients' images and the follow up images after the therapy to see the consequence of the treatment and to assess the need to retreat.
Likewise, non-physiological reasons can be taken into account to motivate the employment of image registration. Due to the irradiation, a part of the body could be swollen or scarred, and registration could help to keep track and compensate these effects.

Consequently from the described concepts it can be perceived that image registration is useful in the stages of treatment planning, delivery, and adaptation.

1-2- Image registration algorithms components

Image registration is the process of defining a mapping between two images so that the coordinates in one image (fixed) correspond to those in the other one (moving). One image is considered as the reference (or fixed) image and the other one is the deformable (moving) image. We denote I the reference image and J the deformable image. The output is a transformation φ which relates the content of the first image to the content of the second image. Let $X = (X1, X2, X3)$ (for a 3D image) be the coordinates of a point in I, $\varphi(X) = X'$ (X prime) is the corresponding point in J: I(X) = J(φ(X)).(Figure 1-4)

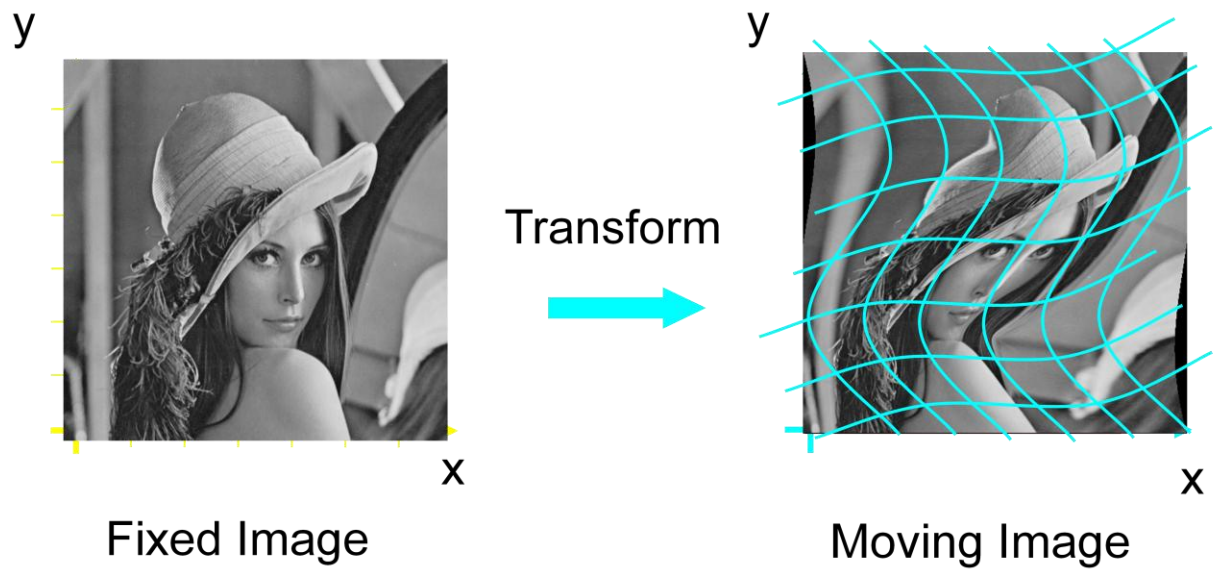


Figure 1-40. Deformable transformation [3]

The transformation φ can be defined as a "deformation field", i.e. as a vector field where the displacement of each voxel is characterized as a vector (magnitude and direction) in 3 dimensions. Each vector in the deformation field represents the distance between X and φ(X).

$$P_{out} = P_{in} - D(P_{out}; P_{in})$$

P_{out} and P_{in} each represent a random point in the fixed and deformed or warped image and $D(P_{out}; P_{in})$ denotes the distance between the latter two points.

Generally in term of transformation function, there are two types of image registration methods: linear vs. non-linear. In the linear method, the transformation function maps line onto the line or points but in the non-linear or deformation method, the mapping of two images contains deformation and the transformation maps lines onto curves. There are two widely used linear registration methods: Rigid and Affine registration which now they are part of most treatment planning systems. Both methods apply the linear transformation function on the moving image, but they have different matrix of transformation to map the moving image onto the fixed image. The details are discussed in part 1-2-1.

The final output of the registration procedure which is the product of aligning two images is called “warped image”. In this thesis we have concentrated on the deformable registration between fixed and target images.

In deformable registration, we fit the transformation that optimizes a given similarity metric. Hence, for applying deformable registration mainly three components are needed: a transformation function, a similarity measure and an optimization strategy. However when we have a undesired discontinuities in the final warped image, we can define another component (regularization function) to penalize the irregularities and misalignments in the registration procedure.
The output of the deformable registration is a deformation field of vectors (vector field) which as shown in the above latter equation, indicates the displacement of one voxel in fixed image comparing to the corresponding voxel in deformed or warped image.

Let’s simulate one iteration of the deformable registration procedure. The idea relies on passing the physical or pixel-wise coordinate parameters (x,y,z) belonging to fixed image to each term of the cost function which is composed of similarity metric function and a regularization function(s). The cost function searches the best alignment between two images depending on type of the similarity metric. For example if we have chosen the intensity based Sum of Squared Difference (SSD) between two images, it will utilize each coordinate parameters of the fixed image and find the corresponding point on the moving image and compute the squared difference between those points. The role of the optimizer is to minimize this difference.
The regularization function computes the displacement (deformation field) between two images at each iteration to compensate the irregularities and misalignment occurred. When the final value of the cost function is calculated, it is passed to the optimizer.
Then based on the “stopping criteria” defined for the optimizer, it evaluates the value received at that specific iteration to whether continue the optimization procedure or it should be stopped by that point. This optimization iteration is done until the optimizer meets the stopping criteria and the distance between two points in term of intensity and the physical coordinate has been set to minimum. In this point the finally obtained transformation parameters are considered as the most optimum and the best for aligning two images.
Eventually using the obtained optimized transformation parameters, we obtain the warped image. This stage is done through the resampling of the moving image with respect to the optimal transformation parameters. (Figure 1-5)

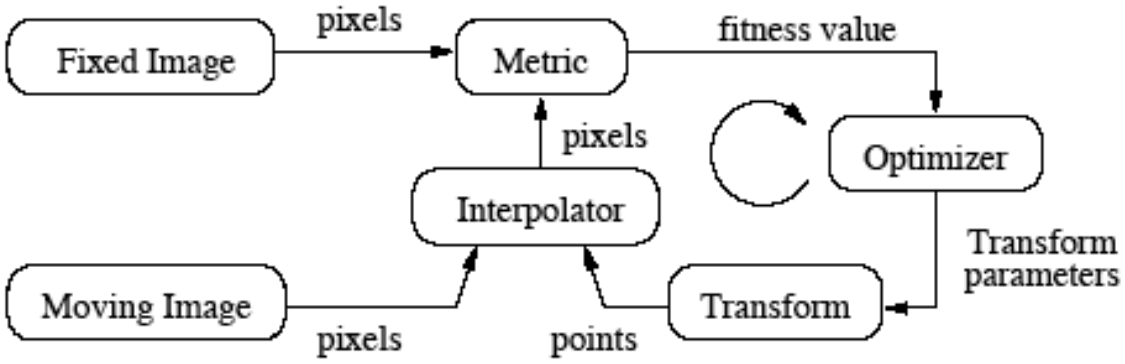


Figure 1-5. The whole procedure of the image registration [3]

Conventionally the only item which is normally included into the cost function is the type of similarity measure that is optimized using a proper optimization algorithm to reach the best alignment. Lately several novel cost functions have been proposed where, in addition to similarity measure, different components are included as regularization functions. These components are mostly assumed as a proper factor to be constrained in order to achieve optimal registration results. By using this approach several techniques have been proposed to obtain optimum values for the registration parameters.

The major aim of registration is to find a deformation field that spatially relates two images, such that the deformed "moving" image $I_m(X + U(X))$ matches the "fixed" image I_f at every position.
The deformation field is represented as $\varphi(x) = (x_1 + u_1(x), x_2 + u_2(x), x_3 + u_3(x))$ where $u_n(x)$ is the displacement vector of each voxel in each x, y and z direction. Ideally we want to determine a displacement field u such that $I_m(X + U(X)) = I_f(X)$.
The ideal approach depends on the type of similarity metric one uses. For instance one can easily minimize the distance of $\sum(I_m(X + U(X)) - I_f(X))^2$ the so-called Sum of Squared Differences (SSD), or one can utilize the probabilistic density estimation function to maximize the joint entropy and consequently the Mutual Information (MI) between two images.

The following demonstration shows the definition of the registration as an optimization problem.

$$\hat{\mu} = argmin\ \mathrm{C}(\mu; I_f; I_m)\ with\ subject\ to\ \mu$$

Where the C (cost function) equals the similarity metric as any type, and μ represents the transformation parameter vectors that parameterize the deformation field. For instance if one uses cubic B-Spline non-linear polynomial function to model the deformation field then μ represents the transformation parameters that containing the B-spline coefficients which is described in the next part (1-2-1). Depending on the type of

optimizer used, the cost function may have multiple local minima. Those local minimum is selected as the solution depending on the optimization algorithm and on the initial alignment of the images. A regularization term can be added to the cost function, to penalize undesirable deformations, and consequently, to reduce the number of local minima. Hence the whole cost function can be revised as follows: [4]

$$C(\mu; I_f; I_m) = -SM(\mu; I_f; I_m) + \omega\mathcal{R}(\mu)$$

In this equation, ω serves as a weighting factor for the regularization term. $SM(\mu; I_f; I_m)$ denotes the similarity measure between $I_f; I_m$ and $\mathcal{R}(\mu)$ exposes the regularization term. Well-known examples for $\mathcal{R}(\mu)$ are the curvature term, the elastic energy, and the volume preserving penalty term.

In the following next parts each component of the deformable image registration is described in detail.

1-2-1- Transformation function

The goal of a registration procedure is to apply a coordinate transform to the moving image, so that it is best aligned with the fixed image. Thus the transformation function is applied to the moving image.

As mentioned in the previous part, image registration is categorized into linear and non-linear in terms of transformation function. For the linear functions there are several matrices to warp the fixed image such as rigid, affine, similitude, mesh based [5] although rigid and affine linear registration are the most widespread. In rigid registration a 6 degree of freedom transformation matrix describes rotation and translation parameters to be applied to the moving image. This means that φ (the transformation function) is parameterized with six numbers: three translation and three rotation parameters. A linear transformation is such that φ can be written as a 3 × 3 matrix A (9 parameters).

$$\varphi(x) = Ax$$

In the affine registration, the transformation matrix has 12 degree of freedom to describe shearing and scaling of the moving image. In this case phi is calculated as

$$\varphi(x) = Ax + b$$

(b belongs R^3) [6].

Depending on the mechanism or method used to model the deformation, the deformable model can usually be categorized into parametric and non-parametric (variational) method. However in some other categorization non-rigid registration are categorized as global modeling (polynomial, harmonic), semi-local (piecewise polynomial with various splines) and local (regularized dense vector fields) modeling (table 1). [6]

The aim of non-parametric model is to model the registration as a deformation process of certain material driven by external forces. For this technique the registration is based

on the regularized minimization of a distance measure. A regularizing term is used to circumvent ill-posedness and to privilege more likely solutions.
For parametric techniques the registration is based on the regularized minimization of a distance measure. In the parametric deformable registration, deformation is characterized by some control points such as coefficients of the radial basis functions.
They are based on parametric radial basis function model such as the Thin-plate splines (TPS) or second order Laplacian splines.
Models based on N order polynomial BSplines are commonly used because of the capability of n times differentiability, along with the computational advantages in the representation rigid translation. The basis functions of the BSpline model are used to represent the local deformation on the image. (Figure 1-6)

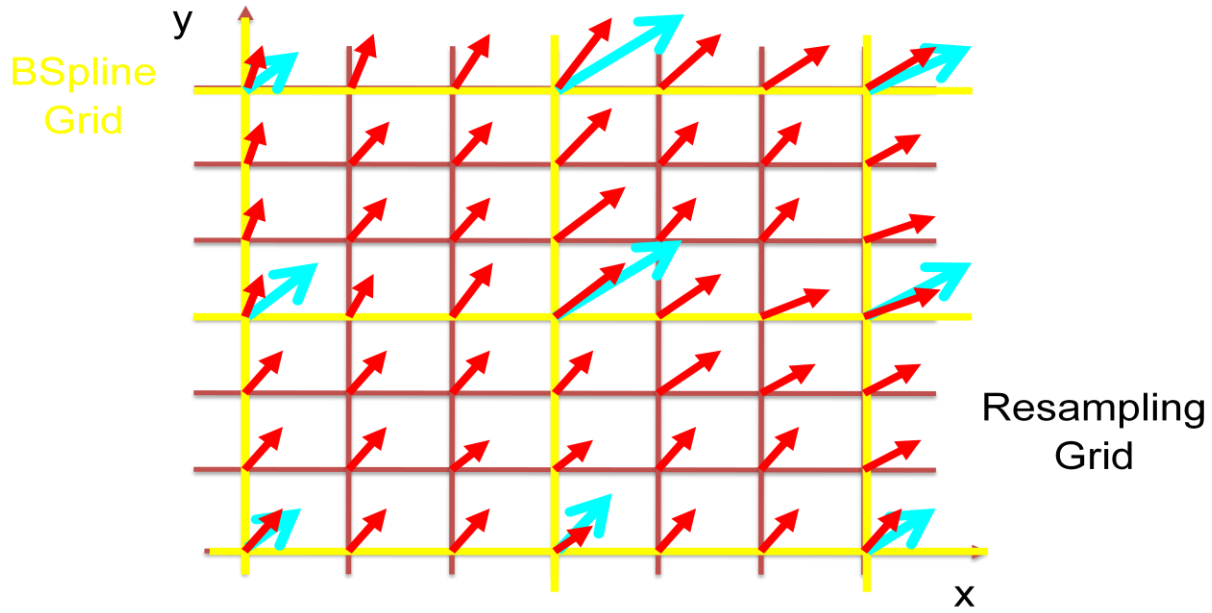


Figure 1-6. BSpline grid nodes. Red arrows ling on the red grid represent the resampling grid, which is the displacement of each vector of the image. The yellow grid with the light blue arrows represent the BSpline grid points.[3]

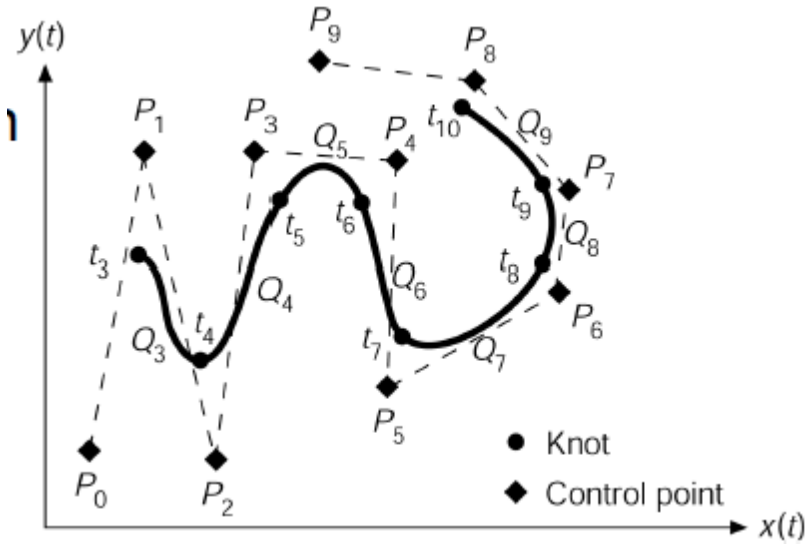


Figure1-7. BSpline control points and knot points

As can be seen in Figure 1-7 Bspline functions can parameterize the transformation relying on control points or knot points. The BSpline function consists of multiplication of the smooth n order blending function and the control points (coefficients), mean each control point is associated with a basis function $B_{i,d}(t)$ as follows:

$$P(t) = \sum_{i=0}^{n} B_{i,d}(t) P_i$$

Here P_i are the control points computed with basis functions (Basis-splines). $B_{i,d}(t)$ B-spline basis functions are blending functions. Each point on the curve is defined by the blending of the control points (B_i is the i-th B-spline blending function and d is the degree of the BSpline).

$$B_{k,0}(t) = \begin{cases} 1, & \text{if } t_k \le t < t_{k+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{k,d}(t) = \frac{t - t_k}{t_{k+d} - t_k} B_{k,d-1}(t) + \frac{t_{k+d+1} - t}{t_{k+d+1} - t_{k+1}} B_{k+1,d-1}(t)$$

As can be seen from the above equation, the BSpline function bigger than degree 1 is defined as a recursive function.

1-2-2- Similarity metric

In term of similarity measure generally there are two approaches for image registration: Feature-based method and Intensity-based method. Briefly in the feature-based there are many control points resulting from the manual selection by the experts over the image and those points are used to establish transformation technique derived by the mathematical method. Each of the selected features is compared with potential corresponding features on the other image. Some pairs of matched features are

selected as the control points. Finally using local transformation models such as piecewise polynomial functions (splines), we interpolate the control points to finally form the deformation field and to calculate the voxel displacement. Using this method obviously human intervention is inevitable. In this kind of approach the optimizer minimizes the distance between features, points or the surfaces of corresponding anatomical structures thus requiring prior feature extraction.

In the intensity-based method, we find a single transformation to be applied on the whole image, and usually it uses voxel gray level or color directly to measure the similarity between the images. The main difference between the two approaches is that in Intensity based technique the registration transformation is determined by optimizing the similarity cost function directly from the voxel values of the images, instead of points or surfaces derived from the images. The global intensity based registration is referred to as automatic registration method and it usually needs more computation time than the feature based registration approach. This kind of approach is not dependant on any feature extraction thus any starting point or initialization. (Table 1-1)

Features space and similarity measures	
Features-based	• *Types : points, lines, surface ...* • *Selection : manual or automatic* • *Pairing : manual or automatic*
Intensity-based	• *Local : SAD, SSD* • *Neighbourhood : CC, MI, CR ...*
Hybrid	• *Mix IB and FB methods*

Transformation models	
Global	• *Affine (almost rigid)* • *Polynomial* • *Harmonic*
Semi-global (RBF, piecewise polynomial)	• *Non-compact support : TPS, Fourier series, wavelets, elastic, div-curl* • *Compact support : Wendland, B-spline*
Local (regularized dense vector field)	• *Membrane model (Laplacian)* • *Bi-harmonic model (2nd order Laplacian, TPS)* • *Linear elastic, fluid* • *Gaussian* • *Jacobian-based ...*
Biomechanical model	• *FEM methods with triangular (surface) or tetrahedral (volumes) meshes* • *Individual organ properties (Young's modulus and Poisson's ratio)* • *Contact-impact analysis*

Table1-1. Short classification of some feature-spaces and transformation models used in deformable registration.[6]

Depending on the type of metric we choose to measure the similarity between the images, some functions are maximized by the optimizer and some other functions are minimized. For example as described in the following part, for measuring the Root Mean Square mismatch between the voxel points in images we must look for minimization of that metric gradually using the iteration done by the optimizer, but for measuring the mutual information between the images we choose the type of optimizer to maximize it. However this matter encounters a problem when we define additional term such as regularization functions. Since usually regularization terms are defined to compensate

the discontinuity of the final transformation, thus the algorithm of the registration expects the optimizer to minimize that kind of penalty function. Therefore for example inside the cost function which represents both mutual information as the similarity metric and the 2D bending energy of a thin plate of metal, we will have to consider that the mutual information must be maximized but the regularization term should be minimized.
D. Rueckert et al [9] theoretically resolved this problem simply by demonstrating a minus sign and negate the similarity metric from the regularities we have defined. A simple example has been described in the following [9]. However when we come to the implementation this matter becomes a complex problem.

$$C(\theta, \varphi) = -C_{similarity}\left(I(t_0), T\left(I(t)\right)\right) + \lambda C_{smooth}(T)$$

In the above equation, $I(t_0)$ is the fixed image and $I(t)$ is the deformed image. T represents the transformation function and $C_{similarity}$ represents the similarity applied between two images. C_{smooth} indicates the regularization function and λ here denotes the weight to regulate the scale between two aforementioned terms. Inverse trends between the two terms have been demonstrated by a negative sign assigned to $C_{similarity}$.
Below some vastly used similarity metrics are briefly described.

Sum of Squared Differences
It simply computes the squared differences between corresponding voxel points on two images in terms of gray values.

$$SSD = \sum_{i \in M} (I_1(i) - I_2(i))^2$$

Where M is the region of overlap between $I_1$and I_2. Since this metric directly depends on the intensity values of the voxels, it is very sensitive and susceptible to noise. Hence using it for registering multimodality images is not suggested and it is widely used for pre-enhanced mono modality images. On the other hand the least square form of SSD (Residual Sum Squared-RSS) makes the measure computationally attractive since fast optimization schemes such as Gauss-Newton or Levenberg-Marquardt can be applied. [1]

Cross Correlation (CC)
This metric could be proper if the gray values of the images have a strong correlation between each other.

$$CC = \frac{\sum(I_1(i) - \bar{I}_1)(I_2(i) - \bar{I}_2)}{\sqrt{\sum(I_1(i) - \bar{I}_1)^2}\sqrt{(I_2(i) - \bar{I}_2)^2}}$$

The global CC algorithm selects the whole fixed image as a convolution mask and moves over the moving image to find the best match in term of desired gray intensity

values. In the local CC the procedure works almost the same as a spatial filtering applied onto the image. A small window of kernel is selected on the fixed image, instead of the whole image. This kernel convolution mask moves from the top left of the moving image shifting to the whole image, finding the best match of the two images.
Usually CC is not suited for multi modality registration since a global linear transformation function of the grey values is insufficient. However if we square and accumulate the local CC values (taking positive as well as negative values into the account) then also multi-modality images can be registered. [1]

Mutual Information (MI)
It is based on the fact that proper registration means proper alignment of significant grey values that lead to pronounced peak in the joint intensity distribution.
Based on the following intrinsic advantages, this metric is considered to be a robust and vastly used similarity metric [1]:

- It is intensity based and can be used as an automatic registration metric.
- It is based on probability density estimation, therefore it utilizes the entropies of both images, which can help us to measure the misalignment and misregistration quite easily.
- It best suited for the 3D/3D multi-modality registration.

The main idea briefly relies on the fact that each point in one image will correspond to a point in the other and these two points each have image intensity associated with them which form the Joint Intensity Histogram (JIH). Then we can compute the joint Probability Distribution Function (PDF) through the following formula:

$$\frac{JIH}{Number\ of\ Pixels} = PDF$$

For each pair of intensities belonging to both images, the Joint Probability Distribution (JPD) is a number equal to the probabilities occur together at corresponding locations in two images.
Hence the joint probability distribution can be used to measure the disorder and increase of randomness in the joint histogram of the images whose irregularities are considered to be the entropies. When the entropies are high we would have misregistration due to uncorrect alignment and the registration is not satisfactory. Therefore one of the metric for evaluation of the final registration procedure is to minimize the joint entropy calculated from JDP. [9]

$$H(A) = -\sum P(A)_i \log P(A)_i$$

H(A) is the information supplied by the i symbols and is called entropy. P_i is the probability of occurrence for each symbol i.

For two images of A and B we have two symbols, hence the joint entropy comes as follows: [9]

$$H(A,B) \ll H(A) + H(B)$$

The more similar the images are, the lower joint entropy two symbols have.

Theoretically the entropy reaches to its maximum value if all the symbols have equal probability of occurrence, on the contrary the entropy is less if probability of one symbol is equal to 1 and all other symbols have the probability equal to 0.

In the application of registration for the aligned images we have two symbols at each voxel location. One belonging to fixed image (A) and the other belongs to deforming image (B). Consequently, MI can be computed using joint entropy of two images by one of the either the following equations:

$$(1)\ MI = -\sum_{j,k}\left(\frac{P_{j,k}^{2D}}{V}\right)\log\frac{P_{j,k}^{2D}}{P_j^1 P_k^2}\ ,\ [1]$$

$$(2)\ (X|Y) = IG(X|Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)\ ,\ [10]$$

$$(3)\ H(X|Y) = \sum_{x\in S_x, y\in T_y} P(X=x, Y=y)\log P(X=x|Y=y) = -\sum_{i=1}^{N}\sum_{j=1}^{M} p_{ij}\log p_{(i|j)},\ [10]$$

In (1) V denotes the volume of overlap, P_j^1 and P_k^2 are the probabilities of grey values j and k in the two images respectively, and $P_{j,k}^{1,2}$ is the probability that grey values j and k occur in the fixed and at the corresponding position in the transforming image.

(2) which is also called Information Gain in classification theory, compute the MI using direct consition joint entropy. However (1) and (2) can be easily converted to each other. (3) represents the conditional entropy between two symbols of X and Y, and $P(X=x|Y=y)$ denotes the conditional probability.

Usually misregistration makes the bright regions of the joint histogram less bright. Also reduced highest values and reduced number of zero in PDF lead to a misregistration because of increase in the joint entropy. Therefore to seek a best registration by using MI we have to seek for the transformation that produce small number of PDF elements with very high probability and gives us many Zero possibility elements as possible in the PDF. This causes to decrease in joint entropy.
Having used the MI a pre-processing image enhancement filtering ought to be applied, since MI is based on the histogram and probability density estimation and is susceptible to noises.

1-2-3-Regularization functions

As briefly mentioned before, the regularization term represents the irregularities of the final deformation field usually denoted by some function of the spatial derivatives of the field. Regularities in the deformation field imply the transformation is physically reasonable and/or consistent with physiological anatomical deformation. (Figure 1-8)

The problem with many image registration techniques is that the image similarity function does not uniquely determine the correspondence between two image volumes. In general, similarity cost functions have many local minima due to the complexity of the images being matched and the dimensionality of the transformation. [11]

A regularization term is usually added to the cost function such as described in [9]. Another reason for adding regularization term is because non-linear image registration is an ill-posed problem and adding additional term to the equation represents a trade-off to achieve optimal registration parameters. Depending on the type of regularization used, regularity of the deformation guarantees the smoothness and invertiblility of the deformation.

There are several regularization functions used in the literature.
The aim is to minimize the weighted sum of the below equation. The purpose of the regularization function $C_{smooth}(T)$ here is to maintain the quality of deformation field such that the deformation is an injective mapping. $C_{similarity}\left(I(t_0), T\left(I(t)\right)\right)$ is the similarity function between $I(t)$ and deformed $I(t_0)$ and T denotes the transformation. γ is the tradeoff between two terms.

$$C(\theta, \varphi) = -C_{similarity}\left(I(t_0), T\left(I(t)\right)\right) + \lambda C_{smooth}(T)$$

In the following part short description of some regularization function examples are presented:

- "Curvature term" regularization function, which instead of gradient it utilizes the Laplacian squared of the deformation field at each iteration of the optimization.

- "Elastic energy function" [12] which by using continuum mechanical model as linear elasticity, it forms the regularization parameter, and the "volume preserving penalty term" [13] which uses local volume preserving incompressibility and is based on the fact that soft tissues is incompressible for small deformation and short time period. Also there is "viscous fluid" constraint type which is almost same equations as for the elastic model but applied to the velocity field instead of the displacement field. [6]

- Mechanical bending energy of the transformation are computed in many literatures [9] which is represented by sum of the spatial second order derivatives of the transformation as the regularization term. This is computed using the jacobian of the transformation. This type of regularization constraint makes the final deformation field smooth and dismisses the cross-vector phenomena over the vector field.

- Inverse consistency of the registration has been evaluated in [12] since the author states that using non-landmark base image registration can encounter to several local minima and is not stable. Hence it utilize the jacobian of forward

and reverse transformation at each iteration of the optimizer to secure the non-singularity of the transformation and guarantee that the jacobian determinant does not lead to negative at all.

- Incompressibility constraint penalty is defined as the integral of the logarithm of the jacobian determinant of the transformation. [13]

Generally the purpose of the regularization constraint is to ensure that the transformations maintain the topology of the both images fixed and moving. In this thesis we have defined a regularization constraint based on vector calculus operator which profits the gradient of the transformation at each direction in order to not only guarantee the smoothness of the transformation by taking the derivative of the cost function containing MI as the similarity metric and divergence and curl of the deformation field, with subject to the transformation of the cubic Bspline polynomial but also to compensate the irregular features resulted from the wrong transformation.

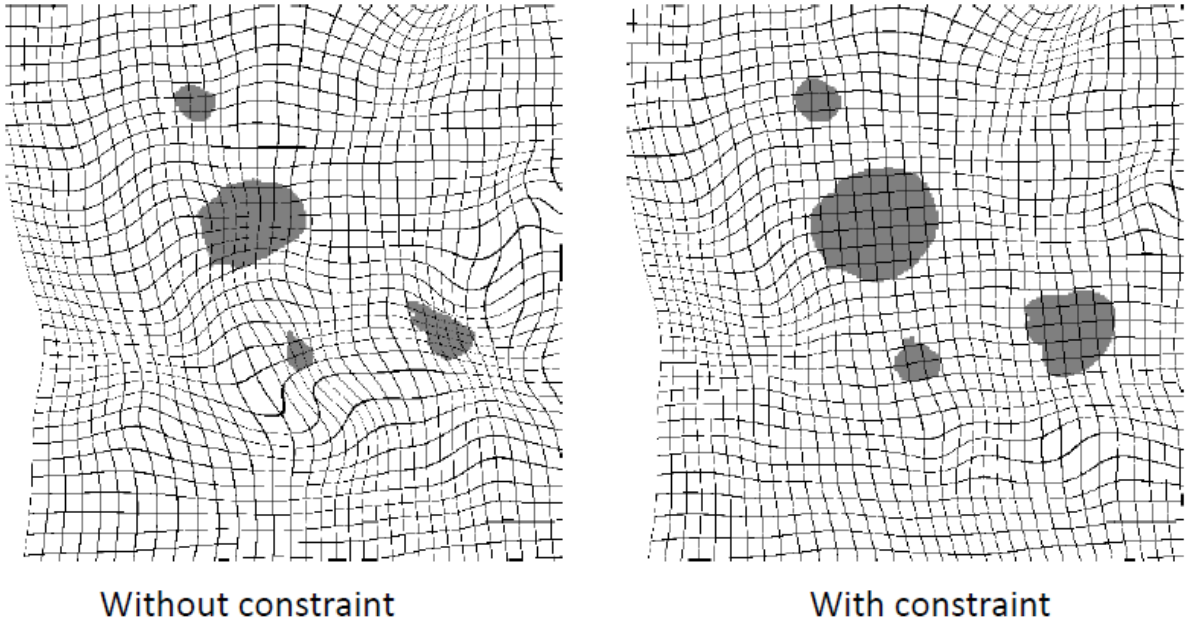


Figure 1-8. Deformation field with regularization term applied and without regularization term.

1-2-4-Optimization procedure

As mentioned before the problem in image registration is to find an optimized transformation that best matches and aligns the fixed and moving images through a maximized similarity measure. If we parameterize the geometric transform the registration problem becomes a parameters estimation problem. For solving this problem we need an optimizer to optimize (maximize/minimize) the cost function.

In the whole procedure of the image registration process, optimizers play a very crucial role for obtaining a well aligned final warped image. The final value obtained from the cost function is passed to the optimizer at each iteration to be compared to the stopping criteria defined for the optimizer.

Depending on the type of optimizer one uses, there are several types of criteria for continuing or stopping the registration process at each stage. In the next chapter, some commonly anticipated stopping criteria for the regular gradient descent optimizer are reported. But the one criterion that is common to all the optimizer is the “number of iterations” assigned to the optimizer before running the registration. This criterion is also considered as one of the most influential registration parameter.

In the following part some of occasionally used popular optimizers are presented:

“Steepest gradient” is one of the vastly used optimizers in the literature because of its naïve and simple approach and implementation. We have also profited this optimizer in this thesis. The main idea is very simple. Since the gradient of a function points towards rapid changes in terms of function value, at each iteration the minimum/maximum of the cost function is found by following the slope of the function itself. At each iteration of the optimizer, in addition to final value of the cost function, the derivative of the cost function is also checked in comparison to the stopping criteria. Hence the derivative of the cost function is also considered as one stopping criterion. Below you can find an algorithm used by D. Rueckert et al [9] to perform the registration process by using the gradient of the cost function $\frac{\partial C}{\partial \varphi}$.

calculate the optimal affine transformation parameters Θ by maximising eq. (7)

initialize the control points Φ.

repeat

calculate the gradient vector of the cost function in eq. (8) with respect to the non-rigid transformation parameters Φ:

$$\nabla \mathcal{C} = \frac{\partial \mathcal{C}(\Theta, \Phi^l)}{\partial \Phi^l}$$

while $||\nabla \mathcal{C}|| > \epsilon$ **do**

recalculate the control points $\Phi = \Phi + \mu \frac{\nabla \mathcal{C}}{||\nabla \mathcal{C}||}$

recalculate the gradient vector $\nabla \mathcal{C}$

increase the control point resolution by calculating new control points Φ^{l+1} from Φ^l.

increase the image resolution.

until finest level of resolution is reached.

Figure 1-9. Deformable registration by using gradient descent optimizer

In the above description, θ represents the affine transformation parameter for a rigid registration and φ represents the BSpline transformation parameters.

Elaborated concepts about gradient descent are described in the next chapter of method and implementation.
The big drawback of local methods specially regular gradient descent is that they get stuck in local minima while searching for global optimization. We have introduced a

nested optimization algorithm for this problem whose elaborated description is reported in the next chapter.

“BFGS optimization method”
This optimization method is a type of Quasi-Newton method which finds the minimum or maximum of a function. In Quasi-Newton methods the hessian matrix of the function, which is the square matrix of second order partial derivatives, does not need to be computed. The BFGS method has been suggested independently by Broyden, Fletcher, Goldfarb, and Shanno, in 1970 and includes a low-memory extension (L-BFGS). In the BFGS method the condition for the optimality is the gradient of the function be equal to zero, and the optimization process does not converge until the function has a quadratic Taylor expansion near an optimum value.

“Genetic Algorithm”
Genetic algorithms belong to a larger class of evolution algorithms which are used to generate and check useful solutions to optimization problems. In genetic algorithms a population of candidate solutions evolves toward better solutions. The evolution is based on a stochastic iteration and usually starts with a random initial population. At each iteration the solutions corresponding to each population are checked to fit the problem, and new solutions are generated for the next iteration until the fitness of the solution is satisfactory, or the algorithm reaches a maximum number of generations.

“Particle Swarm Optimization”
PSO is a new population based evolutionary computed optimization algorithm. The idea is that considering the whole population as a swarm, each node in the population is denoted as a particle. Since the computation algorithm is evolutionary each swarm can be divided into smaller clusters following the top-down strategy, or they can be joined together as a larger cluster following the bottom-up method.
Each particle in the swarm moves to the path of optimal decision on behalf of the whole swarm. Since each particle undergoes its own decision process, it can revise its decision based on the neighborhood. PSO is based on the fact that each particle evaluates its decision by comparing its current experience with its own past experience and experiences from the other particles, until it can reach to the best solution. [15]

1-3- Divergence and Curl vector spline operators

1-3-1- Introduction

As described in the part of regularization functions, there are several functions of the final deformation field (transformation) or its derivative to disregard the irregularities resulting from the misalignment of the similarity metric at each iteration. These functions would remove or compensate undesired irregularities in the deformation field to make it smooth. One of the regularization functions that we have defined in this thesis is using vector calculus of divergence and curl operator of the vector field in the cost function.

As a general theoretical definition divergence of a vector field could be understood in viewing the vector field in fluid, gas or flow. Divergence is defined as the dot product of the ∇ nabla operator with the defined vector field $T(x, y, z)$.

$$Div\ T = \nabla . T = \left(\frac{\partial}{\partial x}\right), \left(\frac{\partial}{\partial y}\right), \left(\frac{\partial}{\partial z}\right) . (x, y, z)$$

$$Div\ T(x, y, z) = {\partial T_x}/{\partial x} + {\partial T_y}/{\partial y} + {\partial T_z}/{\partial z}$$

Therefore, by definition, divergence is defined as the summation of derivative of x,y and z direction of T over the x, y and z direction parameters. The final output of the divergence is a scalar function in a scalar field. Divergence measures the expansion or compression of an object in the field.

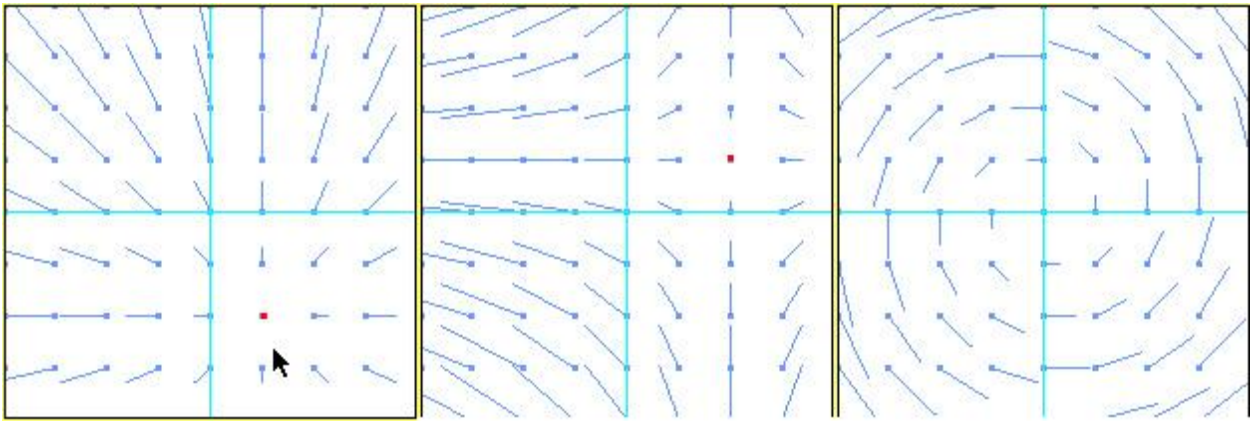


Figure 1-10. Left: represents the sink discovered by applying divergence. Middle: represents source discovered by applying divergence. Right: represents the centroid of the rotational vectors discovered by the curl operator of the deformation field.

Figure 1-10 illustrates the final role of divergence and curl operator of the deformation field. The left image represents the sink discovered by applying the divergence operator. The middle image represents a source discovered by applying divergence and the right image represents the centroid of the rotational vectors discovered by the curl operator of the deformation field.

As can be seen from the above figures during the registration process, the dilation of the vectors is obvious and widening and expansion has brought a contrast with compression or narrowing by applying the divergence. Thus divergence is an operator which specifies change in scale.

Likewise divergence tells us the presence of the sinks and sources on the flow or image. A vector field with zero divergence (null) at every point is called solenoidal. Practically, the values of the divergence bigger than zero are considered as sources, and smaller than zero are considered to be sinks as well. The divergence theorem states that theoretically the summation of all the density of the deformation sources and the sinks must be equal to zero.

$$\frac{\sum Div\, T > 0}{X_n \times Y_n \times Z_n} : density\ of\ the\ deformation\ of\ the\ Source$$

$$\frac{\sum Div\, T < 0}{X_n \times Y_n \times Z_n} : density\ of\ the\ deformation\ of\ the\ Sink$$

$$\frac{\sum Div\, T = 0}{X_n \times Y_n \times Z_n} : density\ of\ the\ Solenoidal$$

Divergence Theorem:

$$\frac{\sum Div\, T > 0}{X_n \times Y_n \times Z_n} = \frac{\sum Div\, T < 0}{X_n \times Y_n \times Z_n}$$

Consequently if the divergence at a region is large, then we have some sources of big deformation, meaning it is spread out. The divergence is then equal to the total amount of fluid or deformation being added or removed at that region. The total amount of magnitude or density of the deformation is shown at a point using divergence at that point divided by the volume of the region.

The curl of a vector field captures the idea how a field may rotate. It shows the circulation density of the field as represented by the cross product of the ∇ nabla operator and the vector field T.

$$Curl\, T = \nabla \times \mathrm{T} = \begin{matrix} \mathrm{i} & \mathrm{j} & \mathrm{k} \\ \frac{\partial}{\partial \mathrm{x}} & \frac{\partial}{\partial \mathrm{y}} & \frac{\partial}{\partial \mathrm{z}} \\ \mathrm{x} & \mathrm{y} & \mathrm{z} \end{matrix}$$

$$Curl\, T(x,y,z) = \left({}^{\partial T_z}/_{\partial y} - {}^{\partial T_y}/_{\partial z}\right)i + \left({}^{\partial T_x}/_{\partial z} - {}^{\partial T_z}/_{\partial x}\right)j + \left({}^{\partial T_y}/_{\partial x} - {}^{\partial T_x}/_{\partial y}\right)k$$

Hence by definition in contrast with divergence, the output of the curl of T is a vector field function.
If we assume a region of a fluid, then we look for the flow along the slides of the region. Sum of the flow rate along the slides denotes the curl F. Curl F specifies the change in orientation and vorticity, and a field with zero vorticity is called irrotational field.
Since the final output is a vector function, hence it has both magnitude and direction. In the above equation of curl, the first term represents the circulation in x direction, the secons term shows the direction of circulation in parallel to x-z plane and the third term denotes the direction of rotation in parallel to x-y plane. Magnitude of the CurlT is the speed of rotation and the direction is its trajectory of the rotation.

Part of the Helmholtz's theorem states that in an unbounded region both divergence and curl of the vector field are assumed to be vanished at infinity. The detail of this theorem has been brought in the next part.

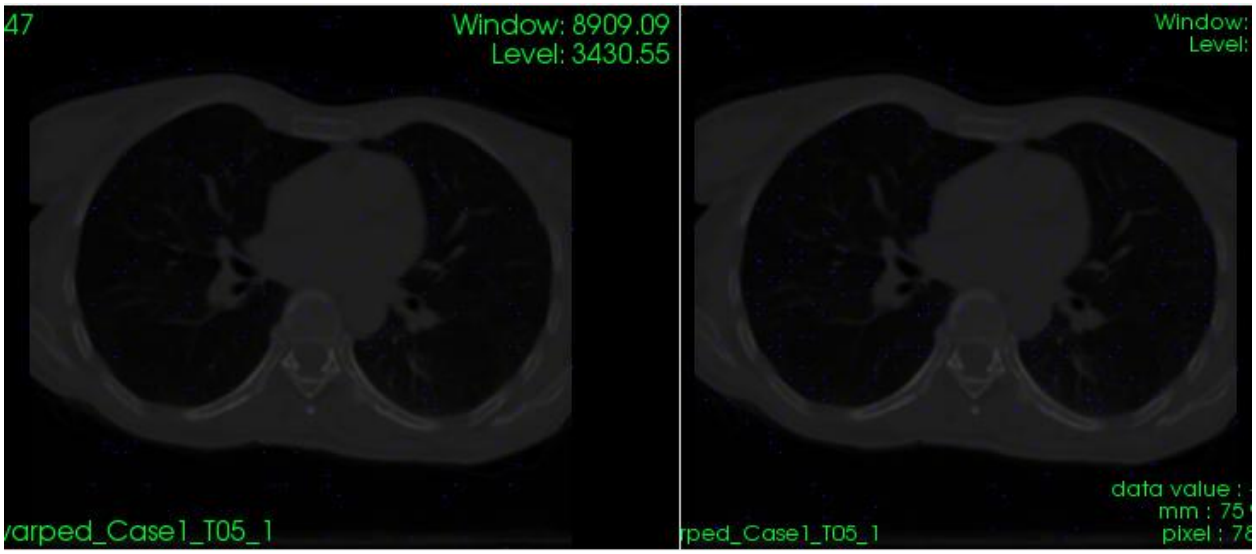


Figure 1-11- Left: the final warped image which divergence of the deformation field is overlaid using VV medical software application. Right: overlaid curl of the deformation field. The blue dots represent the points which in case of divergence sinks or sources are located and for the curl the centroid of rotational vectors are located.

In the above figure 1-11, we penalize vector fields with many points like the blue dots. The blue dots represent the points which in case of divergence sinks or sources are located and for the curl the centroid of rotational vectors are located. Thus divergence and curl force the deformation to be smooth.

As a result, to find the mapping between two images, one must optimize the similarity measure and at the same time to maintain the regularity of the transformation. The operators of divergence and curl try to achieve the consistency by forcing the deformation field to be a diffeomorphism (continuous, differentiable, and invertible, its inverse must also be continuous and differentiable). In principal, deformation based on

BSpline is not invertible, these operators overcome the singularity and non-consistency term to have a complete symmetric registration. These regularization functions turn out to give us a smooth deformation field.

1-3-1- Current state of the art

There are very few works that have been done in the concept of vector splines regularization and specially divergence and curl operators. In terms of analytical theory these operators can be stated and defined quite easily especially in the mechanical concepts. The major reason of difficulty in implementation is that these operators lead to a much larger value during the registration comparing to a similarity metric value such as mutual information. Also since divergence and curl of the deformation field must be obtained at each iteration of the optimizer, hence resampling the large amount of parameters of div/curl need a wise and optimized strategy and also it affects the computational time very sensitively.

In the following we review some contributions which have utilized the divergence and curl operator of the deformation field in their cost function to perform non-rigid registration. The criteria we used to select the contributions are based on how directly the vector fields are employed in the cost function and also how the theorem related to divergence and curl are utilized and have come into the implementation.

1-3-1-1-Inverse consistency registration (ImageJ)

Ignacio Arganda-Carreras et al [16] presented a unique application through ImageJ software dedicated to image analysis and development in Java for an elastic image registration algorithm under the name of bUnwarpJ application. This application is embedded into the software of ImageJ as a plug-in to perform a non-rigid inverse consistent registration. (Figure 1-12)

The whole idea lies on combining BSpline based consistent and elastic registration using vector splines [17]. This application is considered as the very rare software which has the power of introducing the divergence and curl to the user. In the designed user interface there are some options to assign weights to the div-curl operators. The way of choosing the weights is mentioned to be obtained empirically and there is no clear rationale for defining them. The default values are set to 0.1 for div as well as curl. (Figure 1-13)

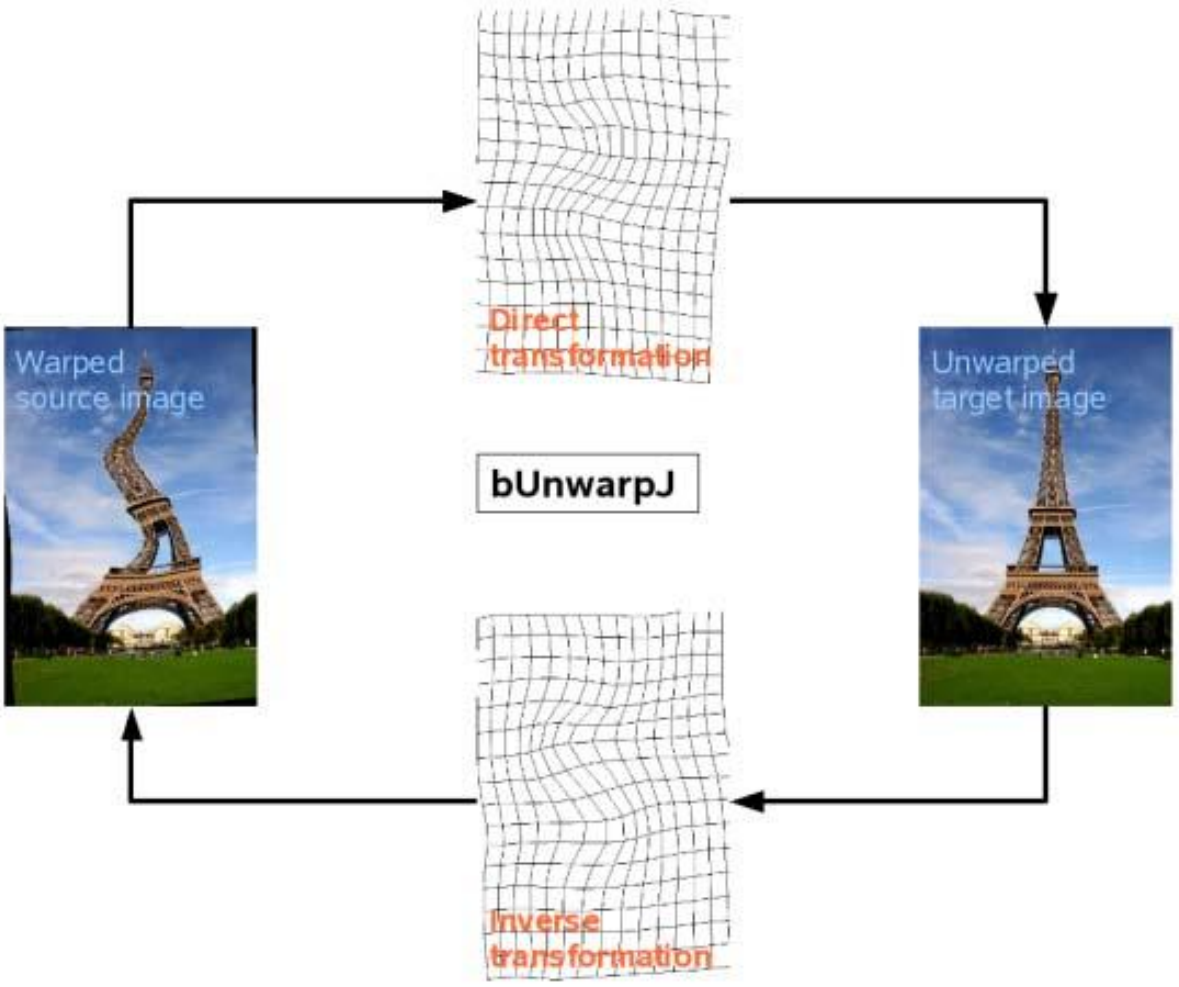


Figure 1-12. Scheme of the direct-inverse registration using bUnwarpJ.

The cost function defined for this application contains E_{img} to measure the dissimilarities between the two fixed and moving images, E_{cons} which is the energy term to calculate the geometrical differences between pixel coordinates of both images after applying both transformations (direct-inverse and inverse-direct). E_{μ} is the term optionally defined for the landmark registration and finally $(E_{div} + E_{rot})$ represents the divergence and curl operator of the vector field.

$$E = w_i E_{img} + w_\mu E_\mu + (w_d E_{div} + w_r E_{rot}) + w_c E_{cons}$$

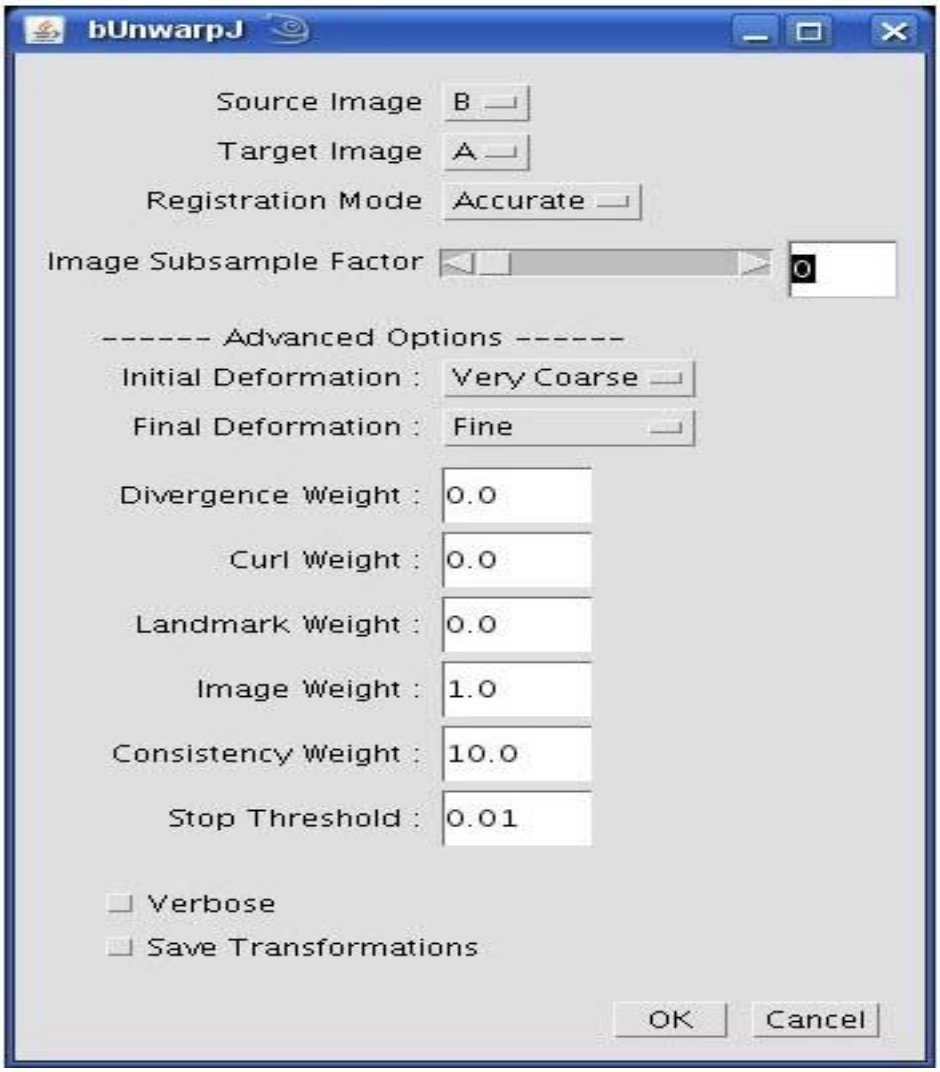


Figure 1-13. User interface of bUnwarpJ to select weights for the div-curl

The main advantage of bUnwarpJ is the fact that, as can be seen from the above figure, in the plug-in the selection of the weights for every terms of the cost function is arbitrary, and if one does not need for instance the inverse consistency or the regularization term, he can dismiss the weights to 0 and perform pure BSpline only registration. The images and the deformation fields both are represented by the BSpline transformation function and the optimization of the cost function is done by Levenberg-Marquardt minimization enhanced by a Broyden-Fletcher-Goldfarb-Shanno (BFGS) estimate of the local Hessian of the goal function.

A multi pyramid registration method is used with the deformation precision from 2^0 x 2^0 = 1x1 interval of BSpline coefficient to 2^4 x 2^4 = 16 x 16 interval, which basically means that more BSpline grid points lead to a more precise registration gradually. Although the drawback of the application is the fact that it is only designed for 2D images and there is no option to distinguish the weights for coarse and fine stage registration.

1-3-1-2-Helmholtz's Theorem Div-Curl solver

HSI-YUE S. HSIAO [18] in his PhD thesis used Helmholtz's method to perform a non-rigid registration. Helmholtz's theorem states that, with suitable boundary condition, a vector field is completely determined if both its divergence and curl are specified everywhere. In an unbounded region, both the divergence and the curl of the vector field are assumed to be vanished at infinity. While in a bounded region, suitable boundary conditions are required to uniquely determine the vector field.

In his PhD thesis, two approaches of the registration have been investigated. First the parametric elastic registration with BSpline parameters to characterize the deformation field at each control grid points is analyzed. This approach is very likely to result in the grid folding and irregularities in the final deformation field if the distance between the adjacent knot points is less than 8. This implies that the high frequency components of the deformation field can not be accurately assessed.

The second approach is to use the Helmholtz's decomposition to use the divergence and curl of the deformation field only on the grid points.
Helmholtz's decomposition states that any sufficiently smooth vector field rapidly decay at infinity can be resolved into irrotational (curl-free) and solenoidal (divergence-free) component vector fields, i.e., given a vector field φ, based on the Helmholtz's decomposition, can be decomposed into φ_i (curl free) and φ_s (divergence free) such that,

$$\begin{cases} \nabla.\varphi_i = \nabla.\varphi \\ \nabla \times \varphi_i = 0 \end{cases} \quad and \quad \begin{cases} \nabla.\varphi_s = 0 \\ \nabla \times \varphi_s = \nabla \times \varphi \end{cases}$$

Based on the Helmholtz's decomposition, we can sum φ_i and φ_s to obtain a div-curl system in 2D, i.e., $\varphi = \varphi_i + \varphi_s$

$$Div\ \varphi = div\varphi_i = \frac{\partial \varphi_{ix}}{\partial x} + \frac{\partial \varphi_{iy}}{\partial y} = \nabla.\varphi_i$$

$$curl\ \varphi = curl\varphi_s = -[\frac{\partial \varphi_{sx}}{\partial y} - \frac{\partial \varphi_{sy}}{\partial x} = -\nabla.\varphi_s$$

The similarity metric used is the sum of square differences along with the regular gradient descent optimizer. To perform a multi-resolution registration started from the coarse grid to fine grid. In the following flowchart the whole procedure is illustrated.

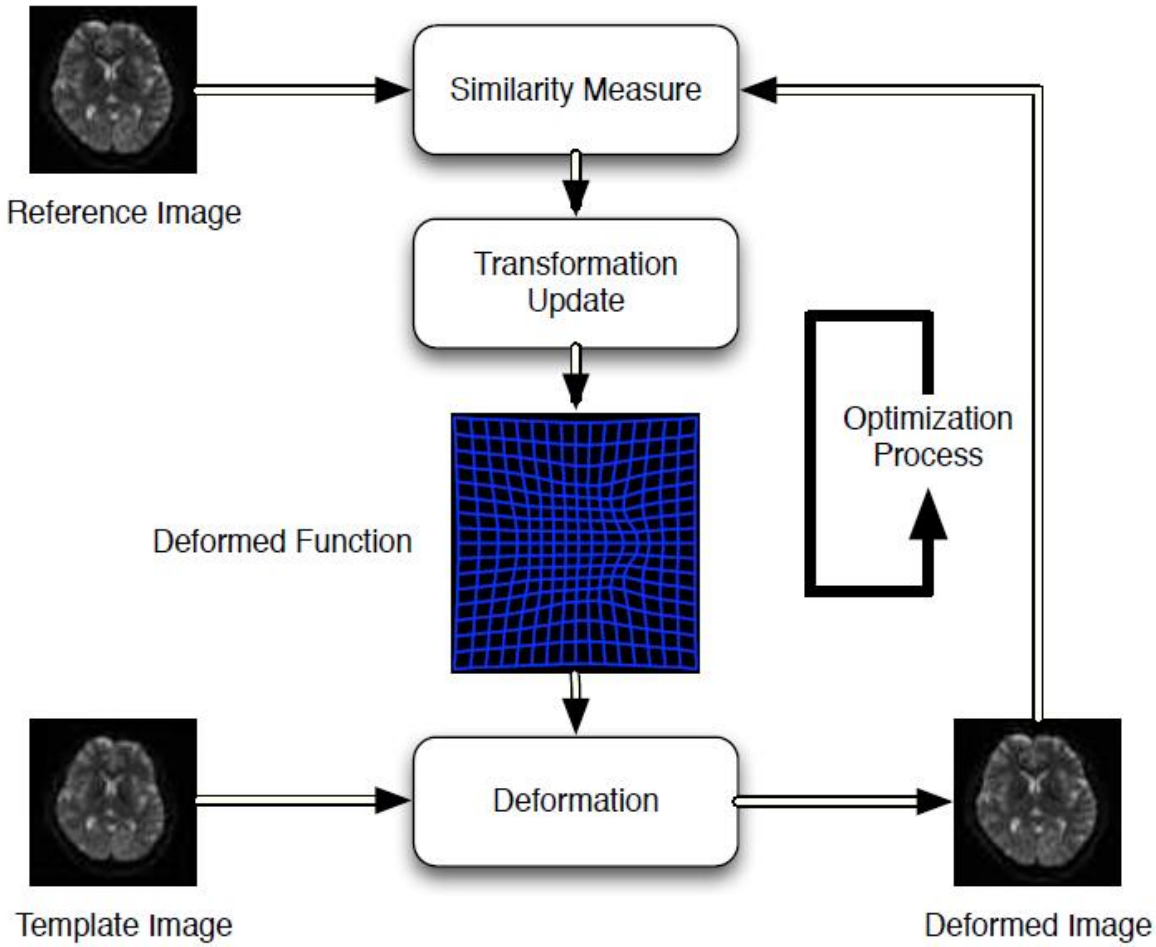


Figure 1-14. The procedure of the registration using Helmholtz's theorem to generate the transformation.

The advantage of this method is to assign the computation of div-curl operator only on the BSpline coefficients not to all the points. This leads to a very time saving in term of computation. Also this method can be extended to any dimension of the image at least theoretically and the calculation of the gradients is quite simple.

The disadvantage of this method is that the mesh folding or irregularities in the final deformation field is not guaranteed and the implementation of 4D images non-rigid registration to 3D or 3D/2D registration using this method is not trivial at all.

1-3-1-3-Grid deformation method

In this work MEHMET ALI AKINLAR [19] used a method (deformation based grid generation) which is able to generate a grid with desired grid density distribution that is free from grid folding. This method gives direct control over the cell size of the adaptive grid and determines the node velocities directly. The adaptive grid system naturally distributes more grids to deprived areas.

The divergence operator is used to maintain the local volume changes in sink and sources whereas the curl operator of the transformation to control the local rotation of the deformation field.
The grid generation method (GDM) is used for construction of differentiable and invertible transformations to solve mesh adaption problems. A moving-grid algorithm is formulated using the deformation method. The idea of this method is to move the nodes with correct velocities so that the nodal mapping has a desirable Jacobian determinant.
The main idea in this dissertation is that it uses the divergence and the curl operator in the phase of the grid generation. Meaning it utilizes the div-curl of the vector filed in the assumption of constructing the mapping of the registration.

Also in this method sum of squared metric is used to measure the dissimilarities between the images along with the Levenberg-Marquardt optimizer.

The inventive point in this work is that the div-curl are not used as the regularization term instead they are used to obtain the deformation field as the assumption of the mathematical framework. Thus this work only has anticipated with one term of the cost function.
The invertibility of the transformation is also included into the mathematical framework that is defined for obtaining the deformation field. In fact the determinant of the Jacobian of the transformation has been taken as another assumption of the framework.

The main advantages can be summarized as follows:

- The method is based on a linear differential system; its numerical implementation is fast, stable, simple and robust.
- It does not require any regularization term.
- The method is general in the sense that it may be used in any optimization problem that involves motion estimation. Thus, it has the potential to be the numerical kernel for a wide range of applications.

The drawback of this method is that although the mathematical framework defined for performing the registration seems to be quite robust, but the fact that it does not have any term to maintain the unsmoothness of the transformation, makes the method susceptible to misregistration.
Likewise in term of implementation, the framework is not trivial and it is not straightforward. Using the defined mathematical framework the inverse consistency and the compensation of the final deformation field is not guaranteed in practice.

1-4-Deformable registration evaluation methods

1-4-1- Introduction and state of the art

Evaluation of non-rigid image registration algorithm and the final result are a hard task since point-wise correspondence between two fixed and warped or fixed and moving images are typically not known. Many researcher communities have been working to define and exploit a standard benchmark and a framework to assess deformable registration algorithms. However there is not a unique and singular application to evaluate all of the registration methods for all kind of images and modalities.

Generally there are two types of ways to evaluate the final database of registration results. The first one is to profit the analytical statistical measures to evaluate the differences between fixed and warped vs. fixed and moving image. Also the final deformation field (transformation) can be evaluated, either using a ground truth deformation field or by computing the jacobian determinant, inverse consistency, etc.

The second way is to use a standardized database to analyze and assess final results. In the following both methods are briefly described.

The following four main statistical methods are usually used to assess the performance of registration.

1- Structure overlap

This kind of statistics measures how well the labeled volumes or surfaces of source image and target image agree with each other after registration. In this method the percentage of overlap of the whole or part of the desired region of the image is calculated before and after registration. The major point which has to be taken care in this kind of evaluation is to use accurate contours for the evaluation. This is because usually a meta 3D/2D image contains extreme elements which are unnecessary for the evaluation of the only patient area. Hence those parts in the image must be masked or cropped. Thus defining a proper contour over the patient area of the image increases the accuracy of the final evaluation.

Some of these methods such as Percentage of overlap, edge overlap, etc., are investigated in the next chapter quite in detail both in term of theory and in term of implementation.

2- Intensity based differences error

These measure intensity difference between deformed and target intensity images. Examples of these errors include intensity variance (RMS error), mutual information and average volume method (mean and median).

3- Deformation field statistic error

In this method the specification of the final deformation field or the transformation are evaluated. In case of availability of the synthetic deformation field, the difference between the finally obtained deformation field of the registration and the synthetically applied transformation are evaluated.

4- Landmark error

These statistics measure the distance between deformed landmarks and corresponding target landmarks. Distance between two point sets can be measured using Euclidean distance, closest distance or any other suitable metric.

The other method to evaluate the registration performance is to use the benchmarks provided by the research communities for the assessment. These research groups provide the community with images to register and then evaluate the results. The “Retrospective Image Registration and Evaluation Project" [5] led by Jay West Fitzpatrick of Vanderbilt University took this approach to evaluate inter-modality registration algorithms. A common set of images were used to evaluate the performance of registration algorithms. Researchers registered the images with their own registration algorithms and then send an ASCII codes containing the original and transformed points back to Vanderbilt. Registration algorithms were evaluated using the target registration error.

Castillo et al. [6] evaluated deformable image Registration (DIR) spatial accuracy using large sets of expert-determined landmark point pairs. Each of their data sets has associated with it a coordinate list of anatomical landmark point sets which serve as a reference of evaluating DIR spatial accuracy within the lung. They provide published DIR spatial accuracy results on their website (http://www.dir-lab.com). Results are reported as mean 3D Euclidean magnitude distance between calculated and reference landmarks.

If ever there is rarely a “Gold Standard" or “Ground Truth” correspondence map that could be a best way to judge the performance of a registration algorithm.

The website (http://www.vmip.org/) set up by Pierre Jannine directs people to papers and references about validation and evaluation in medical imaging processing, and a list of validation data sets.

Chapter 2
Methodology and implementation

2-1- Introduction

As mentioned in the preface part, the main aim of this thesis is to supply a framework in order to design and implement an image registration method which contains vector calculus weighted cost function using ITK libraries. The framework has been tested to empirically obtain the optimal registration parameters for parametric registration algorithms applied to adaptive radiotherapy. The optimization process will consider both the similarity of the images and the anatomical consistency of the transformation, which is required for radiotherapy applications.

For this purpose we implemented a multi-resolution image registration application with a multi-metric framework containing mutual information similarity metric and a vector spline regularization term based on divergence and curl of the deformation field. The multi-resolution technique is based on BSpline grid points, starting with a coarse grid to cover the global dissimilarities and moving to a fine to compensate local dissimilarities in the registration approach. (Figure 2-1)

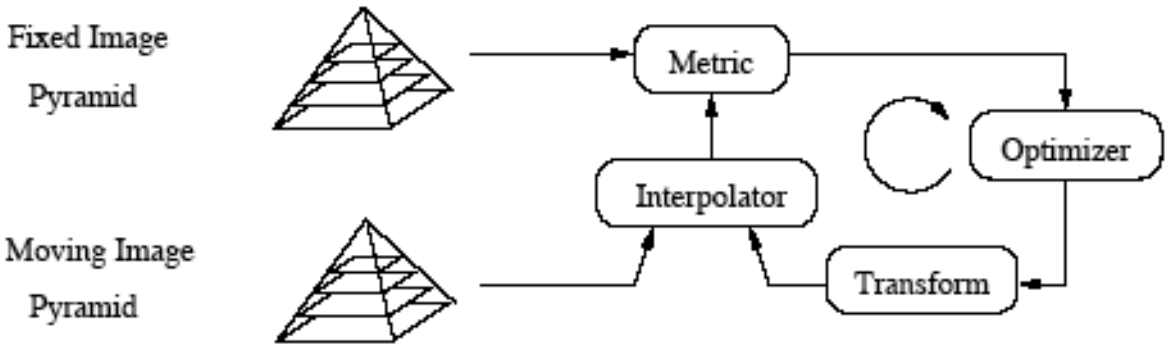


Figure 2-1. A schema for the multi-resolution registration. [4]

In the practical implementation we confined our registration algorithm to: Parametric registration of Bspline transformation functions by defining mutual information as the similarity measure and the gradient descent as the optimizer.

The major inventive point of our method is to integrate the divergence and curl of the deformation field into the cost function along with the mutual information as the similarity metric. This approach to the best knowledge has not been done by any other developer so far. Dan Ruan et al [1] utilized divergence and curl of the deformation field as the regularization term in image registration, limited to the Mean Square Error (MSE) as the similarity metric.
For this purpose we first implemented a stand-alone div-curl calculator that was later integrated into the cost function within a multi-stage registration framework. We needed to compute the div-curl of the deformation field at each iteration and add its value to the MI metric to be fed to the optimizer. Likewise we had to compute the derivative of div-

curl of the deformation field and sum the whole parameters to the MI parameters and return it to the gradient descent optimizer along with their values.
The major issue during the implementation was related to ITK multi-metric registration support. Since our method is designed as a multi-metric registration, we needed to implement a multi stage multi-metric registration framework by integrating the div-curl operator into the MI metric, while ITK does not support a multi-metric registration approach.
Moreover we have defined four types of weights the user can assign to the div-curl parameters before performing the registration process. Two weights are assigned for the coarse grid and two others are assigned for fine grid deformable stage, allowing to maintain the regularization terms of the cost function and obtain the desired deformation field. Optimal weights are obtained empirically Design Of Experiment (DOE) method base on observation experimental sensitivity assessment approach: a detailed discussion is presented in chapter 4.

In order to get started with our methodology, in addition to the definition of the components of the registration procedure, the particular properties or specification of each registration component must be confined in terms of the implicit influential registration parameters. These registration parameters are either defined by default while designing each component in the implementation phase or they are assigned by the user as input parameters.
The reason for restricting the registration parameters is because changing or regulating each of them would affect the final result. The results which we obtained are based on these particular parameters, hence the comparison to any other type of registration algorithm would have the obligation of setting the same pre-registration parameters.

Nevertheless, in chapter 3, we have regulated these pre-registration parameters (e.g. Number of grid points, subsampling resolution, number of iteration) to see which are the best settings based on our experimental data.

We have selected a parametric registration by using B-spline grid point coefficients to characterize the transformation function and to recover the warped target image. There are many criteria which influence the registration procedure and cause the final warped image to be well aligned. Types and specification of the images and modalities, type of transformation for constructing the deformation, type of optimization for optimizing the cost function, etc., are crucial for the result of the registration. Since registration is an optimization problem, one of the main concepts that varies among different registration techniques is the objective or energy function to be maximized/minimized in order to obtain the final optimal transformation parameters. Those parameters are applied on the moving/fixed image to be warped and meet the best similarity with the fixed/moving image. A cost function which includes the similarity between two images and a regularization term to penalize discontinuities could vastly affect the final registration result.

So far many combinations of similarity and regularization terms have been proposed and tested in the literature. There is not one specific function that works best for all kind

of registration problems. Especially it depends on type of image digitization, modalities of the images and type of definition for the intensity values in both images, which define what type of cost functions is suitable for the application.
In our application we have defined a cost function which is composed of Mutual Information as the similarity measure and two terms of regularizing functions which are divergence and curl of the transformation. These terms maintain the quality of deformation field, such that smoothness of the final transformation is guaranteed. Despite MI is usually used for multi-modality image registration, it is also applicable to mono-modality registration.
We also profited the regular gradient descent optimizer to maximize the cost function. It is done by differentiating all the terms inside the function, meaning the probability density estimation between the images and two vector calculus based penalty terms. Utilizing the gradient descent optimizer causes to simultaneously maximize the MI similarity measure and minimize the regularization function.

The following equation shows our cost function defined on images in a 3 dimensional space:

$$F(x, y, z) = E_{sim} + (w_d E_{div} + w_c E_{curl})$$

2-1-1- ITK characteristics and registration framework

In order to implement our defined cost function, we profited of an open-source and cross-platform library named Insight Segmentation and Registration Toolkit (ITK). ITK is a very large library provided for development of medical applications through dedicated programming methodologies. In ITK, functions that implement image processing are called “Filters”, i.e. templated C++ classes. These templates make the whole library portable when they are combined with CMake open-source building system and this union helps developers to integrate the library into their own applications.

While using ITK there are lots of examples for almost any kind of registration approach: we focus merely on the deformable registration examples. Consequently we picked an example as a sample to modify and revise it to fit our cost function and to integrate the implementation of the divergence and curl regularization.

2-2- Experimental Dataset

In this thesis we have tested two types of dataset. The first one comes from a radio-equivalent phantom of the head and neck. These images are synthetically deformed and thus the synthetic transformation is considered the gold standard. Therefore, we apply our method to the phantom images, then compare our obtained final deformation field to the synthetic deformation field and calculate the difference in term of statistical quantification.

Another dataset we employed to exploit our method, is the clinical real patient cases obtained from (www.dir-lab.com) [2]. The dataset is designed to rigorously and objectively assess the spatial accuracy performance of deformable image registration

components. Since the dataset are provided with pathology approved landmarks, we used them as a robust benchmark to evaluate our application. (www.dir-lab.com)
The dataset consists of 4DCT images of the lung: using this dataset we can evaluate our registration method in terms of similarities between the images and assess the final deformation field, but also we can try out the potential of inverse consistency of the transformation as well as evaluating the landmark errors.

2-2-1-Synthetic dataset

In this dataset the approach is to use a combination of simple synthetic transformation methods to derive quantitative measures on the performance of several nonlinear registration algorithms. We assume that the calculation of many different synthetic deformations, each of them testing a different behavior, along with a large number of different evaluation measures leads to a thorough evaluation and comparison. Part 3 in chapter 3 shows our results concerning this type of dataset assessing the both final warped images and deformation fields using several weights.

The image is scanned using CT and it is a 3D meta-image (.mha image). The main size of the image is 512x512x128, but since the computation time is quite important criterion, we resampled into 256x256x61 [2 2 1] and used this resolution as our prior specification.

Detailed specifications of the images are shown in table 2-1. Coarse grid and fine grid spacing are the spacing imposed between adjacent BSpline grid points and refer to multi-resolution registration approach.

Image type	Image size	Subsampling resolution	Coarse grid spacing (mm)	Fine grid spacing (mm)	Image spacing
Simulated phantom of Head and Neck	256x256x 61	[2 2 1]	[43, 43, 32]	[16, 16, 12]	[1.875 1.875 6]

Table 2-1. specification of the “rando-phantom images” we used in our experiments

Our basic strategy for the evaluation is in these cases identical. We take an original image, apply a synthetic transformation and store the synthetically warped image and the resulting displacement field. The displacement field will be our ground truth to be compared. Now each of the investigated nonlinear registration algorithms gets the synthetically warped image as fixed input and the original image as moving input. Then we try to find a displacement field that warps the original image to the synthetically transformed one. In this way we can finally compare the warped image with the synthetically warped image and the ground truth displacement field with the calculated displacement field. Note that only this way it can be guaranteed that the displacement fields represent transformations in the same directions (from fixed to moving image). In this case the deformations are obtained by using superimposition of 3 Gaussians with 3 different centers. (Figure 2-2)

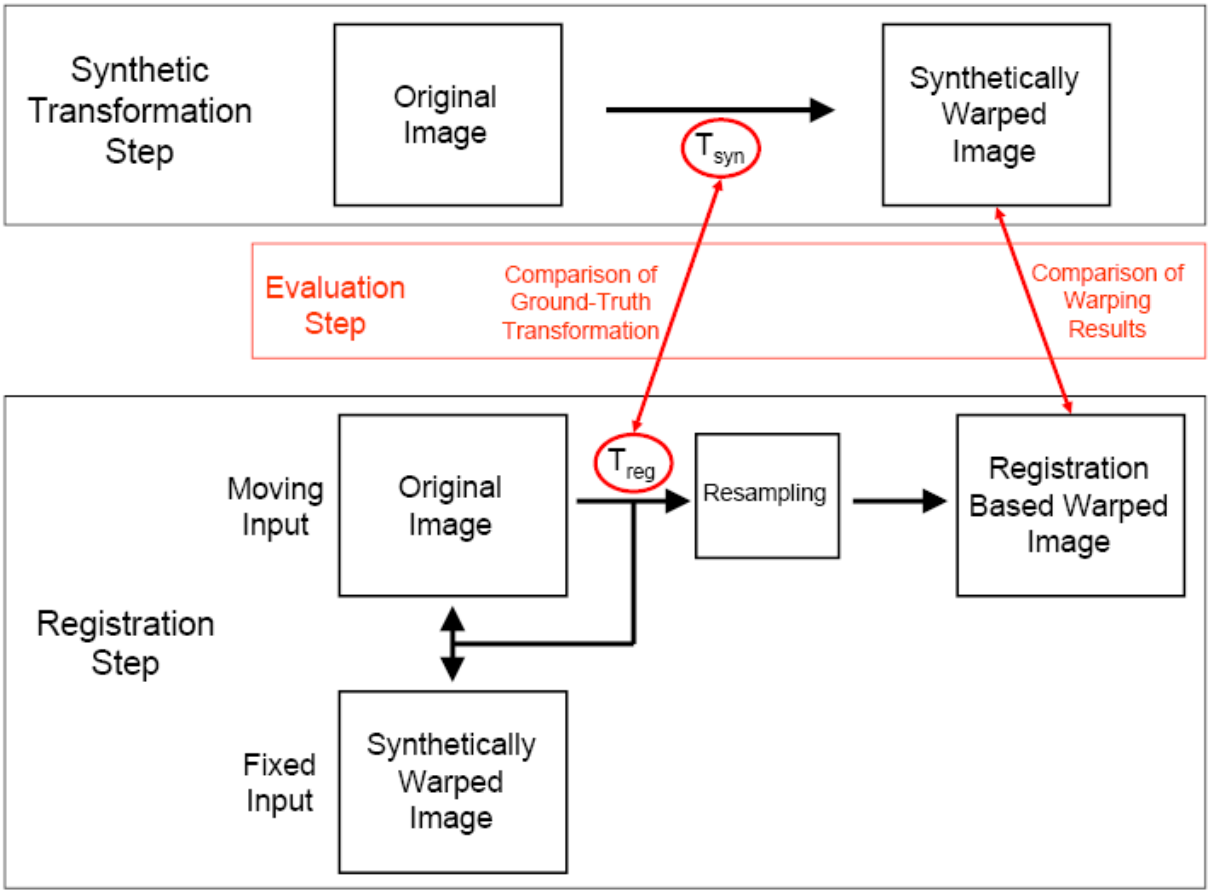


Figure 2-2. The basic setup for the synthetic transformation evaluation experiments. The synthetic transformations are applied to the original image, the nonlinear registration of original and synthetically transformed image leads to a transformation and a warped image which can be compared.

2-2-2-Real Clinical dataset

The images belong to lung 4DCTs, started from the inhale breathing phase (Case T00-T05) to the end of exhale phase (Case T05-T90). The type of images is mono-modal 4DCT containing coordinate information for the reference landmarks on the extreme phase images (T00 and T50). The database is very helpful since it comes with XYZ coordinate of the landmarks and the average landmark displacements taken from inhale to exhale phase. Hence it can be used for a very accurate registration assessment framework. We also have utilized this dataset to evaluate the inverse consistency of our method by inverting the role between the fixed and moving image. Part 3-4 explain the assessment framework of this clinic dataset in a thorough way.

Image type	Image size	Subsampling resolution	Coarse grid spacing (mm)	Fine grid spacing (mm)	Image spacing	Image type	Image size
Real patient images took from [6]	256x256x61	[2 2 1]	[15 15 15]	[33 33 33]	[43, 43, 32]	[16, 16, 12]	[1.875 1.875 6]

Table 2-2. specification of the "real patient clinical images" we used in our experiments

In the above table 2-2 the complete specification of this type of images are shown. In the repository of the website, there are 10 cases of 4DCT available to download. They are sorted based on the deformation they have between inhale and exhale. We chose three cases among to test out and assess our method. We chose the first cases with a minimum deformation (Case1) and Cases 7 and 8 with the largest deformation. In the following table 2-3 also the specification of cases 1, 7 and 8 are described.

Cases	Image Dimension	Voxel Dimension (mm)	Avg landmark displacement (mm)
Case 1 (T00-T09)	256x256x94	0.97x0.97x2.5	2.91
Case 7 (T00-T09)	512x512x136	0.97x0.97x2.5	7.87
Case 8 (T00-T09)	512x512x128	0.97x0.97x2.5	9.11

Table 2-3. Specification of the clinical cases that were selected to test our registration framework.

2-3- Implementation of Registration Components

By introducing the following components we are not trying to describe each component separately again as we did in chapter 1. We are strictly intending to follow the registration procedure we have done step by step in terms of development, by approaching the subsequent stepwise major components of the registration flowchart. It starts from the selection of transformation parameters, interpolation, role of the similarity metric and optimizer and it arrives to our major work, implementation of div-curl operator. Finally the method of implementation of multi-resolution registration is described.

2-3-1- BSpline transformation function and Interpolation

The theoretical details about BSpline function are completely described in chapter 1, part (1-2-1). Here we focus on the practical approach of BSpline control points and the interpolation of the vector field to obtain a 3D image.

The main profitable idea of the BSpline is that it employs the local compact support for each voxel on the image by using its grid points. The support region is rectangular-shaped and supports the deformation of the voxels in that region. Depending on the order of BSpline functions the support region is different. For cubic BSplines, the support region is (3+1)x(3+1)x(3+1). The deformation of the points that are not on the grid points are computed using "interpolators".

The interpolator will evaluate the intensities of the moving image at non-rigid positions. We have used the linear interpolator to intersect or connect those points that are not considered as the control points.

Therefore the deformation of any point in the image is influenced by the deformations of all control points in the support region, and is computed by interpolating them according to the following Equation.

$$T(x,y,z) = \sum_{l=0}^{2}\sum_{m=0}^{2}\sum_{n=0}^{2}[\theta_l(u)\theta_m(v)\theta_n(w)]\varphi_{i+l,j+m,k+n}$$

$$i = \left|\frac{x}{\partial}\right| - 1, \quad j = \left|\frac{y}{\partial}\right| - 1, \quad k = \left|\frac{z}{\partial}\right| - 1$$

$$u = \frac{x}{\partial} - \left|\frac{x}{\partial}\right|, \quad v = \frac{y}{\partial} - \left|\frac{y}{\partial}\right|, \quad w = \frac{z}{\partial} - \left|\frac{z}{\partial}\right|$$

In the above equations, θ is the n-th basis function of the BSpline, ∂ represents the spacing between the parameters. The most important parameters are the transformation parameters φ which are the mesh of the control points, whose place and localization are needed for the final calculation. These parameters are changed at every iteration dynamically by the optimizer. In fact the optimizer searches the best transformation parameters which are the optimum solution for our cost function.

The number of parameters produced by the BSpline function is a MxN array of parameters, where M is the number of nodes in BSpline grid points and N is the dimension of the space. For instance if the number of grid points is equal to 10, with a 3D image, the number of transformation parameters would be a matrix of 10x3. However by considering the border of the BSpline space the number of parameters is defined by a 13x3 matrix.

In term of implementation in ITK there is a very vastly used filter
"itk::BSplineDeformableTransform<CoordinateRepType,SpaceDimension,SplineOrder>"

As it can be seen this filter is a template over type of coordinate representation, which we have chosen "double", dimension of the image, which is 3D here, and spline order (cubic). The deformation field grid is defined by the user and it is specified by GridRegion, GridSpacing and GridOrigin functions. Each grid/control point has associated with it N deformation coefficients, representing the N directional components of the deformation. Deformation outside the grid plus support region for the BSpline interpolation is assumed to be zero. [3]
In order to consider the border of the grids there must be one extra node at one side of the image and two extra nodes at the other side and this applies to every direction. The border size is equal to the order of the BSpline function. Therefore when we state that the coarse grid of BSpline resided on the image is [12 12 12] in every direction, that means that we have to consider additional 3 grid nodes in every direction if we use cubic BSpline and it becomes [15 15 15]. This matter is also true for the fine grid stage. In terms of implementation this fact is done using the following functions.

NumberofGridNodesinOneDimensionCoarse = 12
Represents the number of grid points in low resolution coarse grid BSpline.

GridSizeOnImage(NumberofGridNodesinOneDimensionCoarse)
Report the number of grid nodes to the BSpline Transformation classes.

GridBorderSize.Fill (SplineOrder)
1 node for lower border and 2 nodes for the upper border.

Therefore the total grid size is

TotalGridSize = GridBorderSize + GridSizeOnImage

Before transforming the parameters from input point to the output point, the above functions must be set with their input parameters for both coarse grid and fine grid stage.

After setting the above necessary inputs the transformation parameters of the BSpline are set and the input point on the fixed image are calculated on the corresponding point on the moving image. Finally both points are passed to the cost function terms for the necessary operations of alignment assessment.

As mentioned before, in order to model and characterize the non-rigid regions which are not resided on the grid points, we use the Linear Interpolator because of its naïve and simple approach. The ITK filter is as follows:
itk:: LinearInterpolateImageFunction< MovingImageType, double>

2-3-2- Mutual Information Similarity Metric

Actually using MI as the similarity metric is not as trivial as using for instance Sum of Squared Differences (SSD) defined for voxel intensities of images. The major reason can be understood from the mathematical equations of the metrics described in chapter 1.

$$SSD = \sum_{i \in M} (I_1(i) - I_2(i))^2$$

$$H(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Default specification boundaries we employed to characterize our MI metric are presented in table 2-4.

Spec.\Reg. Type	Rigid	Affine	Coarse grid	Fine grid
Histogram bin size	64	64	64	64
Number of spatial samples	10000	50000	Number Of BSpline Parameters * 100	Number Of BSpline Parameters *number Of Pixels

Table 2-4. properties of the similarity metric between the images (mutual information) we used in our method

In SSD the role of the metric is to focus on extracting differences of intensities, hence it directly works out with the voxel magnitudes. This fact is a huge advantage specially when using gradient descent as the optimizer, because the user has to return the final value of the metric along with its derivative matrix at each iteration. Hence, if alongside the similarity metric we include a regularization function, then the metric has to return the gradient of both terms.
In our case we have the regularization function defined by divergence and curl of the vector field. These two regularization terms are directly correlated to values of the voxels. MI is instead not directly related to voxel's intensities but rather it requires to compute the joint distribution histogram and the probability density estimation to estimate the information gain between the images.

Therefore the gradients of these two terms are not in the same scale and they follow different ways of calculation. Hence we ought to compute them separately and then join them together in the same matrix of derivatives to pass them to the optimizer. The solution we used is to resample the div-curl parameters to bring them to the same scale of the MI.
In terms of implementation 90% of our work has been done in the ITKMutualInformation.cxx class, where the implemented div-curl regularization function was embedded.

The template ITKMutualInformation.cxx class is responsible for the following process:

- To compute the MI final value by using the related functions and filters of the histogram joint distribution and probability density estimation.
- To compute the derivative of MI as resampled parameters and return them to the optimizer.
- To set the transformation parameters passed form the main program to this class and then pass them to each of the metric terms and finally pass the computed values to the optimizer.
- To control all the parameters that must be returned to the optimizer.

We subsequently modified following one of major ITK filters (class) to calculate MI.

MattesMutualInformationImageToImageMetric<FixedImageType, MovingImageType>

This class has two very important major functions which control the computation of the metric and its derivative.

- Function “GetValue()”. This function computes the final value of the MI by using the aforementioned methods. If an additional regularization term is intended to be added to a similarity metric i.e. MI, SSD, etc., in a cost function, this function is significant. For designing a multi-metric cost function the major and critical changes must be done here in order to pass the summed value to the optimizer.
- Function “GetValueAndDerivative()”. This function is the crucial one when we use gradient descent optimizer or any other optimizer which uses the derivative of the cost function to compute the optimum final solution. So in this function the final value of the cost function and its derivative are computed.

In our case we changed the latter function, because we are using the gradient descent optimizer. This means that we pass the div-curl parameters computed in a separate function at each iteration to this function along with their derivative parameters.

In order to initialize the MI metric function, it is necessary to define the Number of Spatial Samples. It is done before the execution of deformable registration as follows.

Number Of Samples = Number Of BSpline Parameters * Number Of Pixels;
SetNumberOfSpatialSamples (Number Of Samples);

Using the above method, the number of samples is set for the metric. However this computation is only true for the deformable registration: for rigid and affine registration, we have to multiply an arbitrary value (e.g. 50000) to the number of pixels in the image. Also another function that has to be set for the initialization of the metric is the number of histogram bins. The default value for ITK is 50, but we have set it to 64 which is power of 2.

The main class in ITK which controls the metrics such as MI, SSD etc., is the “ImageToImageMetric” class. All other classes are derived using the inheritance property and if one is interested in writing his/her own metric, naturally this ad-hoc metric must be inherited from the latter class.

2-3-3- Regular Gradient Descent Optimizer

The role of the optimizer is very critical in image registration algorithms. It looks for the optimum solution i.e. the transformation parameters that best align the fixed and moving image.

We have used regular gradient optimizer which simply takes the derivative of the cost function at each iteration. The default specification of the optimizer we used is presented in table 2-5.

Step length\Registration Type	Rigid	Affine	Coarse Grid deformable	Fine Grid deformable
Optimizer max step size	0.2	0.2	10	10
Optimizer min step size	0.0001	0.0001	0.01	0.01
Relaxation Factor	0.7	0.7	0.5	0.5
Tolerance	0.0000001	0.0000001	0.00000001	0.00000001
Number of iteration	200	200	500	500

Table 2-5. properties of the optimizer of Gradient descent we used in our method

The problem of getting stuck in a local minimum was minimized by profiting of a multi-resolution registration approach.
By this approach the maximum of the cost function is found by following the slope of the function. (Figure 2-3)

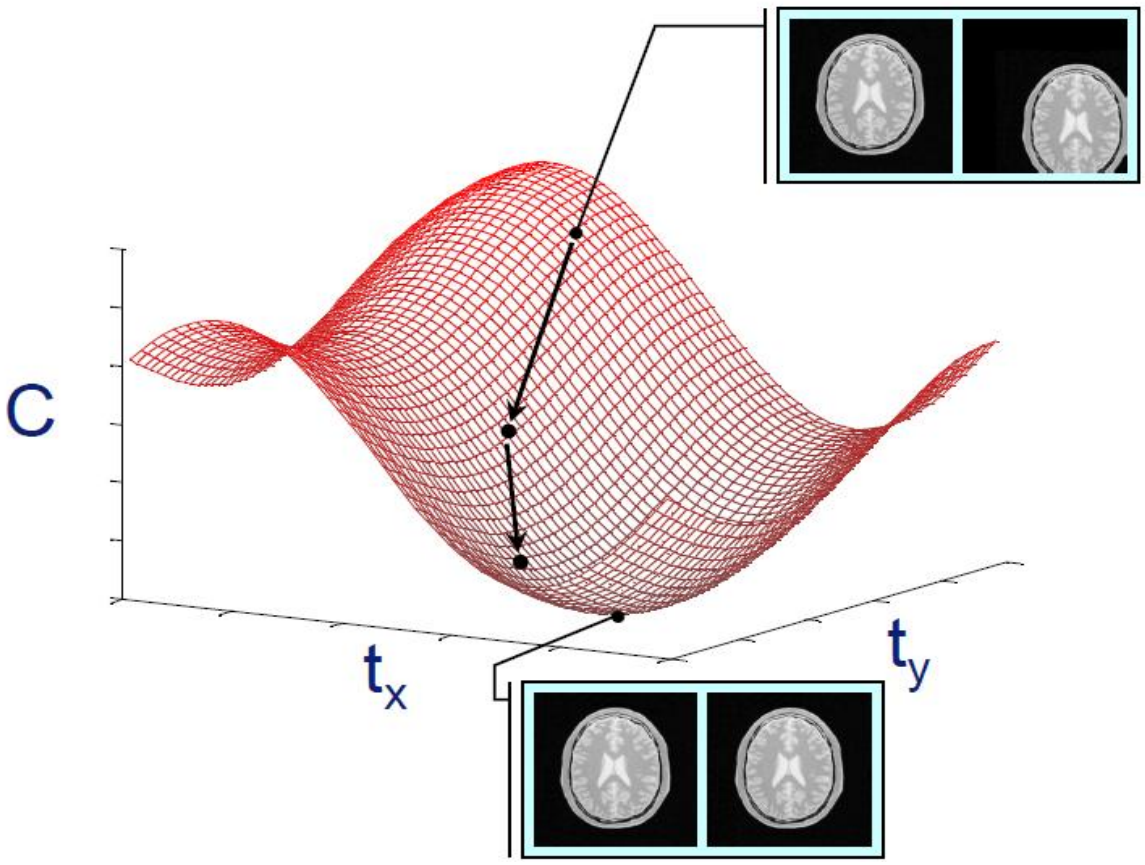


Figure 2-3. Optimization approach of the gradient descent

The main class in ITK for the gradient descent optimizer is called "*itkRegularGradientBaseOptimization.cxx*".

The main importance of this class relies on the stopping criteria defined for the optimizer, to alert the end of the iteration process and convergence of the registration.

The criteria are as follows:

- **Gradient Magnitude Tolerance**. This is the main criterion for the gradient descent to check the final parameter values obtained from the derivative of the cost function. These parameter values come from the class of the cost function where the similarity metric and the regularization terms are defined. Usually this tolerance value is equal to a very small value (nearly zero). In table 2-5 the tolerance value is shown for our optimizer.
- **Minimum and maximum step length**. The optimizer has three types of step lengths: minimum/maximum/current. Usually the optimization starts with setting the current step to the maximum step length at the very beginning of the optimization process. In our case the maximum step is set to 10, however the user can change it at the beginning of the program execution. Therefore the current step length varies during the iteration process and it is adaptive. The only fact that change the current iteration is a variable called "ScalarProduct". The whole procedure is executed as follows.

  ScalarProduct = GradientofCurrentIteration * GradientofPrevIteration

  If ScalarProduct < 0 then
  CurrentStepLength = CurrentStepLength * RelaxationFactor.

  The variable "RelaxationFactor" is equal to a small number in our case 0.5. Hence if for example the current step length is set to 10 which is the value of maximum step length, then after computing the gradient of the cost function, if it is smaller than zero, then the gradient moves one step toward the optimum solution and current step length is set to 5. Then the current step length is compared with the minimum step length. If the current length is still larger than the minimum length, then the iteration is continued and the observer asks to the cost function metrics to refine the misalignment once again. This process is iterated several times until the current step is smaller than the minimum step and the optimizer assumes that the optimum transformation parameters solution has been found and the registration process is terminated.
- The final criteria which is common to all the optimizer is the **number of iterations**. The number of iterations is set for each stage separately and in our case the user can enter this parameter for the coarse and fine stage.
  Usually the most common registration parameter that causes the registration process to get converged is the number of iterations. In table 2-5 the default number of iterations for our registration case is shown.

Optimizers are controlled by the observer to search for the solution in ITK. Observer functions are essentially defined in the *"itkRegularGradientBaseOptimization.cxx"* class

and are called at the very beginning of the main class. Thus the inputs for this class are the classes derived from the *"ImageToImageMetric"*. The initial parameters are set using the *"SetInitialPosition()"* function. Also the final optimized parameters can be obtained using the *"GetCurrentPosition()"*.

2-4- Implementation of Divergence and Curl embedded cost function

2-4-1-Approach

Divergence of the vector field is calculated in order to measure the expanding or compressing of the vector field. Similarly curl of the field is a metric to measure the changes in rotation orientation.
Usually divergence and curl of the deformation field are applied to each component of the voxels (x,y,z) but in this work we have set the Divergence and Curl operator onto each grid point of the BSpline as the parameters. This matter would save the computational time in terms of cost function and optimizer and also the amount of memory needed to store the cost function.

As can be seen from the cost function, there are two weights assigned to the divergence and curl of the deformation field:

$$F(x,y,z) = E_{sim} + (w_d E_{div} + w_c E_{curl})$$

Here E_{sim} is the similarity metric, which is the mutual information defined between the two images.

Those terms in brackets are the regularization terms defined as follows:

$$E_{div} = \sum_{\varphi} \|Div\, T\|^2 \quad (Eq.1)$$

$$E_{curl} = \sum_{\varphi} \|Curl\, T\|^2 \quad (Eq.2)$$

Therefore the final equation would be as:

$$F(x,y,z) = [H(X) - H(X|Y)] + \sum_{\varphi} \|Div\, T(x,y,z)\|^2 + \sum_{\varphi} \|Curl\, T(x,y,z)\|^2 \quad (Eq.3)$$

In the above equation 3, X, Y represent fixed and moving images respectively. T denotes the transformation function or practically speaking it is the deformation field which is characterized by the BSpline control points coefficients. Here φ is the counter defined for each component of the transformation. In fact φ moves on the vector channels of each vector of the deformation field.

Hence if we rewrite the divergence based on this, it would be as follows:

$$Div\,T(x,y,z) = {\partial T_x}/{\partial x} + {\partial T_y}/{\partial y} + {\partial T_z}/{\partial z} \quad \text{(Eq.4)}$$

$$Curl\,T(x,y,z) = \left({\partial T_z}/{\partial y} - {\partial T_y}/{\partial z}\right)i + \left({\partial T_x}/{\partial z} - {\partial T_z}/{\partial x}\right)j + \left({\partial T_y}/{\partial x} - {\partial T_x}/{\partial y}\right)k$$

(Eq.5)

The flexibility of our method is that the user can enter four types of weights to maintain the regularity of the final deformation field. These weights are assigned in order to regulate the functional quantity of the similarity term and the regularization term.

In order to implement divergence and curl of the vector field we applied the following approaches:
1- Since we are dealing with a vector field, this means we have a “Vector Image”. In a vector image each voxel of the image is specified as a vector, hence each voxel has a magnitude and a direction.
In this approach of implementation, the divergence and curl of each voxel in the image are computed separately and then combined together to have div-curl vector image of the deformation field. In this method three dimensions (x,y,z) of each voxel is obtained and base on the div-curl equations (eq.4 and eq.5) the parameters are calculated.
This approach is considered as a bottom-up technique, as it starts with the smaller particle of the deformation field to model the div-curl of the whole vector image.

2- Instead of computing div-curl of each voxel of the deformation field, we compute the div-curl of the whole deformation field for each direction separately (x,y,z). So it has x,y and z direction with the magnitude of summation of all the directions. This approach can be considered top-down approach as it starts with the computation of div-curl of the whole deformation field to reach each voxel. (Figure 2-4)
We chose to use the second top-down approach, since it is trivial to implement and it has a reduced computational time.

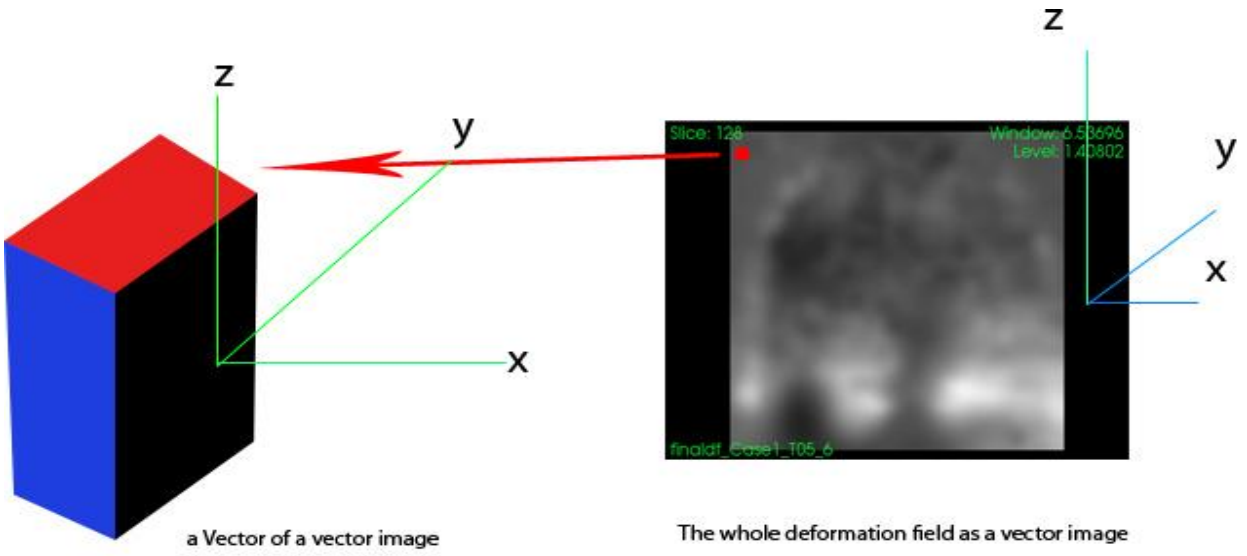


Figure 2-4. Left: choosing vectors to compute div-curl (Bottom-up), Right: choosing the whole deformation field to compute div-curl (Top-down)

2-4-2-Divergence

For constructing the divergence of the DF (Deformation Field), we tried to model the mathematic equation of the divergence (Eq.4). In order to apply the top-down approach, the whole procedure is structured as follows to obtain the divergence of the deformation field.

1- We give the deformation field as input to the process. This must be done at each iteration of the optimizer, since we need the divergence-curl of the deformation field to be summed to the similarity metric at each iteration. For this purpose we defined a function name “SetDeformationField()”. This function only calculates the point to point difference of the fixed and the warped image at each iteration, hence constructing the deformation field.

2- Split each component of the deformation field to x,y and z component separately. This is done using the itk::SplitComponentsImageFilter< DeformationFieldTypeGlobal, ScalarImageType, 3 >.
This class filter receives the whole DF as input and gives out 3 scalar image as output. So by now we have 3 scalar images for each component of the deformation field. T_x, T_y, T_z.

3- Then we have to compute the derivative of each of the components separately. $\partial T_x/\partial x, \partial T_y/\partial y, \partial T_z/\partial z$.

For this action, we used the class of itk::DerivativeImageFilter<ScalarImageType, ScalarImageType>, allowing to compute the directional derivative of an image based on a central difference algorithm (figure 2-5). The input must be a scalar image as well as the output.

We apply this filter on 3 components of DF as we obtained. For instance it computes the derivative of x component of the deformation field with respect to the parameters of x direction. The SetOrder() function specifies the order of the derivative which is 1 in our case and function SetDirection() specifies the direction of the derivative.
If we set SetDirection(0), it means we are intending to compute the x direction of DF with respect to parameters alongside the x direction.

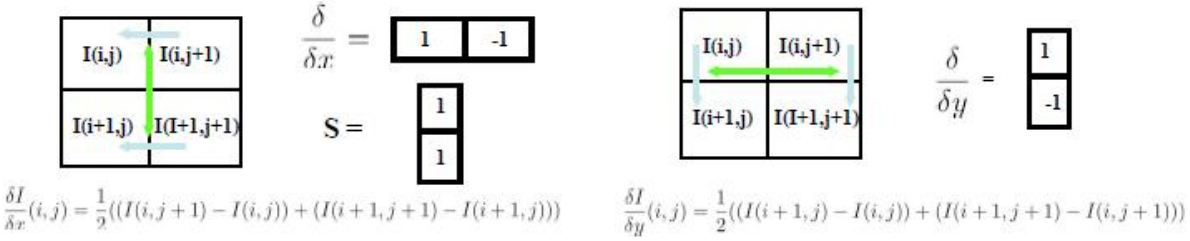


Figure 2-5. taking first order gradient derivative of an image in x direction (left) and y direction (right)

4- After obtaining the directional derivative of each component of DF at each iteration, we have to combine them together to construct the $^{\partial T_x}/_{\partial x} + {}^{\partial T_y}/_{\partial y} + {}^{\partial T_z}/_{\partial z}$ and model the divergence.

For this purpose, we profited of the *itk::ImageToVectorImageFilter<ScalarImageType>* class. This class receives n components as scalar images and combines them together by iterating over the image components to obtain the voxel magnitudes. Then it produces a scalar image product of the whole components. The final scalar image is the divergence of the DF, but since we need the values of the divergence to add them together to the curl and also to the MI, hence we have to compute Euclidean norm (L2 norm) of the divergence parameters.

5- Compute the L2 Norm of the Divergence as stated in Eq.4. This is done by iterating over the div image and computing the L2 Norm as Eq.1. In this step the following user defined weights are multiplied to the div-parameters, thus the div image is constructed and the parameters are passed to the optimizer.

$$W_{dc} = \textbf{weight assigned to div parameters in coarse stage}$$
$$W_{cc} = \textbf{weight assigned to curl parameters in fine stage}$$
$$W_{df} = \textbf{weight assigned to div parameters in coarse stage}$$
$$W_{cf} = \textbf{weight assigned to div parameters in coarse stage}$$

The following UML activity diagram depicts the procedure of designing divergence of a vector field using the components of our method. (Figure 2-6)

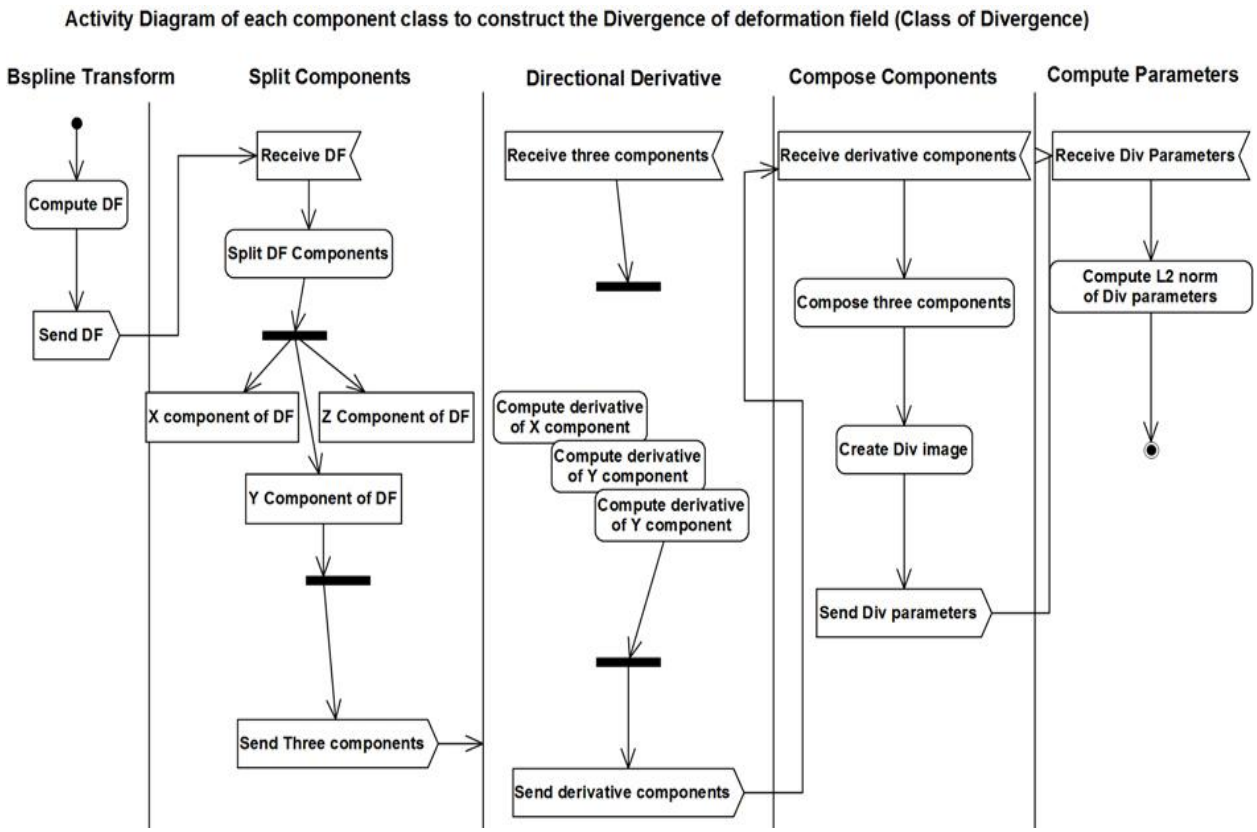


Figure 2-6. Activity diagram of divergence construction.

2-4-3-Curl

The procedure of producing the curl of the deformation field is almost the same as divergence, only it needs additional computation of the derivatives as its equation shows. (Eq.5)

If we illustrate the matrix of all the combination of DF derivatives with respect to all the directions as following matrix, the schema of construction of div-curl operators is clearly denoted. The diagonal elements of the matrix denote the divergence which is bounded by a red ellipse and the top-middle, middle and bottom-middle elements denote the curl operator that are shown by dashed blue ellipse in the following matrix..

$$\begin{bmatrix} \frac{\partial T_x}{\partial x} & \frac{\partial T_x}{\partial y} & \frac{\partial T_x}{\partial z} \\ \frac{\partial T_y}{\partial x} & \frac{\partial T_y}{\partial y} & \frac{\partial T_y}{\partial z} \\ \frac{\partial T_z}{\partial x} & \frac{\partial T_z}{\partial y} & \frac{\partial T_z}{\partial z} \end{bmatrix}$$

Problems and solutions

The main problem occurs when we intend to compute the derivative of div-curl parameters of the DF. Generally we encountered to the following two major problems.

Problem 1- When computing the values of div-curl parameters, we see that these values are not in the same scale as value of MI. The value of MI is typically much smaller than the final magnitude of div-curl. It means that the metrics of similarity and regularization functions are not maintained and regulated correctly.

Solution
We defined four weights to tackle this problem. Two weights for div and curl in the coarse grid stage with low resolution BSpline transformation and two weights for the fine stage at high resolution.
Defining four types of weights solution is the robust part of our method. This supplies the user with a high flexibility of regulation framework. The user can set two different weights to maintain the smoothness of expansion/compression of the coarse grid and set up two other weights to regulate the irregularities of the fine stage. The results and discussion about how to assign the weights and what is the regularization balance between the coarse grid weights and fine grid weights are described in chapter 4.

Problem 2- The second problem is related to computation of the derivative of div/curl parameters. The scale of the number of final parameters of the derivative of div/curl are not regulated correctly to match to the final parameters of derivative of MI. The number of parameters for these two terms must be matched since they are going to be combined and given to the optimizer to compare it with the tolerance defined in its criteria for the gradient descent optimization.

Solution
The solution is based on resampling div/curl parameters base on BSpline control points. This approach is to resample the div/curl parameters of derivative basen on mapping points of the BSpline grid nodes. This is exact same way that ITK deploys to resample the parameters of MI on the fixed image. The idea is that we only choose the parameters that are located on the grid nodes. For executing this idea we first need to find out where the BSpline indices are located on the image, then convert them to physical points and choose only those parameters as the array of derivative of div/curl parameters. Eventually we add the derivative parameters of both terms of the cost function and then pass them to gradient descent optimizer. The optimizer decides to finish the registration process or to go for another iteration. Figure 2-7 illustrate the idea of this strategy.

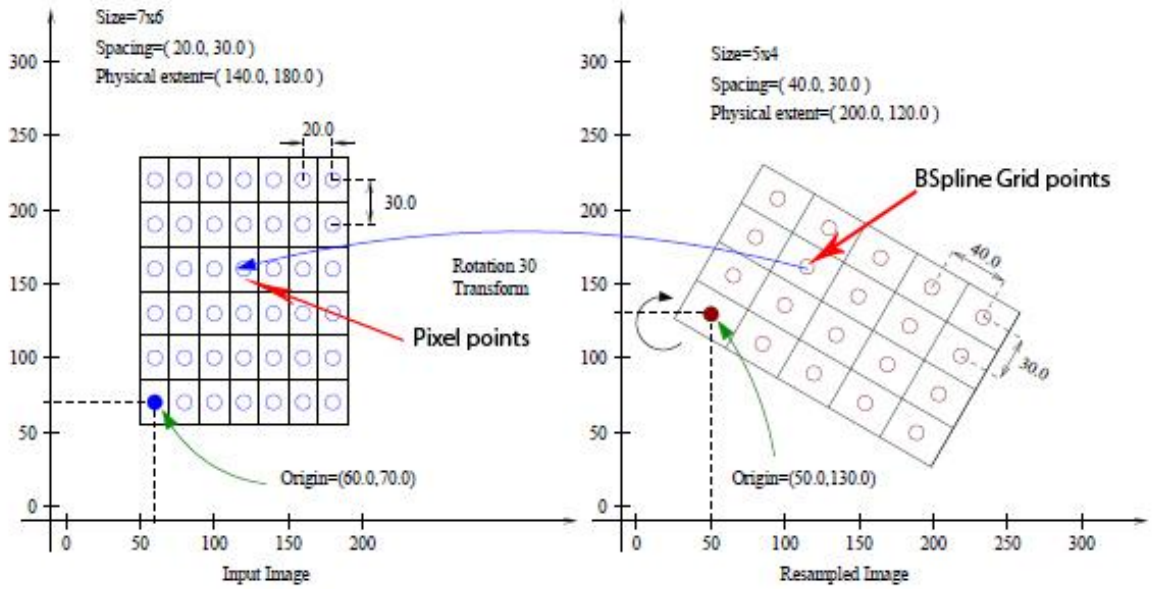


Figure 2-7. Resampling pixel values of the divergence image to rescale the div parameters

2-5- Implementation of Multi-Metric Multi-Resolution Registration

2-5-1- Introduction

In multi-resolution registration the main process is as follows:
1- Global pose differences are compensated by a linear registration algorithm (rigid-affine).

2-Non-rigid registration using a coordinate transformation that is parameterized by the function used for transformation. A small number of control grid points (5-20) are defined to model global non-rigid deformation.

3- A second non-rigid registration in order to model local deformation by defining a large number of control grid points (20-40).

However multi-resolution registration is defined for the parametric registration where the grid points of the transformation functions such as cubic –Bspline are characterizing the final deformation. Therefore the number of grid control point or the grid control spacing are critical parameter for performing the parametric non-rigid multi resolution registration.

We have used this technique to converge our registration method to a smoother and regulated field and the best aligned warped image. The procedure is as follows.
1- fixed and moving input images are roughly aligned using a transformation initialization.

2- They are registered using a rigid transformation which the result is used to initialize the next affine registration.

3- They are registered using affine registration. The output result of affine transformation is used as the main transformation of BSpline deformable registration. The transformation result of affine registration is called “bulk transformation” here. We have to mention that usually in the biological images i.e. Breast images, the global motion is modeled by an affine transformation.

4- Coarse grid deformable registration is performed by defining a small number of BSpline grid control points. The default control point we defined is [12 12 12] for this stage as can be seen from table1. In table 1 the number of grid points is written [15 15 15]. This is due to considering of the border of BSpline transformation in every direction x,y and z.
In this stage since the number of grid points is small, hence we have a larger grid spacing in order to warp and recover the global deformation. Usually in this stage 5, 10, 20 number of grid points are selected and the grid spacing depends on voxel spacing or dimension of the voxel (spatial resolution) defined in the image. The formula to obtain grid spacing using number of grid points is described in the third part of chapter 3.

In ITK the number of nodes in low resolution coarse grid is represented by the “NumberofGridNodesinOneDimensionCoarse” variable.

Thus the coarse stage registers the macroscopic dissimilarities in the image. For example M.F. Spadea et al. [10] empirically evaluated Plastimatch [10], an open source software command prompt base toolkit exclusively designed for image registration and segmentation, in terms of registration parameter optimization. In this work they have set the coarse grid spacing to [100 100 100] mm which is considered a quite big spacing on the image.

After development of coarse grid registration, we move to a fine grid stage registration in the last phase of our registration procedure, to cover the local deformation on the image and register the small scale variations. In this step we increase the number of grid points to 20,30 and 40 and in contrast decrease the grid spacing to focus on more localized transformation. In [10] for this step they have set [60 60 60]mm as the grid spacing between the control points of the BSpline.

In ITK number of nodes in high resolution fine stage is shown by “NumberofGridNodesinOneDimensionFine” variable for each dimension of the image.

It also has to be mentioned that naturally the number of parameters chosen over the fixed image defined for the deployment of transformation in the coarse stage is usually much smaller than in the fine grid stage. For instance in our implementation case, the number of transformation parameters chosen for the coarse grid with the same specification of table 1 is about 6591 parameters and for the fine grid is about 36501 parameters. These are the BSpline control point coefficients that are responsible to interpolate the non-rigid region on the image. The whole scheme of the multi-resolution registration is illustrated in the following figure 2-9.

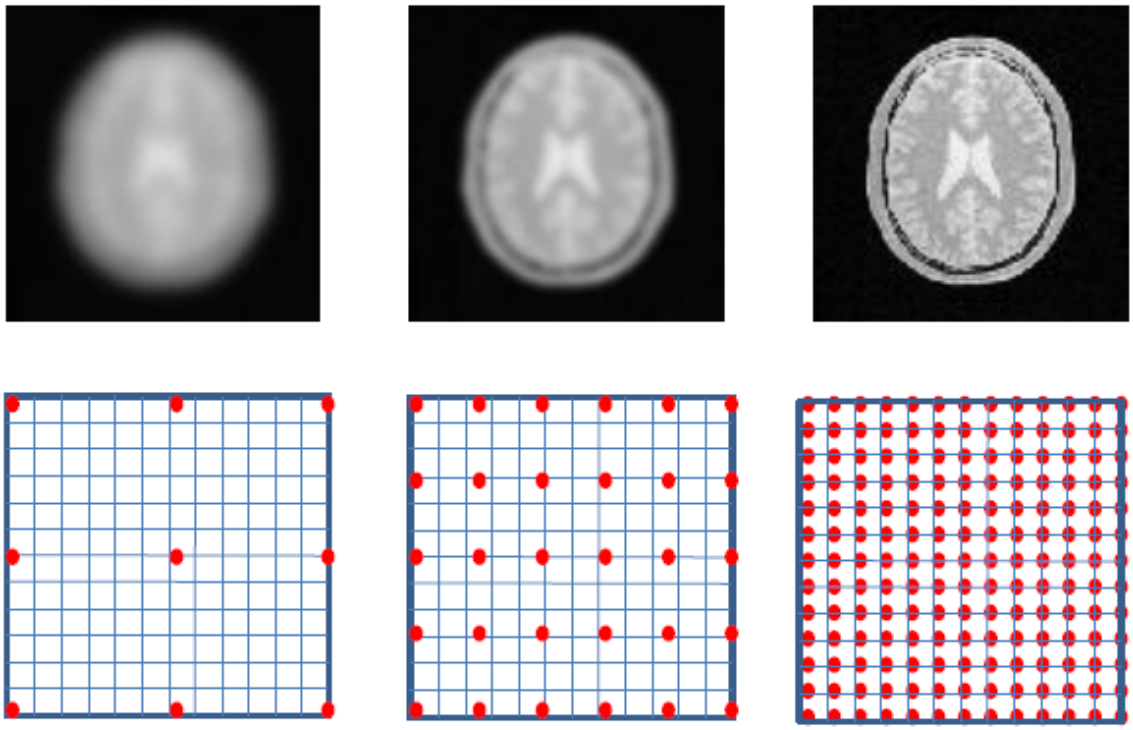


Figure 2-9. Multi-resolution registration. From coarse to Fine

2-5-2-Integration of the whole image registration procedure

In order to implement multi-resolution registration the following approache has been utilized. The flowchart of different classes used at each step of development is illustrated in figure 2-10.

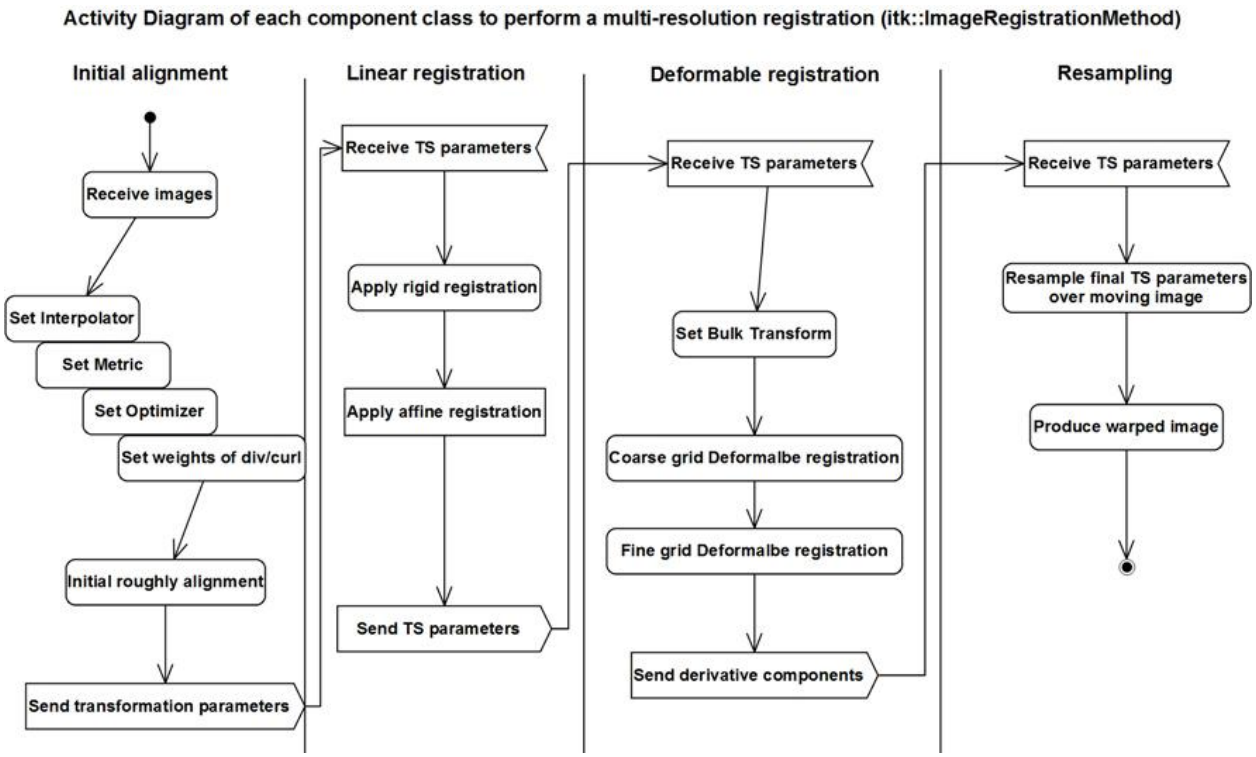


Figure 2-10. Activity diagram of multi-resolution registration procedure.

1-Using ITK filters

Use *"MultiResolutionPyramidImage"* filter to build fixed and moving pyramids along with *"MultiResolutionImageRegistrionMethod"* class to model the multi-resolutional registration framework. Having used latter filter classes, the problem here is that ITK does not supply the users with adequate functions and properties. Therefore most of the structure of the procedure must be done by the developers. In this regard, maintaining the compatibility and the consistency of the new codes with the ITK classes could become a time consuming process.

2- Stage-wise registration procedure

In this method we tried to develop and design each stage of the aforementioned (part 2-5-1) linear and non-linear registration in a consistent way to construct the whole process. The main inputs and outputs of our program will be described in the beginning of chapter 3 in more detail. Here we will only mention to the critical components of the whole registration process.

The general class that performs and controls the whole registration process is the class "*itk::ImageRegistrationMethod<FixedImageType, MovingImageType* >". By this class one can specify the registration type.

After defining each object and instance of the MI metric, BSpline transformation and the optimizer, we introduce those objects by the following commands to the main class of the registration.

registration->SetMetric(metric);

registration->SetOptimizer(optimizer);
registration->SetInterpolator(interpolator);

Then the alignment of two images is started using the following initializers:

initializer->SetTransform(rigidTransform);
initializer->SetFixedImage(fixedImageReader->GetOutput());
initializer->SetMovingImage(movingImageReader->GetOutput());

After performing the rigid and affine linear registration, using the following function of bulk Transform, we introduce the previous linear registration as the input point of the BSpline deformable registration.

bsplineTransformCoarse->SetBulkTransform(affineTransform);

The way that we call the regularization functions of divergence and curl of the DF that is computed at each iteration of the optimizer, is to design a function called *"SetWeights()"* which accepts 5 sorts of variables as input. The first four variables are the weights that are assigned by the user at runtime and the fifth variable is the number of grid nodes either in coarse or fine grid.
Hence the *"Setweight()"* function passes the variables entered by the user to the class of "*itkMattesMutualInformationImageToImageMetric*" and the weights are multiplied to the div/curl parameters and the final values of pure div/curl and their derivatives are structured.

The very eventual step that models the warped image from the moving image is the "resampling filter". This filter applies the finally obtained optimal transform parameters to the moving image by using resampling filters. It uses the resulting transform to map the moving image into the fixed image space. The output of the filter is passed to a writer that will store the image in a file. An *"itk::CastImageFilter"* is used to convert the pixel type of the resampled image to the final type used by the writer. By applying this step the warped image is produced.

2-6- Evaluation methodologies

An introduction to the major methods and procedures of registration performance evaluation are described in the ending part of chapter 1. Here in this part we will bring our methods of evaluation henceforth. We tried to discuss each method in a quite elaborated way in term of both theoretical and practical implementation programming code.

2-6-1- Difference of fixed and warped image

These metrics are calculated using a manual contour thresholding, thus background and the patient area have been discriminated by simply defining a threshold of value -1200 (in Hounsfiled Units). So before calculation of every metrics each voxel value has been compared with this threshold value and if it is different than the value, then the specific metric is calculated. By doing this static global binarization thresholding we can

define an automatic contour and segment out the foreground from background. For this propose local adaptive thresholding method such as Otsu's method could also be used.

Among the following evaluation methods, depending upon the type and approach of registration, some of them could be more robust, useful and stable for evaluation. However there might be no registration process which all of the following metrics could be utilized for as the benchmark.

Likewise, since we have been doing a sensitivity testing for our weights of divergence and curl to figure out the best domain and confidence interval, thus the more metric we could have used the more deductive result we can obtain. Also each metric could be an indicator of some criteria or point for the testing result.

1-RMS Intensity Difference

It is simply the root mean squares of the difference between fixed intensity and warped intensity [1].

RMSint $= \sqrt{\frac{1}{N}\sum(If - Iw)^2}$

N is total number of pixels excluded from the threshold value and "If" is intensity at reference image and "Iw" is the intensity at warped image.

2-Joint Entropy and Mutual Information

Using the {JoinImageFilter} we can obtain joint histogram of each image at first and then by calculating the joint entropy among two images, the mutual information is obtained. By simply adding entropies of each image and subtracting it from the joint entropy we can obtain mutual information. [1]

Mutual Information = Entropy of image1 + Entropy of image2 – Joint Entropy

For evaluation of our second data type of clinical patients we use Normalized Mutual Information (NMI), where the value of Mutual Information gets divided by the mean entropy of the input images as follows:

Normalized Mutual Information =

2.0 * (Mutual Information) / (Entropy of image1 + Entropy of image2)

3-Edge Overlap

For computing edge overlap of the images, the Canny edge detector was used [1], which extract the high frequency part of the image by simply computing the strong gradient of the image. Then the sum of the absolute differences divided by the number of voxels in the overlap region is used to compare the binary images. This function lies between 0 and 1 with 0 denoting optimal overlap. As can be seen from figure 2-11 we investigate on edge overlap between fixed and final warped mage.

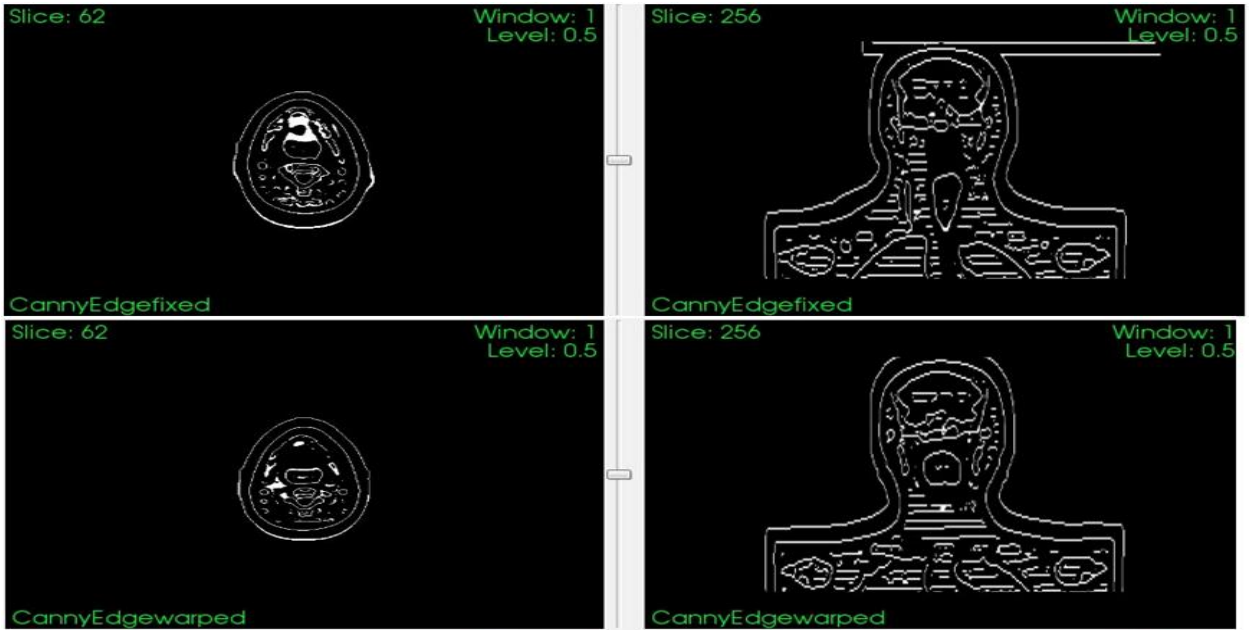


Figure 2-11. Upper images are axial and coronal view of edge detection done by canny filter on fixed image and the bottom images are the same filter applied on warped image.

4-Center of Mass

Simply calculated by ITK filter applied on fixed image and warped image. Then the mean of the two moments has been calculated for each direction [2]. The main idea is to compare moment of the fixed and warped image which the least difference represents the better result.

This is the idea:

```
Mean_center[0]=(CenterOfMass_fixed[0] + CenterOfMass_warp[0])/2;
Mean_center[1]=( CenterOfMass_fixed [1] + CenterOfMass_warp [1])/2;
Mean_center[2]=( CenterOfMass_fixed [2] + CenterOfMass_warp [2])/2;
```

5-TP, FN, TN, FP

For finding the intersection or overlap region between two images, in terms of intensity of pixel values, we could not consider the threshold value of -1200 here, because most of the overlapped region would be the background region, which is not needed for evaluation. However if we do not consider the threshold value, the percentage of the overlapped region between two images is reduced even with the native Only-MI registration.

When we look at the intensity values of output warped image performed by the native only MI registration done by ITK or SSD and if we directly compare the intensity pixel value with the fixed image (in our case rando phantoms), most of the pixel values in the patient area do not match to each other.

There could be two reasons for this fact. First and obvious reason is that the interpolation done by the transformation function to model the non-rigid regions construct some in-between values for some voxels that are not located on the grid nodes. The second reason could be of resampling filter that is performed at the end of

registration to fit the transformation parameters on the moving image and model the warped image.

Therefore, comparing the pixel value of both images directly could not be a quite accurate evaluation technique; instead we have to consider the max, min, median or probability of occurrence of the pixels by using histogram for most suitable evaluation.
For computing intersection of two images, first the pixel value corresponding to the index should exist inside the fixed image then that pixel value must be equal to the pixel value of the corresponding index on the moving image.

For computing these metrics we have implemented the following equations: [2], [4]
TP (true positive) = overlapped region = $If \cap Iw$
TN (true negative) = $If^c \cap Iw^c$
FP (false positive) = $If \cap Iw^c$
FN (false negative) = $If^c \cap Iw$

6-Histogram of differences
This metric is a very common evaluator which calculates the histogram of intensity differences between fixed and final warped image. As is described in chapter 3, we have used two approaches to demonstrate this metric of evaluation. Since this metric uses intensity values for the evaluation, thus most of the intensities of difference image should be zero or very close to zero. In this way we can say we have a better final result for the registration. Hence the histogram we obtain from the difference of fixed and warped image after the registration must be aggregated among zero to have a better result. The more diffusive values we have in the histogram, the worse the final warped image will be.

2-6-2- Difference of synthetic and final deformation field

1-RMS displacement field
Root mean squares of the difference between the synthetic displacement field and the obtained deformation field [1]:

RMSdisp = $\sqrt{\frac{1}{N}\sum(\boldsymbol{\varphi syn} - \boldsymbol{\varphi df})^2}$

Φsyn = sysnthetically obtained deformation field
Φdf = the finally obtained deformation field related to each case

2-Robust Max displacement field difference
It simply calculates the maximum value of the difference between the displacement fields by first iterating over the fields then computing the difference and comparing each difference value with the previous value to finally obtain the maximum difference value. It is used as robust maximum, because the maximum intensity difference is defined as the intensity difference that is larger than 95% of all other values. [1]

2-6-3- Jacobian determinant of the transformation

The Jacobian matrix is second order matrix of the deformation.

$$J_{\varphi}(x) = \begin{matrix} \frac{\partial \varphi_1(x)}{\partial x_1} & \frac{\partial \varphi_1(x)}{\partial x_2} & \frac{\partial \varphi_1(x)}{\partial x_3} \\ \frac{\partial \varphi_2(x)}{\partial x_1} & \frac{\partial \varphi_2(x)}{\partial x_2} & \frac{\partial \varphi_2(x)}{\partial x_3} \\ \frac{\partial \varphi_3(x)}{\partial x_1} & \frac{\partial \varphi_3(x)}{\partial x_2} & \frac{\partial \varphi_3(x)}{\partial x_3} \end{matrix}$$

The Jacobian matrix encodes the local stretching, shearing and rotating of the deformation field. Useful measures that can be derived from the matrices are the determinants, which indicate relative volumes before and after spatially transforming. A region of negative determinants would indicate that the one-to-one mapping has been lost.

If φ_c is the deformation that results from the composition of two deformations φ_B and φ_A (i.e. $\varphi_c = \varphi_B . \varphi_A$), then the resulting Jacobian field can be obtained by the matrix multiplication $J\varphi_C = (J\varphi_B . \varphi_A)\, J\varphi_A$. This leads to a similar scaling and squaring approach that can be used for computing the Jacobian matrices of deformations.[9]

In practical approach we have written a program to evaluate the jacobian determinant of each voxel of our final deformation fields using two following ITK filters.

itk::DisplacementFieldJacobianDeterminantFilter<DeformationFieldType, PixelType>

Computes a scalar image from a vector image (e.g., deformation field) input, where each output scalar at each pixel is the Jacobian determinant of the vector field at that location. This calculation is correct in the case where the vector image is a "displacement" from the current location. The computation for the jacobian determinant is: $\det\left[\frac{dT}{dx}\right] = \det\left[I + \frac{du}{dx}\right]$. [7]

itk::MinimumMaximumImageCalculator<ImageType>

This calculator computes the minimum and the maximum intensity values of an image. It is templated over input image type. To compute Maximum or Minimum value ComputeMaximum() (ComputeMinimum()) functions can be called, otherwise Compute() will compute both. [7]

This application is based on [2] and it calculates the jacobian determinant of the first partial derivative of the transformation. It computes the minimum and maximum jacobian value in the whole vector field so that we can figure out whether we have any negative jacobian value for any voxel or not. It also computes the percentage of voxels which its jacobian is less than 1, more than 1 or equal to 1. It also gives us the final jacobian image applied on the vector field. (Figure 2-12)

In a Deformation Field, Jacobian values near 1 shows no local volume change in term of quantification and in the jacobian image it is in grey, values lower than 1 show volume decrease and are dark grey to black, and values above 1 (volume increase) are light grey to white. [11]

If we have the bigger percentage of jacobian values equal to 1 in any vector field, it might be more consistent, non-singular and invertible. Also in term of minimum jacobian value in the jacobian image, the positive and bigger number we have, the smoother deformation we will obtain. That is because we go far away from the zero and negative values. In contrast in term of maximum jacobian value the smaller number we have the smoother vector field we will expect.

For example for the 3D CT images of lung obtained from [6] (www.dir-lab.com), suppose we have performed a non-rigid registration using our method.
The fixed image is the very beginning of inhale phase of the patient and moving image is the very beginning of exhale phase of the patient. We have chosen these two cases here with four weights of [0.5, 0.5, 0.5, 0.5] to evaluate its jacobian determinant of the transformation. $\boldsymbol{W_{dc}}$ represents weight assigned to divergence parameters in coarse stage, $\boldsymbol{W_{cc}}$ is for curl parameters in coarse stage, $\boldsymbol{W_{df}}$ is for divergence in fine stage and finally $\boldsymbol{W_{cf}}$ is for curl in fine stage.
For the better perception of the content, in the table 2-6, the result for one case can be seen as an example of minimum and maximum jacobian and percentage of jacobians bigger and smaller than 1 as described previously.

Cases	**Case1_T05_6**
Weights	$\boldsymbol{W_{dc}}$**= 0.5** $\boldsymbol{W_{cc}}$**= 0.5** $\boldsymbol{W_{df}}$**= 0.5** $\boldsymbol{W_{cf}}$**= 0.5**
Min Jacobian	0.336578
Max Jacobian	1.72964
Percentage Jac >1 Jac<1	61% 38%

Table 2-6. representing jacobian determinant of the deformation field of the case1_T05_6

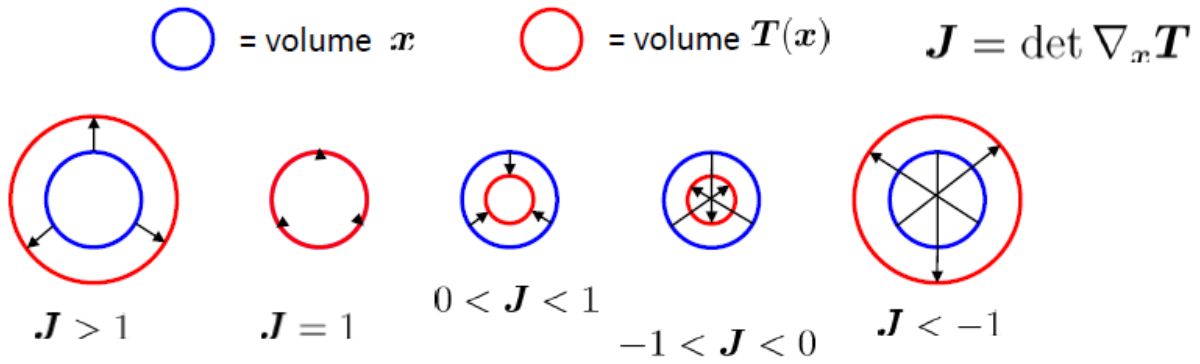


Figure 2-12. Volume increase, decrease and no volume change after evaluating Jacobian determinant of the final deformation. [11]

The detail discussion about these values has been done in chapter 4 for both real cases and the synthetic cases.
This application is very useful especially in the real-patient cases, since using this application, user can brightly obtain the ability of the invertibility of its final deformation field and it shows to the user the potential of having negative jacobian in the transformation which help for the final evaluation of the deformation field.
The results outcome from the application can be seen in chapter 3 along with the program scheme user interface, its input and output parameters.

Chapter3
Results and application evaluation

After describing our analysis methods, design and implementation strategies, in this chapter we have demonstrated our methodology. We have developed a main application of deformable image registration which utilizes weights for different stages of non-rigid registration. Assigning these weights helps researchers to easily balance and regulate the type of smoothness they are looking for in terms of vector spline regularization.

In Part 3-1 we describe the user interface of our application and inputs needed to execute it. The application is run on a command prompt shell and all the process can be seen by the user, either on the shell or in a stored text file as a log. The application accepts several types of 2D/3D meta-images.

We also developed three small applications for the registration evaluation. We utilized some parts of already-written open source codes of ITK to implement these evaluation frameworks. In part of 3-2 we have described all of the three frameworks in detail. The inputs and a screenshots of the application are explained for the researchers to better figure out the deployments.

As described in chapter 2, in this thesis we have tested two types of dataset. The first one is an image dataset relying on an anthropo-morphic head and neck phantom. Part 3-3 shows our results concerning this type of dataset. We have obtained better results if compared woth other registration frameworks which have the same pre-registration parameters assigned.

Another dataset we employed to exploit our method consists of clinical real patient cases. Because of the anatomical landmark coordinates that are annotated on this type of dataset, hence it can be used for a very accurate registration assessment framework. We also have utilized this dataset to evaluate the inverse consistency of our method. Evaluating our framework by using this dataset showed us some crucial points which we did not get when using the phantom synthetic dataset. Part 3-4 explain the assessment framework of this clinic dataset.

Eventually in the final part of this chapter we have tried to obtain the optimal registration parameters. These optimal parameters are obtained empirically using our experiments based on a trial and error method and observational sensitivity testing. The result can help researchers to have an idea about what are the optimal registration parameters using BSpline transformation parameters, MI as similarity metric and divergence/curl of the deformation field as the regularization function

3-1- Application execution

This part explains our main application used to perform the multi-resolution registration. First the components of the application are described, and then the method of execution and interface is demonstrated by screenshots.

3-1-1- Application Components

In chapter 2, we have described the whole implicit components of the multi-resolution registration algorithm, starting with an approximate initializer and ending with a fine grid base deformable registration. Here we discriminate the explicit application components as detailed below.

1- Input parameters related to the class of main registration

- Fixed image
- Moving image
- Output warped image (to be produced by the application)
- Difference of Fixed and Moving images before registration (to be produced by the application)
- Difference of Fixed and Warped images after registration (to be produced by the application)
  Usually this difference image is used for the final evaluation.

2- Input parameters related to transformation parameters

- Final Deformation field (to be produced by the application)
- Final Transformation parameters (to be produced by the application)
  These parameters are the solution of the registration procedure if we look at the registration as an optimization problem. The moving image is resampled by these parameters to obtain the final warped image.
- Number of coarse grid nodes
- Number of fine grid nodes

3- Input parameters related to the optimizer

- Number of iterations
  In our application user can only enter the number of iterations for two coarse grid and fine grid deformable stages, whereas thenumber of iterations for rigid and affine stages are set by us in the program experimentally as 200.
- Maximum Step length
  The default value is 10 but the user can regulate the number to check the result based on different assumptions.

4- Input parameters related to the regularization function

- Four types of weights assigned to the vector spline div/curl operators
  These weights are the most robust part of our application. The user can enter four weights, each for separate stages of the deformable registration procedure. The optimum weights are discussed in chapter 4 for each separate case exclusively.

In the following image the user interface of the main application can be seen. (Figure 3-1)

```
sadegh@Workstation-CART:~> nohup ./sadegh_registration ../Lung_Images/Case1_T00_
s.mha ../Lung_Images/Case1_T50_s.mha out_warped_Case1_T05_15.mha diffaft_Case1_T
05_15.mha diffbef.mha finaldf_Case1_T05_15.mha trans_Case1_T05_15.txt 12 30 10 5
00 0.5 0.5 0.01 0.01 >& logFile.txt &
```

Figure 3-1. the interface of the main application of our registration framework

The input variables are assigned in the following order respectively.

The executable application is named as “sadegh_registration” which executes the main program.

- Fixed image: First parameter which assigns the fixed image is “Case1_T00_s.mha”.
- Moving image: “Case1_T50_s.mha”.
- Warped image: “out_warped_case1_T05_15.mha”
- Difference image after registration: “diffaft_case1_T05_15.mha”.
- Difference image before registration: “diffbef.mha”
- Final Deformation field: “final_df_case1_T05_15.mha”
- Final Transformation parameters: “trans_case1_T05_15.txt”
- Number of grid points in coarse stage: 12
- Number of grid points in fine stage: 30
- Max step length: 10
- Max number of deformable iterations: 500

W_{dc} = weight assigned to div parameters in coarse stage = 0.5.
W_{cc} = weight assigned to curl parameters in fine stage = 0.5.
W_{df} = weight assigned to div parameters in coarse stage = 0.01.
W_{cf} = weight assigned to div parameters in coarse stage = 0.01.

Consequently we have applied the above application to the two types of dataset to evaluate our method and we have obtained the results which are presented in parts 3-3 and 3-4.

3-2-Development of the evaluation application

We developed four different applications for evaluation by profiting the metrics described in chapter 2, part 2-6. We utilize the metrics after performing the registration to assess our final result and quantify the alignments errors. The results of our registration method are reported in parts 3-3 and 3-4.

The four command base applications are as follows:

3-2-1- “Compare Displacement Fields”

This application is based on [1] and [3] and it has two types of calculations. First it calculates the difference between synthetic deformation filed and the finally obtained deformation field.

Second, it calculates the difference between reference image and warped image and also reference image and target image. It is mostly based on statistical calculation of min, max, mean, median, MI, joint entropy and Edge overlap.

We have to mention that this application is applicable only to the case where the synthetic deformation field is available, i.e. with simulated images. If we are performing a registration procedure where the synthetic deformation field is not available, such as for real patient cases, then we can use the next application along with the jacobian determinant application.

The user interface and how it is executed by the final user is as follows.

Figure 3-2. the scheme of the command prompt user input of “ComparedisplacementFields”

The input parameters are:

- Fixed image. In the above example it is “rando_warped.mha”
- Output warped image to be produced. The example is “out_warped.mha”
- Moving image. In the example it is “rando_p.mha”
- Synthetic deformation field. In figure x it is “synthetic_df.mha”.
- Finally obtained deformation field. ”final_df.mha” in the example.
- Contour thresholding to mask out the unnecessary black area of background of the image to decrease the computation time and increase the accuracy and sensitivity of the registration.
  This fact is very important especially in the evaluation of the registration. Even thought we perform the registration without any contour which discriminate the patient area of desired with the unnecessary background, but in the phase of evaluation we have to consider the discriminant approach. This is due to the fact that if we consider the whole image in order to evaluate the final result, the final accuracy will be affected by a large background ares which is perfectly overlapping by definition.
- Two text files to store the final results, one for the image statistic result and the other for deformation field results.

The outputs are exactly the metrics which have been described in the part of deformation fields in chapter 2. (2-6-1) and (2-6-2)

The code is written in VC++ 2010 using the ITK library. The major ITK filters have been used in the program is as follows.

- itk::JoinImageFilter<Int16ImageType,Int16ImageType> and itk::Statistics::ImageToHistogramGenerator<VectorImageType >
  These filters are used to calculate the joint entropy of each image and consequently the joint entropy of both images. In this procedure the calculation of

histogram bins and samples are needed to obtain the joint probability distribution function. Eventually the mutual information is obtained as well.

- itk::CannyEdgeDetectionImageFilter<FloatImageType,FloatImageType>
  As described in the previous chapter, this is the main filter of Canny edge detection to compute the edge overlap between the images.
- Most of the other part of the program is merely statistical calculation of the median, mean, max/min intensity evaluation using iteration over the images by itk::ImageRegionConstIterator<Int16ImageType> and itk::ConstNeighborhoodIterator<DeformationFieldType>.

3-2-2- "Compare Warped Images"

This application is the part of previous "Compare displacement Fields" application, but it only calculates the difference of the fixed and warped image and also fixed and moving image. The reason of designing two different kind of applications is due to the type of dataset we are dealing with and the registration procedures that were performed. The previous application cannot be used in case of real patients, where we do not have a synthetic deformation field.

The whole structure of this application is almost the same as "Compare displacement Fields", except it does not compute any metric regarding the deformation fields. The output metrics are as (2-6-2).

3-2-3- "Image Moment Evaluation"

This application which is based on reference [4], [2] and [3] (ordered with priority) and it calculates image statistics between reference image and the warped image, mostly concentrating on images moments and center of mass of the fixed and warped image, overlap region and TP, FP, TN, FN.

Also for this application we have considered -1200 intensity value of the thresholding for only the synthetic cases of the "rando phantom". Hence if one is going to use this application for the real patients, before using it, the images must be masked or cropped and the desired patient area must be calculated and discriminated from the background or extreme regions.

In the following the interface of the user input is shown.

```
D:\Thesis\ITK\ITK_bin\Examples\Registration\DeformableRegistration30.dir\Deforma
bleReg\Only_MI\256>imagemomenteval dataset/rando_warped.mha out_warped.mha -1200
img_result.txt
```

Figure 3-3. user interface using command prompt to execute "imagemomenteval" application

The inputs are:

- Fixed image. Here "rando_warped.mha".
- Finally supposed warped image. "out_warped.mha".
- The threshold value, supposed to discriminate the patient area and the undesired background. -1200 synthetically and artificially set in background.

- The final result of metric in text.

The most important ITK filter used here is itk::ImageMomentsCalculator<FixedImageType>, which computes moments of a n-dimensional image.

The output metrics of the application are as (2-5-2)

3-2-4- "Determinant Jacobian"

This application which is based on [2] calculates the jacobian, which is determinant of the first partial derivative of the transformation. It computes the minimum and maximum jacobian value in the whole vector field, so that we can figure out whether we have any negative jacobian value for any voxel or not. It also computes the percentage of voxel whose jacobian is less than 1, more than 1 or equal to 1.

This application is very useful especially in the real-patient cases, since using this application the user can judge the invertibility of its final deformation field. An example of the jacobian image is shown in figure 3-5.

The scheme of the user interface is as follows:

Figure 3-4. jacobian determinant user interface

The inputs are:

- Final deformation field vector image. Here "final_df.mha".
- Jacobian image (to be produced)
- The result of the jacobian as a text file.

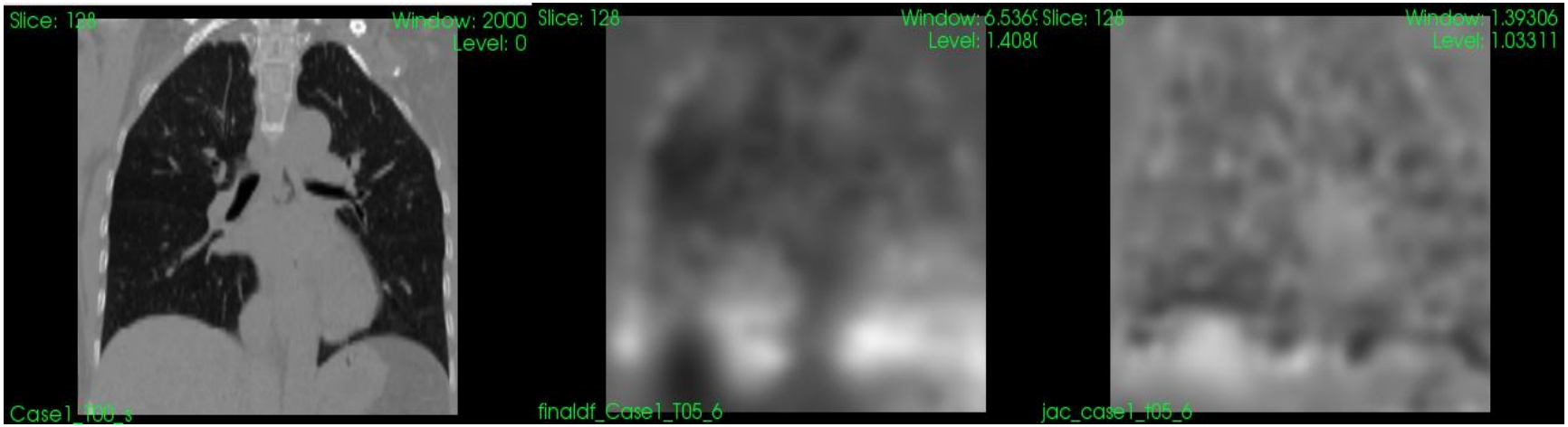


Figure 3-5. Left: the original final warped image. Middle: the final deformation field obtained from the registration procedure. It is a vector image. Right: the jacobian image of the final deformation field .

3-3- Synthetic Deformation cases

After implementation of our weighted cost function for non-rigid registration, we have to run the application and test it out using the real images. Description about this dataset is already presented in the second part of chapter 2. We performed several cases (more than 60 experiments) and tried several weights to obtain the best alignment and recovery (warping) of the deformed image. In chapter 4 the discussion about how to choose weights and what are the confidence intervals are discussed along with comparison of our method to the MI-only and SSD cost function method which do not utilize any regularization term in their cost function.

3-3-1- Image similarities evaluation

In the following tables 3-1 and 3-2 we have shown only the cases which have a crucial impact on the registration process, hence we present 14 cases here out of 61 cases we tested using these synthetic images. The output results are obtained using the four evaluators introduced in the previous part and the assessed metrics are the exact same metrics introduced and discussed in chapter 2. For all the following cases the subsampling resolution is set to [2 2 1] and number of grid points for coarse stage and fine stages are [15 15 15], [33 33 33] respectively. It must be again mentioned that the weights order are as following.

$$W_{dc} = weight\ assigned\ to\ div\ parameters\ in\ coarse\ stage$$
$$W_{cc} = weight\ assigned\ to\ curl\ parameters\ in\ fine\ stage$$
$$W_{df} = weight\ assigned\ to\ div\ parameters\ in\ coarse\ stage$$
$$W_{cf} = weight\ assigned\ to\ div\ parameters\ in\ coarse\ stage$$

Cases	Case 22	Case 24	Case 28	Case 36	Case 37	Case 38	Case 39	** Case 44	*** Case 45
Weights	0.1 0.1 0.1 0.1	0.5 0.5 0.5 0.5	4 4 4 4	1.5 1.5 1.5 1.5	2 1.5 1 0.5	0 1 0 1	1 0 1 0	0 0 1 1	0.1 0.1 0 0
Iterations(Coarse, Fine)	217 32	220 32	500 63	224 31	225 32	222 31	217 32	217 32	217 32
Time sec (Coarse, Fine)	7025 938	7333 956	16611 1883	7221 895	7235 918	7201 902	7079 927	7051 958	7087 958
MI (percentage)	0.909034	0.90509	0.865025	0.908364	0.904282	0.905006	0.908678	** 0.910419	*** 0.911669
Edge overlap	0.0256265	0.025642	0.0350401	0.0255223	0.0255377	0.0255506	0.0256353	0.0255853	*** 0.0246439
RMS intensity difference (intensity)	37.19	37.1838	41.6092	37.2091	37.2041	37.1893	37.1886	37.1883	*** 36.1901
TP TN FN FP (Number of pixel)	3439417 4064 544823 9392	3439606 4073 544634 9383	3378444 3676 605796 9780	3439731 4119 544509 9337	3439807 4104 544433 9352	3439566 4083 544674 9373	3439491 4070 544749 9386	3439448 4057 544792 9399	*** 3441411 4163 534829 9293

Table 3-1. Cases selected to assess our registration in term of similarities between the fixed and warped images.

Cases	Case 48	Case 49	** Case 56	Case 57	Case 59
Weights	1 0 0 1	0 1 1 0	0 0 0 4	0.1 0.1 1 1	0.1 0.1 0 4
Iterations(Coarse,	225	222	217	217	217

Fine)	32	31	32	32	32
Time sec (Coarse, Fine)	7580 930	7540 937	7044 932	6992 918	7003 917
MI (percentage)	0.906933	0.906815	** 0.91053	0.906341	0.906175
Edge overlap (mm)	0.0255962	0.0255897	0.0255439	0.0256069	0.0255763
RMS intensity difference (intensity)	37.1901	37.1872	37.1914	37.1883	37.1914
TP TN FN FP (Number of pixel)	3439556 4067 544684 9389	3439557 4078 544683 9378	3439553 4059 544687 9397	3439510 4065 544730 9391	3439562 4067 544678 9389

Table 3-2. Cases selected to assess our registration in term of similarities between the fixed and warped images.

Unit of each metric is written in parenthesis in front of its name. We always compute the evaluation measures only on the overlapping regions of the input image data sets and the deformation fields, i.e. background regions in either of the data set are excluded from the computation. This is especially important in the case of synthetic transformations, where certain areas might vanish due to a shrinking like behavior, we always mark these regions with special values during synthetic transformation and we omit those marked regions from the evaluation process.

Therefore all the above results have only considered the patient area to evaluate the similarities between fixed and warped image. This is a very helpful fact in synthetic images which it supplies us with a “contour” to separate the desired regions. If we do not consider the contour while evaluating the similarities, obviously it will give us a fake and unreal percentage of high alignments and it causes high percentage of false-positive.

We have not been supplied by a clear contour in the real clinical patient cases, and it will force us to mask the images as described in the next part.

The criteria for choosing the weights in table 3-1 and 3-2 are as follows.

1- Select small weights as well as large weights to figure out the domain and barriers of the weights. Case 22, 24 and 28.

2- Select inverse combination of divergence weights and coarse weights such as cases 38 and 39 to see which operator has a bigger impact on the registration process and final warped image and transformation.

3- Assign weights to only coarse stage or only fine stage to see which stage is more critical and crucial for the registration process. Such as case 44 and 45.

4- Assign weights to divergence of coarse stage and curl of fine stage and vice verse to see the impact and consequences. Case 48 and 49.

5- Try out different values of weights for different spline operators of different stages to see whether it's better to keep the consistency in the weights and select the same values for every stages or even if we choose disordered weight values for the stages will we get a good alignment or not. Case 37 or Case 22.

6- Assign ZERO to three operators out of four, to see the results and impacts. Case 56.

7- Select a combination of weights based on the previous good results we obtained and see whether the combination of good results also lead to an acceptable alignment or not. Case 57 and 59, where case 57 is the combination of cases 44 and 45, and case 59 is the combination of cases 45 and 56.

All of the above results are the comparison between the fixed image and warped image. The detail discussion about the results and which are the best results among the above cases and its reasons will be debated in chapter 4 elaborately. Here we can only say the results of case 45 are completely satisfactory for this type of dataset and are shown with three stars (***). The second best case is the case 44 and 56 which is shown by two stars (**).

In the following figures 3-6 and 3-7 differences between fixed and moving images before registration and difference of fixed and warped images after registration are shown. Likewise in the figure 3-8 and 3-9 the final warped image resulted from our method is shown along with the fixed and moving image for the comparison. The below case belongs to Case 45 with weights of [0.1 0.1 0 0].

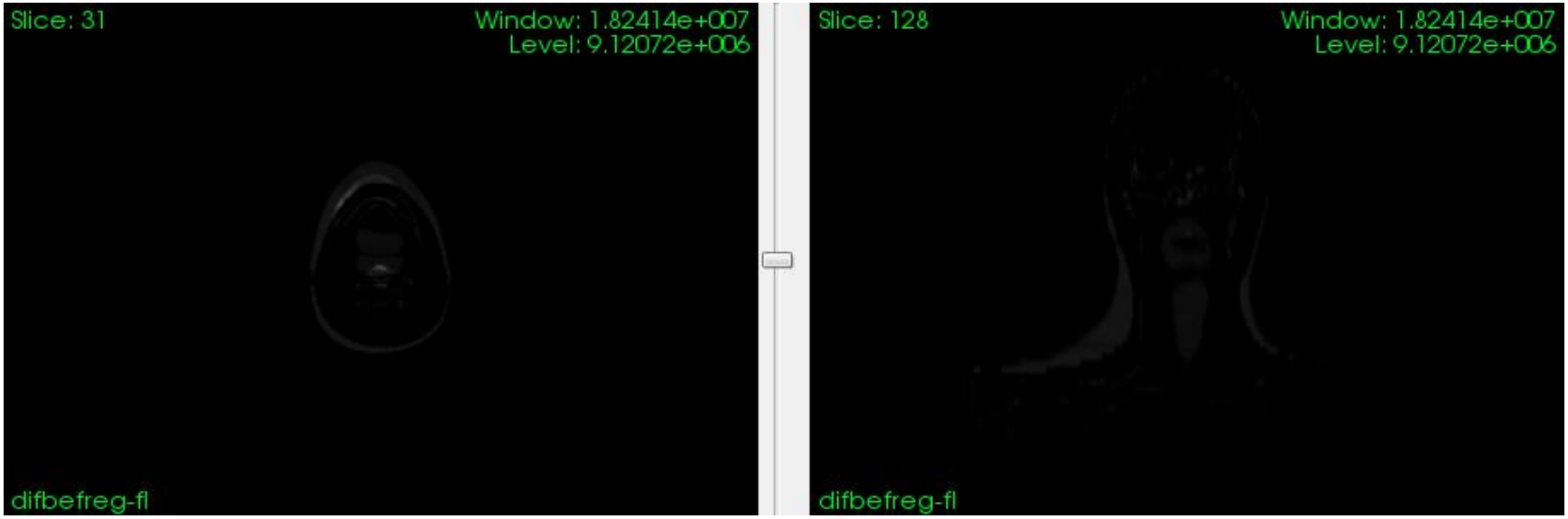


Figure 3-6. Difference before the registration between fixed and moving mages (axial and coronal view).

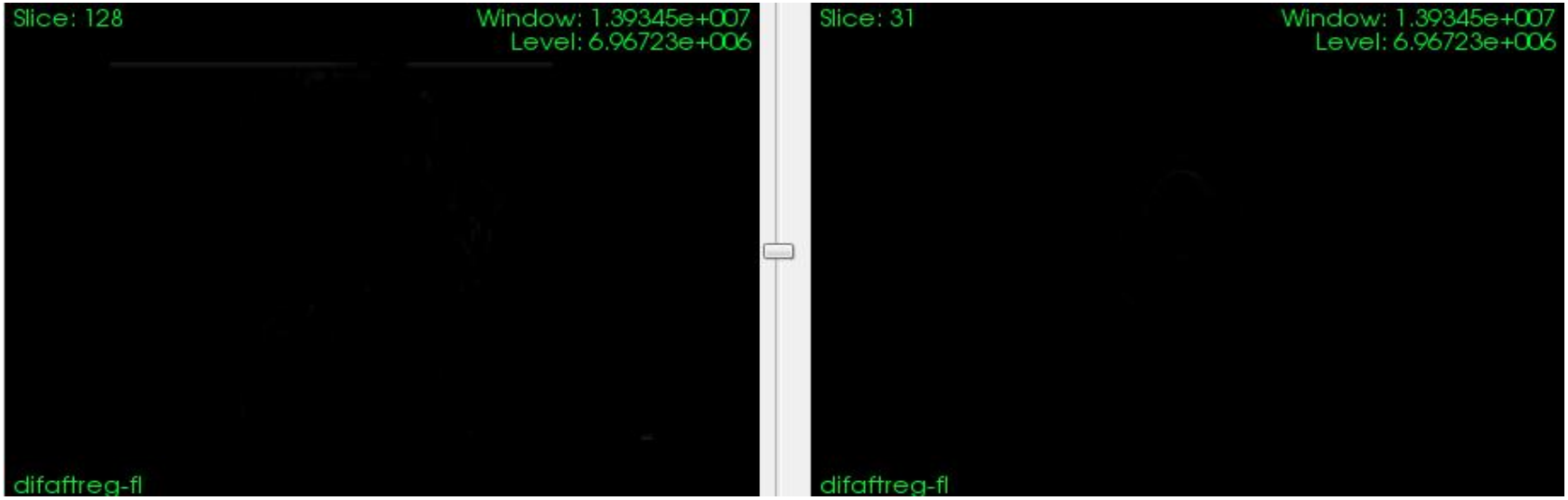


Figure 3-7. Difference after the registration between fixed and warped images (axial and coronal view).

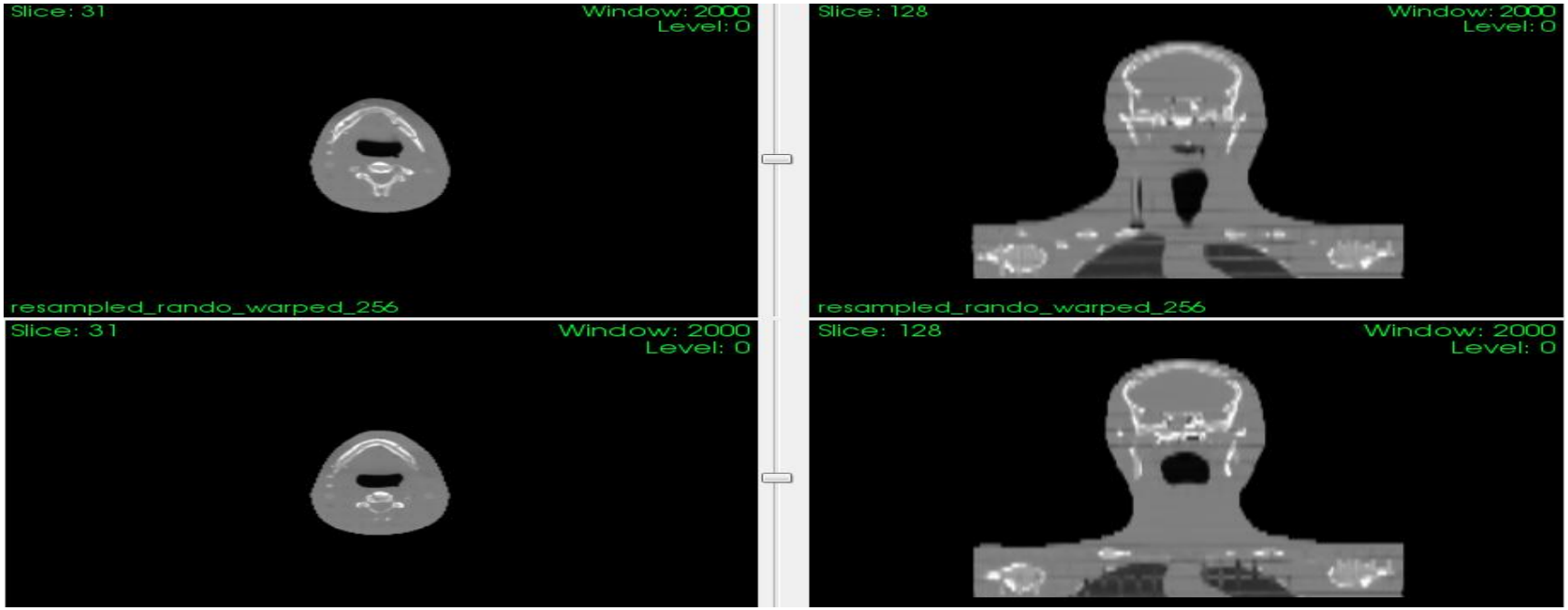


Figure 3-8. Two upper images are axial and coronal view of the fixed image and two bottoms belong to moving image.

Figure 3-9. Final warped image of case 45 with the weights assigned [0.1 0.1 0 0].

3-3-1-1- Histogram of differences

We have also obtained the differences of fixed and warped image in terms of histogram of the difference image. In the following screenshot, the code to be written in MATLAB is shown to compute the histogram. However the function *"readmha()"* is a function written in Plastimatch [9] and it is called here in MATLAB for reading meta-images (.mha and .mhd) files. We have done two approaches to calculate the histogram of differences. In the first idea, we directly took the histogram from the image in figure 3-7 which denotes the difference of fixed and warped image and this image is obtained using the application itself. In the second approach we used the function of MATLAB to take the differences of fixed and warped images. The code and the histograms can be seen in the following (Figure 3-10 and 3-11). We have done this evaluation method for only case 45.

Command Window

```
>> fixed = readmha('Images/resampled_rando_warped.mha');
>> differ = readmha('diffaft_C45.mha');
>> ind = find(fixed > -1200);
>> hist(differ(ind),128)
>> outlier=zeros(size(differ));
>> outlier(find(differ>100))=1;
>> imshow(sum(outlier,3))
>> output = readmha('out_warped_C45.mha');
>> differenza=output-fixed;
>> figure; hist(differenza(ind))
>> hist(differ(ind),128)
>>
```

Figure 3-10. The code written to take the histogram of differences of fixed and warped image for case 45. The very first 3 lines of codes represent the first approach to get the direct histogram, the rest lines of codes represent the second approach to use the MATLAB function and calculate the histogram.

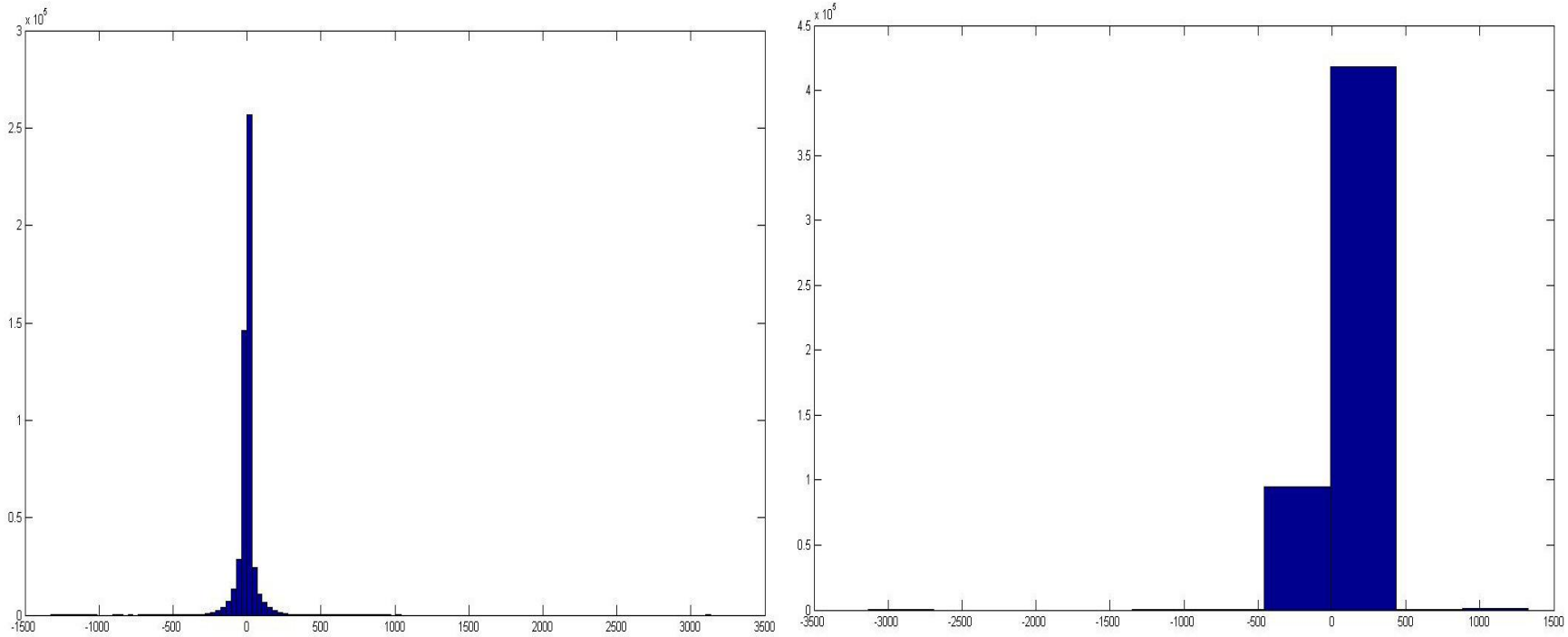


Figure 3-11. X axis represents intensity (grey value) of the pixels and Y axis represents number of pixels exist. Left: histogram of differences between the fixed and warped image obtained using the first approach of direct calculation of the difference image. Right: histogram of differences obtained using the MATLAB function to subtract the fixed and warped image from each other and take the histogram of differences.

3-3-2- Deformation fields evaluation

In this part the general idea is to assess the similarity of a synthetic ground-truth deformation field and the deformation field computed by our non rigid registration algorithm. To do this we use the metrics introduced in chapter 2 part of displacement field evaluation. In the table 3-3 and 3-4, we have presented the same cases in the previous part with the same specification of [15 15 15] number of grid points for coarse stage and [33 33 33] for fine stage. The cases in table 3-3 and 3-4 are the exact same cases discussed in table 3-2 and 3-3 about their image statistics similarities.

Cases	Case 22	Case 24	Case 28	Case 36	Case 37	Case 38	Case 39	** Case 44	*** Case 45
Weights	0.1 0.1 0.1 0.1	0.5 0.5 0.5 0.5	4 4 4 4	1.5 1.5 1.5 1.5	2 1.5 1 0.5	0 1 0 1	1 0 1 0	0 0 1 1	0.1 0.1 0 0
RMS displacement field (mm)	4.70518	4.70824	5.02648	4.71556	4.71577	4.7099	4.70593	4.70569	4.69505
Robust Max displacement field difference (mm)	11.3034	11.2998	11.6008	11.3014	11.3023	11.2994	11.2996	11.3033	11.2834
Min and Max Jacobian	0.73666 1.29941	0.735736 1.30038	0.659354 1.55562	0.735873 1.30305	0.734815 1.3036	0.73653 1.30043	0.73659 1.30034	0.73673 1.29909	0.746655 1.29941

Table 3-3. Differences of the synthetic DF and the DF obtained for each case with different weights. These cases are the exact same case as table 3-2.

Cases	Case 48	Case 49	** Case 56	Case 57	Case 59
Weights	1 0 0 1	0 1 1 0	0 0 0 4	0.1 0.1 1 1	0.1 0.1 0 4
RMS displacement field (mm)	4.7063	4.70949	4.70729	4.70602	4.7077
Robust Max displacement field difference (mm)	11.2998	11.2991	11.3029	11.3035	11.303
Min and Max Jacobian	0.736739 1.30034	0.736586 1.30043	0.736657 1.29909	0.736648 1.29941	0.736706 1.29941

Table 3-4. Differences of the synthetic DF and the DF obtained for each case with different weights. These cases are the exact same case as table 3-3.

Unit of displacement of DF is always in millimeter here, because every time we compute the DF between the fixed and image being deformed, the coordinate system is converted from pixel coordinate to physical coordinate system (world coordinate). As can be seen except some cases such as case 28, case 37 and case 45 the other cases have almost the same differences. This is considered to be due to the phantom images and its specification which has been gained by the synthetic deformation field. For this reason regulating the weights to obtain proper and acceptable weights which give us a totally discriminating result is hard to find. Also in terms of differences of deformation fields case 45 showed good result compared to the others.

Therefore for this type of images the best weights are considered to be **[0.1 0.1 0 0]** and it gives us the following quite smooth deformation field and a good alignment of the images.

In the following images, we have shown the original synthetic deformation field, the obtained deformation field of case 45 after the registration process and the jacobian image. (Figure 3-12)

For the jacobian values in the latter tables we have to mention that for this synthetic case we did not obtain any negative jacobian and the values are acceptable. Even if we assign bigger weights we do not go to the negative values. But as will be mentioned in the next part of clinical cases, we will encounter some cases with negative jacobian values.

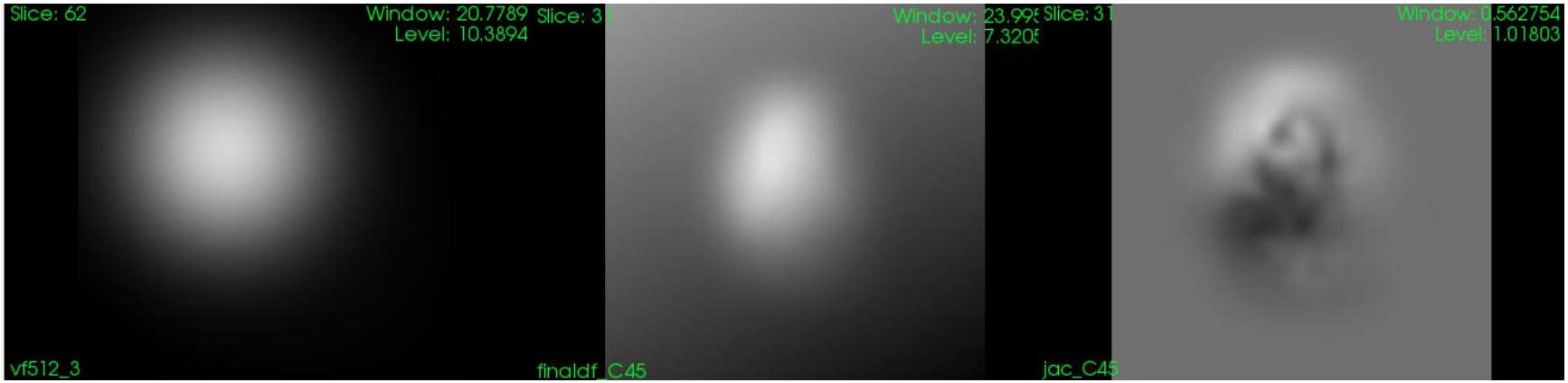


Figure 3-12. Left: Original deformation field which all the cases subtracted from it. Middle: obtained deformation field from case 45 after utilizing our registration method, Right: the jacobian image of the deformation field of case 45.

In order to get better perception about the divergence and curl of the deformation field we have stored div/curl images obtained during the registration process and can be seen in figure 3-13. The top left image shows the divergence of DF,

top right image is derivative image of the divergence obtained by optimizer during the convergence and iterations done for maximizing the cost function. The bottom left image shows the curl of the DF and bottom right is the derivative of curl obtained in the same way as divergence.

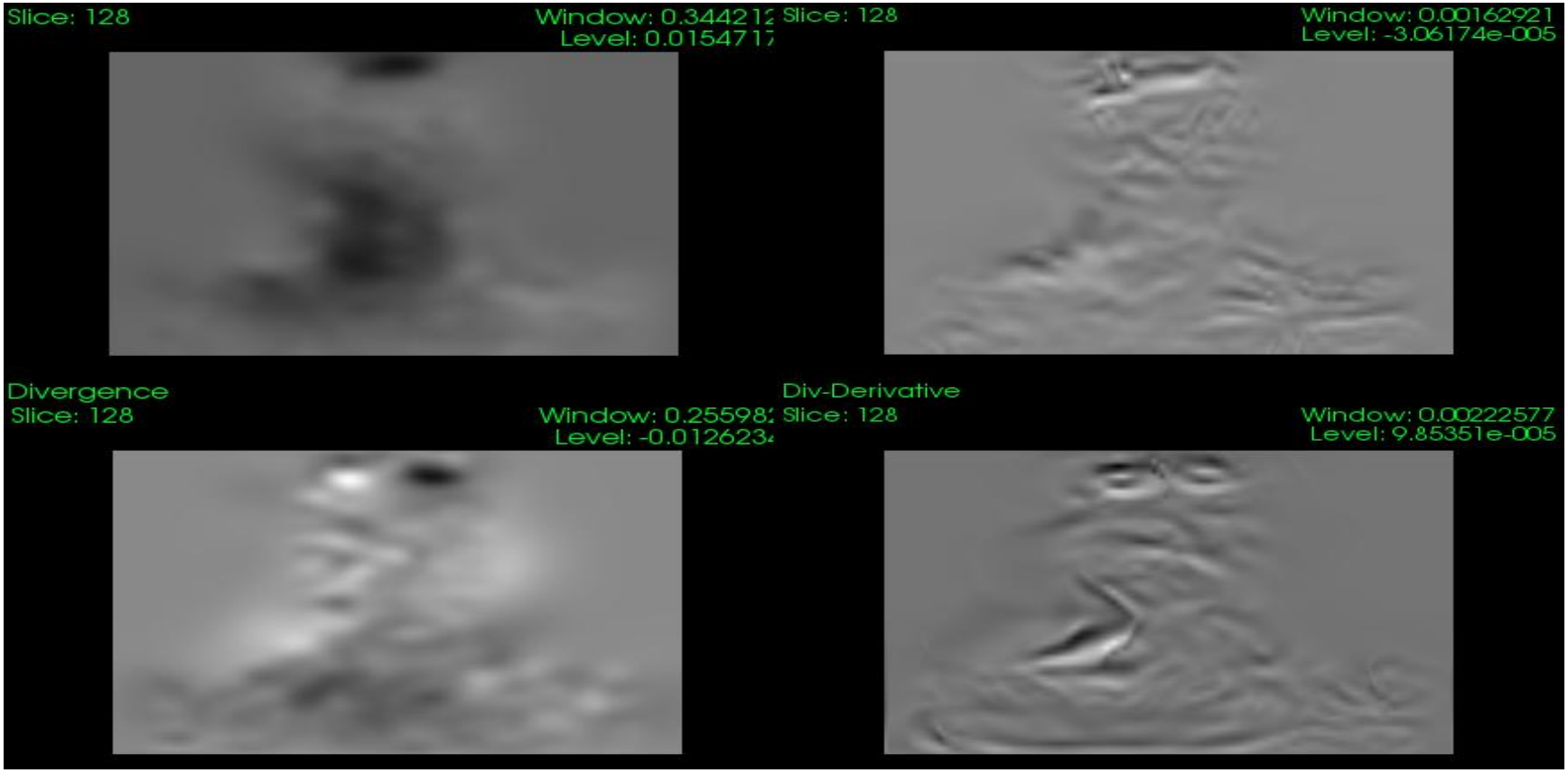


Figure 3-13. Top left: divergence of DF. Top right: derivative of the divergence obtained by the optimizer. Bottom left: curl of DF. Bottom right: derivative of curl of DF.

As can be seen from the results of both image similarities and the deformation field differences, the differences between the cases in term of several metrics such as RMS differences of intensity or displacement fields or TP, TN, FP, FN etc., may not be so sensitive and they have a small percentage of the differences. This is because of type of the images we use and because of essence and specification of the image which is a phantom and the deformation is done synthetically.

When we move to the real clinical cases, the differences between the cases becomes more obvious and discrimination between cases are much easier, as can be inferred in next part.

3-4-Real Clinical Patients

3-4-1- Approach and problems

After testing our method and application with the phantom dataset and evaluated using multiple metrics, this time we tried to test our algorithm on real patients.

As mentioned in part 2 of chapter 2, we have obtained this dataset from the repository library by R. Castillo et al [6], which provides researcher with image datasets, associated with a coordinate list of anatomical landmarks that have been manually identified and registered by an expert in thoracic imaging.

We chose three cases to assess our method. The criteria were the selection of one case with minimal deformation (Case1) and two cases (Cases 7 and 8) with the largest deformation.

Problems

The type of the images is analyze header format (.img) which need to be converted to meta-images format (.mha or .mhd) in order to be loaded in our application. For this reason, we used Plastimatch application [9] *"writemha()"* function to convert two types of images. This function reads the matrix of images loaded into MATLAB using the original instruction and writes it as .mha file. However in the phase of conversion we had to consider the difference in orientation between Matlab and C++ in writing the meta-images. Hence we did apply a three lines of codes to fix it properly. The following images show the code written for the image conversion and the loaded image for case 7.

```
imageImprt
case_t50 = double(case8_T50);
case_t50 = case_t50-1024;
for sl=1:128
case_t50_r(:,:,129-sl)=case_t50(:,:,sl)';
end
writemha('Case8_T05.mha',case_t50_r,[0 0 0],[0.97 0.97 2.5],'short')
```

Figure 3-14. A Matlab code to convert the analysis row image format (.img) to meta-image format (.mha) using the Plastimatch function.

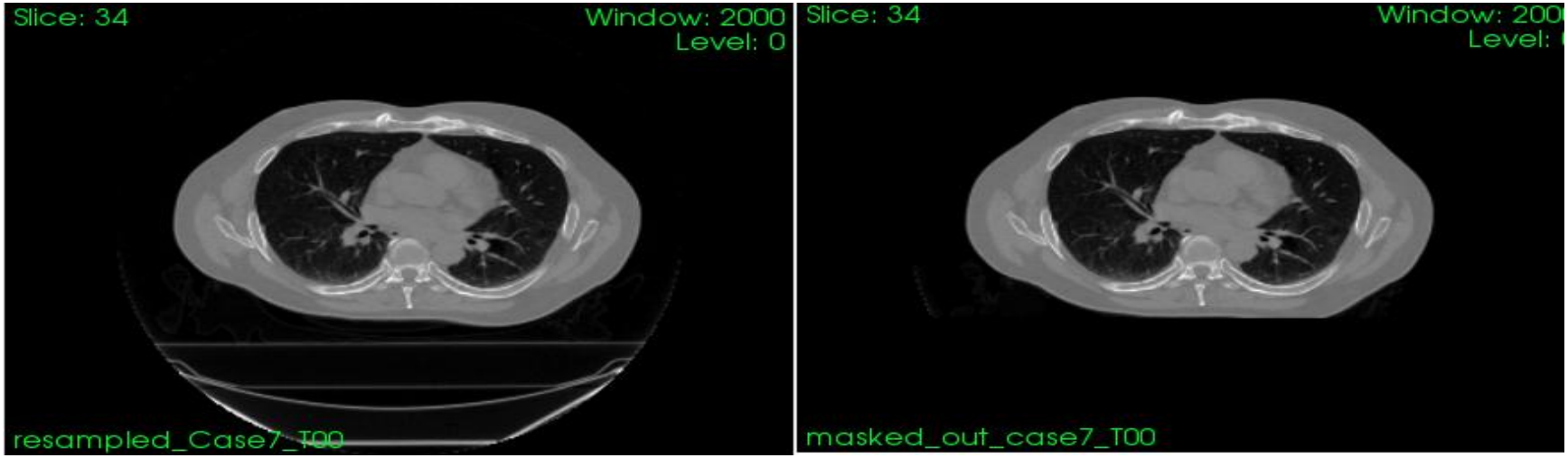


Figure 3-15. Axial view of the case 7 for the beginning phase of inhale of the patient. Left: The artifacts are clear in the background. Right: masked image using the Plastimatch functions to remove the couches out of the image.

Having looked at figure 3-15 the major problem emerges here. The main problem we encountered while trying to use this dataset as our input was the problem of discrimination between patient area and the background. In left image of Figure 3-15 the presence of a couch is clear and these structures needed to be removed in the evaluation stage.
One of the main differences between the synthetic cases and the real clinical patient cases is the extraction of the actual patient region. Depending on type of the real image, normally there are some methods of feature extraction using e.g. preprocessing unsharp masking or fuzzy filters.
For tackling this problem we utilized two functions of Plastimatch to mask the images. There are two specialized functions designed in Plastimatch to remove external structures around the patient from the image.

"Plastimatch_mask imgname.mha mask_imgname.mha". using this execution code, it creates a mask file namely "mask_imgname.mha" out of the main image. Then after creating the mask by using the following execution instruction, it masks out the couch and extracts the patient area.
"mask_mha imgname.mha mask_imgname.mha -1200 masked_imgname.mha". As can be seen in the instruction, the value of -1200 is assigned to the mask area of the background. This is a very useful fact which creates us a synthetic contour for discrimination. We will use this value to remove out the background and only consider the patient area in the evaluation. Right part of figure 3-15 shows the masked image.

3-4-2- Images similarity and deformation fields evaluation

Now that we have masked images, we are ready to perform the registration evaluation. In the following tables (3-6, 3-7 and 3-8) we report the results obtained using the evaluator metrics. As noticed, in this case we are not supplied by the synthetic deformation field to compare it with our cases, therefore we mostly used the jacobian of the deformation field to assess our transformation, using negative percentage of jacobians and the inverse consistency. For the similarity between the fixed and warped image, we used the prior metrics. In table 3-7 one important fact has to be remarked. There are two types of cases in this table denoted by for instance “Case1_T05” and “Case1_T50”. The first name denotes the direct registration which means, giving “case1_t00.mha” (inhale) as fixed and “case1_t50.mha” (exhale) as moving. The second name denotes the inverse registration with substituting the fixed and moving image to evaluate the capability of inverse consistency of our registration (3-4-3). The direct and inverse registrations are shown next to each other in the side by side columns. Table 3-6 shows the several experiments assigned to only case 1: we will discuss about the consequences and the conclusion in chapter 4. In table 3-8 experiment results of case 7 and 8 are illustrated.

Cases	Case1_T05_4	Case1_T05_5	Case1_T05_8	Case1_T05_11	Case1_T05_12	Case1_T05_13
Weights	**0.5** **0.5** **0** **0**	**0.1** **0.1** **0.1** **0.1**	**1** **1** **1** **1**	**0.5** **0.5** **0.1** **0.1**	**0.5** **0.5** **1** **1**	**1** **1** **0** **0**
Number of iteration Coarse Fine	400 78	381 82	500 500	420 84	420 500	500 90
Time(sec) Coarse Fine	16877 3471	16076 3271	23166 19985	19334 3826	17682 20541	21049 4005
Min Jacobian	0.325311	0.308853	-0.45822	0.303755	-0.459611	0.296989
Max Jacobian	1.729	1.66886	4.61871	1.56392	4.53601	1.53156
Percentage Jac >1 Jac<1	61% 38%	61% 38%	61% 38%	61% 38%	61% 38%	62% 37%
NMI (percentage)	0.598792	0.595763	0.571714	0.595221	0.571485	0.593535
Edge	0.0751109	0.0748579	0.0836427	0.0747558	0.0834799	0.074983

Overlap (mm)						
RMS Intensity Difference (intensity of pixel)	36.6033	36.62	39.4657	36.6778	39.3991	36.7891
Center of Mass (Reference) [x,y,z] ---------- (Warped) [x,y,z]	119.514 107.508 128.481 ---------- 119.143 106.784 129.139	119.514 107.508 128.481 ---------- 119.135 106.785 129.139	119.514 107.508 128.481 ---------- 118.919 106.801 129.214	119.514 107.508 128.481 ---------- 119.122 106.797 129.146	119.514 107.508 128.481 ---------- 118.949 106.788 129.192	119.514 107.508 128.481 ---------- 119.086 106.808 129.172

Table 3-6. Several experiments done on case 1 direct registration.

Cases	Case1_T05_2	*Case1_T50_2	Case1_T05_3	*Case1_T50_3	Case1_T05_6	*Case1_T50_6
Weights	**0.1 0.1 0 0**	**0.1 0.1 0 0**	**0 0 1 1**	**0 0 1 1**	**0.5 0.5 0.5 0.5**	**0.5 0.5 0.5 0.5**
Number of iteration Coarse Fine	381 82	316 36	380 500	319 96	401 78	327 38
Time(sec) Coarse Fine	16125 3291	13417 1483	16051 19490	13547 3823	16908 3116	14858 1656
Min Jacobian	0.307128	0.538915	-0.450691	0.436796	0.336578	0.50761
Max Jacobian	1.66851	1.78141	4.96959	2.06648	1.72964	1.83097
Percentage Jac >1 Jac<1	61% 38%	32% 67%	59% 40%	35% 64%	61% 38%	33% 66%
NMI (percentage)	0.596175	0.574306	0.57518	0.559891	0.608383	0.573302
Edge Overlap	0.0748265	0.0839756	0.0839348	0.0854573	0.0742535	0.0832862

(mm)						
RMS Intensity Difference (mm)	36.618	36.9941	39.224	37.513	35.6149	36.7859
Center of Mass (Reference) [x,y,z]	119.514 107.508 128.481	119.45 105.678 131.162	119.514 107.508 128.481	119.45 105.678 131.162	119.514 107.508 128.481	119.45 105.678 131.162
----------	----------	----------	----------	----------	----------	----------
(Warped) [x,y,z]	119.134 106.787 129.139	119.011 106.011 132.388	119.285 106.499 129.451	118.975 105.898 132.645	119.151 106.779 129.142	118.965 106.13 132.005

Table 3-7. Direct registration and inverse registration done on three cases. Cases signed with (*) are the inverse registration cases. The best case is case1_T05_6 signed with three stars.

Cases	Case7_T05_1	*Case7_T50_1	Case7_T05_2	Case8_T05_1	*Case8_T50_1	Case8_T05_2
Weights	0.1 0.1 0 0	0.1 0.1 0 0	0.5 0.5 0.5 0.5	0.1 0.1 0 0	0.1 0.1 0 0	0.5 0.5 0.5 0.5
Number of iteration Coarse Fine	500 496	500 464	500 451	500 500	500 500	500 500
Time sec Coarse Fine	17165 14926	17243 14007	17128 13834	18672 15109	16879 14319	17668 14730
Min Jacobian	0.245716	0.198681	0.260167	-0.222756	-0.00652866	-12.62
Max Jacobian	2.28283	2.09276	2.30337	2.54068	3.29677	14.0191
Percentage Jac >1 Jac<1	60% 39%	44% 55%	60% 39%	55% 44%	46% 53%	43% 56%
NMI (percentage)	0.522228	0.516843	0.555335	0.482285	0.480806	0.162026
Edge Overlap	0.0768829	0.0793446	0.0750648	0.088021	0.0874627	0.190305

(mm)						
RMS Intensity Difference (mm)	52.0865	51.2415	50.0157	48.5576	48.2949	86.199
Center of Mass Reference [x,y,z]	244.182 253.027 170.169	244.031 252.885 170.172	244.182 253.027 170.169	248.248 269.575 159.483	248.289 269.474 160.115	248.248 269.575 159.483
----------	----------	----------	----------	----------	----------	----------
Warped [x,y,z]	244.286 252.895 170.142	244.343 252.68 170.109	244.278 252.902 170.151	248.316 270.505 160.052	248.241 269.89 160.308	241.711 321.853 164.133

Table 3-8. Result evaluation of case 7 and case 8. Cases signed with (*) are the inverse registration cases.

Here the obvious facts are the main differences between these real clinical cases and the synthetic cases. Real cases are much more sensitive to changes in the weights. Depending on the specific case, a small change in the weights will have a big impact on the final DF, hence on the warped image. For instance in case8_T05_1 (large deformation between inhale and exhale), when we assign the weights of [0.1 0.1 0 0] the negative jacobian is almost close to 0. But for the same case with the weights of [0.5 0.5 0.5 0.5] (case8_T05_2) the negative jacobian of DF goes to -12, while with the same weights of [0.5 0.5 0.5 0.5] we have got the best results for case1. This means for case 8 we can not go further the latter weights. This fact demonstrates the very crucial fact that assigning and choosing the weights strictly depends upon the type of images and deformation they have.

The criteria for choosing the weights are almost the same criteria we chose for the synthetic cases, however we tried to assign ubiquitous weights to cover the whole assumed values. The best weights for case 1 is considered “Case1_T05_6” with [0.5 0.5 0.5 0.5], weights which has the best NMI (Normalized Mutual Information) and quite acceptable minimum jacobian values.

For case 7 the same weights as case 1 resulted in a good similarity between fixed and warped image and also a smoother DF, however for case 8 it seems that if we go further than [0.1 0.1 0.1 0.1] we are going to have non smooth DF, hence we ought to keep the weights very small.

In the following figures we have shown the fixed and moving image belonging to case 1 and the warped image resulting from our registration “Case1_T05_6” with the best weights of [0.5 0.5 0.5 0.5]. Also the DF and its jacobian and eventually divergence and curl obtained during the registration process are depicted.

Figure 3-16. Case 1_T05_6. Left: Fixed image. Middle: Moving image. Right: Warped image

Figure 3-17. Case1_T05_6. Left: Divergence of the DF. Right: Curl of the DF.

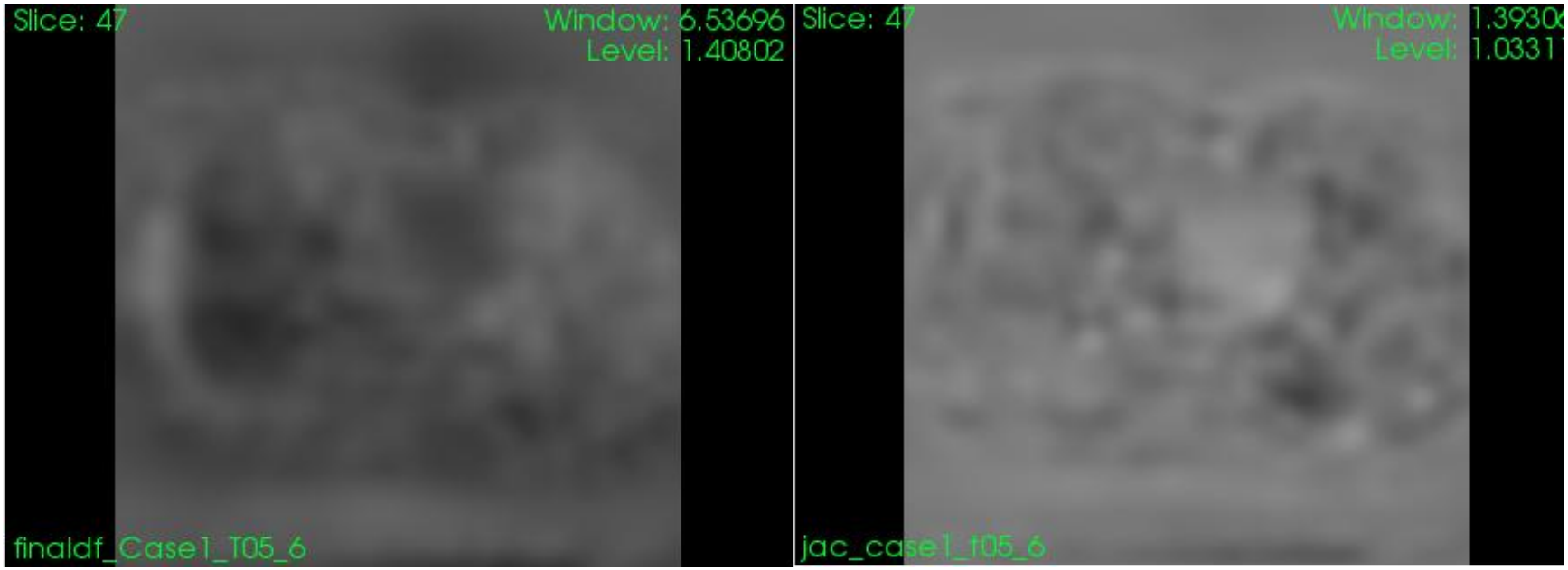


Figure 3-18. Case1_T05_6. Left: Final DF (Transformation). Right: Jacobian image of the DF

3-4-3- Inverse consistency

In this part we evaluate what is the potential of our method for consistency of the direct and reverse registration. As mentioned before a stable deformation field has to be invertible and with a non-singular determinant matrix. This is mostly identified by positive jacobian determinant matrix of the final DF. However in this experiment we have chosen case 1, 7 and 8 from the clinical dataset and we have tested direct and reverse registration on the cases. We chose three sets of weights to be assigned to case 1, as can be seen from table 3-9. But for the cases 7 and 8 we only chose one set of weights. In the direct registration the fixed image is the inhale phase and the moving image is the exhale phase. However in inverse registration strategy, we substituted the fixed and moving image. For both registrations, we obtained the final deformation field and computed the differences of both DF. Obviously, the difference should be close to zero. In the following table we have calculated the RMS and Max displacement field difference between the DF of direct and inverse registration, as obtained using our evaluator application.

Cases	Case1_2	Case1_3	Case1_6	Case7	Case8
Weights	**0.1** **0.1** **0** **0**	**0** **0** **1** **1**	**0.5** **0.5** **0.5** **0.5**	**0.1** **0.1** **0** **0**	**0.1** **0.1** **0** **0**
RMS displacement field Difference of Direct and Reverse registration DF (mm)	2.97728	2.88039	2.87151	2.60121	3.5601
Robust Max displacement field difference (mm)	6.31834	5.96702	6.02053	4.37401	7.41414

Table 3-9. Difference of direct and inverse registration done in real clinical patient dataset to evaluate the inverse consistency of our method

3-4-4- Landmark errors evaluation

In this part, we will exploit the landmarks individuated on each fixed and moving image and extract the RMSE between our warped landmarks and the landmarks of the original dataset. The main schema of the work is illustrated in the following figure.

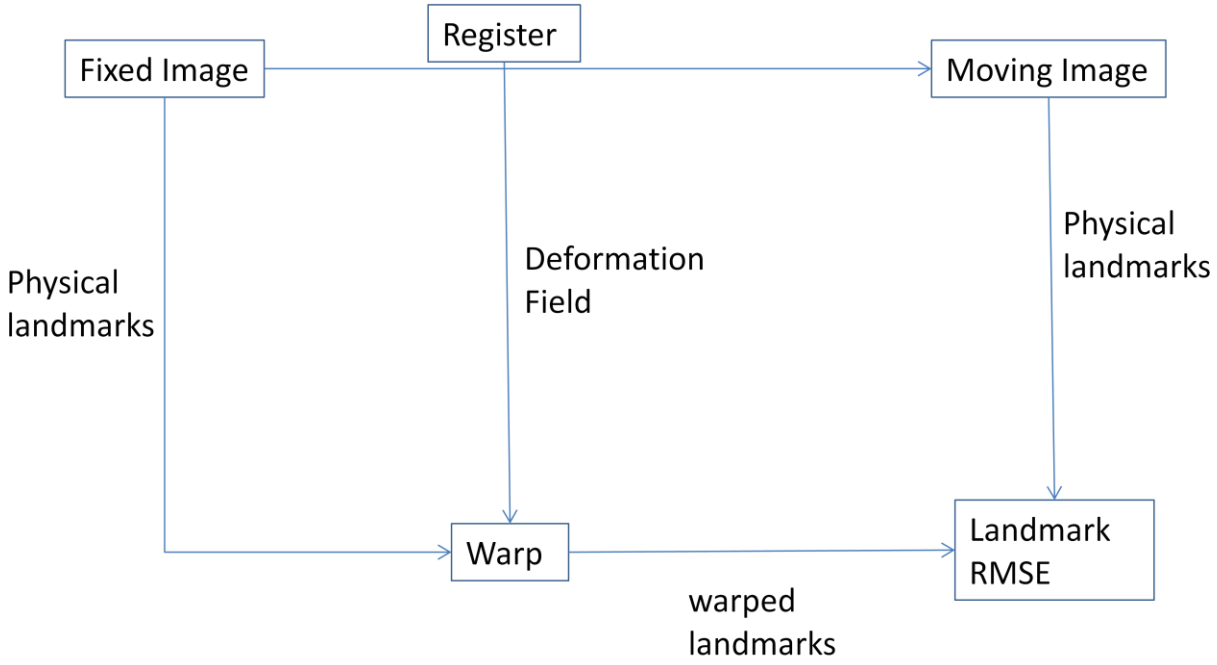


Figure 3-20. Schema of performing landmark error evaluation

According to the above diagram, after doing the registration between fixed and moving images, the final DF is obtained and kept.

Our evaluation method runs as follows:
1- First landmarks on the fixed image are warped with deformation field obtained from the registration.
In output we obtain a set of physical coordinates of landmarks warped according to the DF.

2- We compute the median and percentiles of the distance between warped landmarks and the landmarks of the moving image, in order to evaluate the error after deformable registration.

Since we were provided with landmarks in pixel coordinates, we had to convert them to physical coordinate by simply.

$$Physical\_Coordinates = origin + Pixel\_Coordinates * spacing$$

Results for cases 1, 7 and 8 are presented in table 3-10.

Cases	Case1	Case7	Case8
Weights	0.5 0.5 0.5 0.5	0.5 0.5 0.5 0.5	0.1 0.1 0 0
Landmarks RMSE (mm)	1.2675	5.1889	9.0056
[25th 75th] Percentile of the distribution of RMSE (mm)	[0.8798 1.5935]	[2.4241 12.8365]	[2.5161 21.2152]

Table 3-10. RMSE and 25th and 75th percentile of the landmark errors for case1, 7 and 8.

In the following figures the overlaid warped landmarks (blue) and the original landmarks (red) on the moving image for cases 1, 7 and 8 respectively are illustrated. The X axis shows the physical coordinates of each landmark in X direction and Y axis denotes the physical coordinates in Z direction.

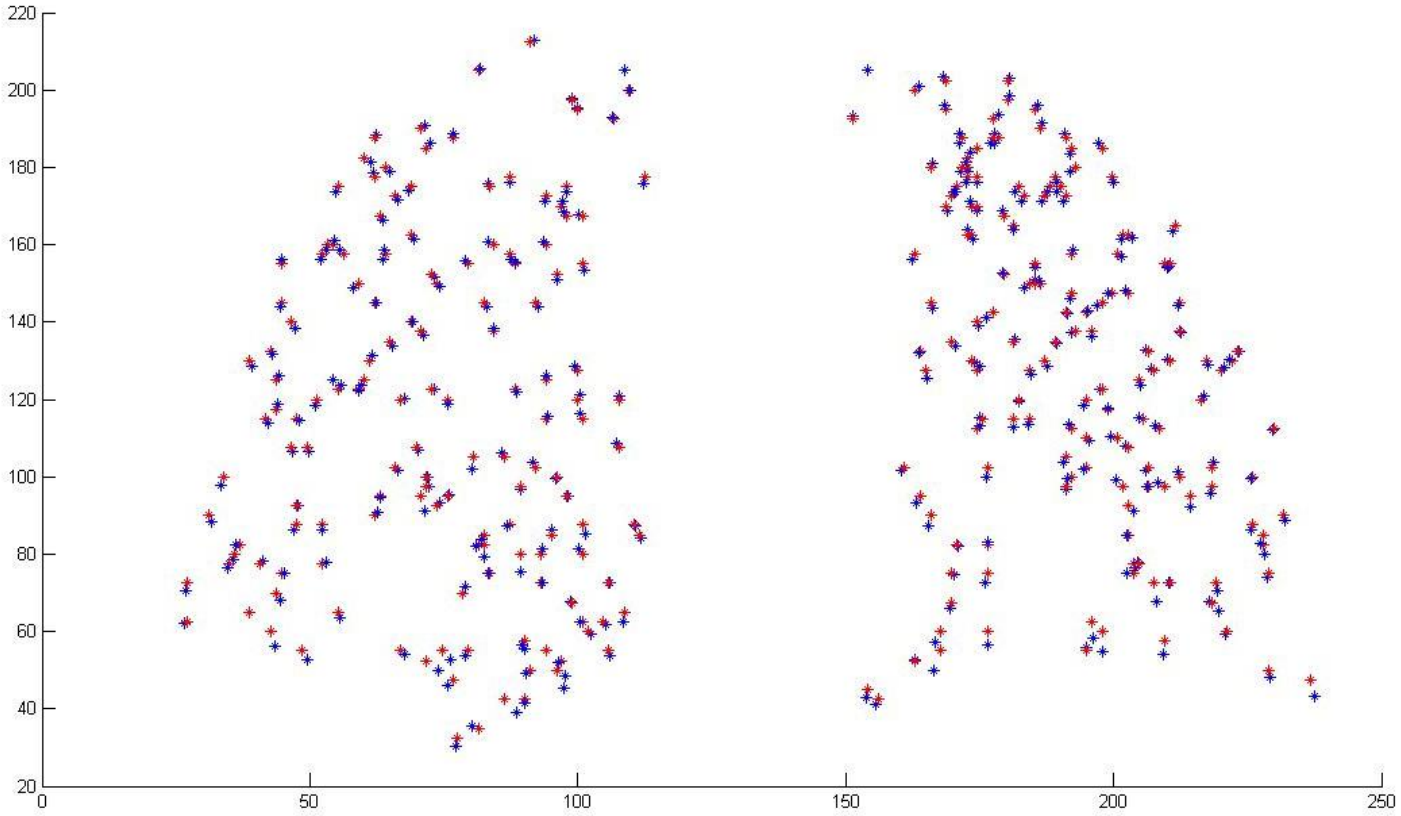


Figure 3-21. Warped landmarks are overlaid on original landmarks of the moving image for Case1. X axis shows the physical coordinates of each landmark in X direction and Y axis denotes the physical coordinates in Z direction.

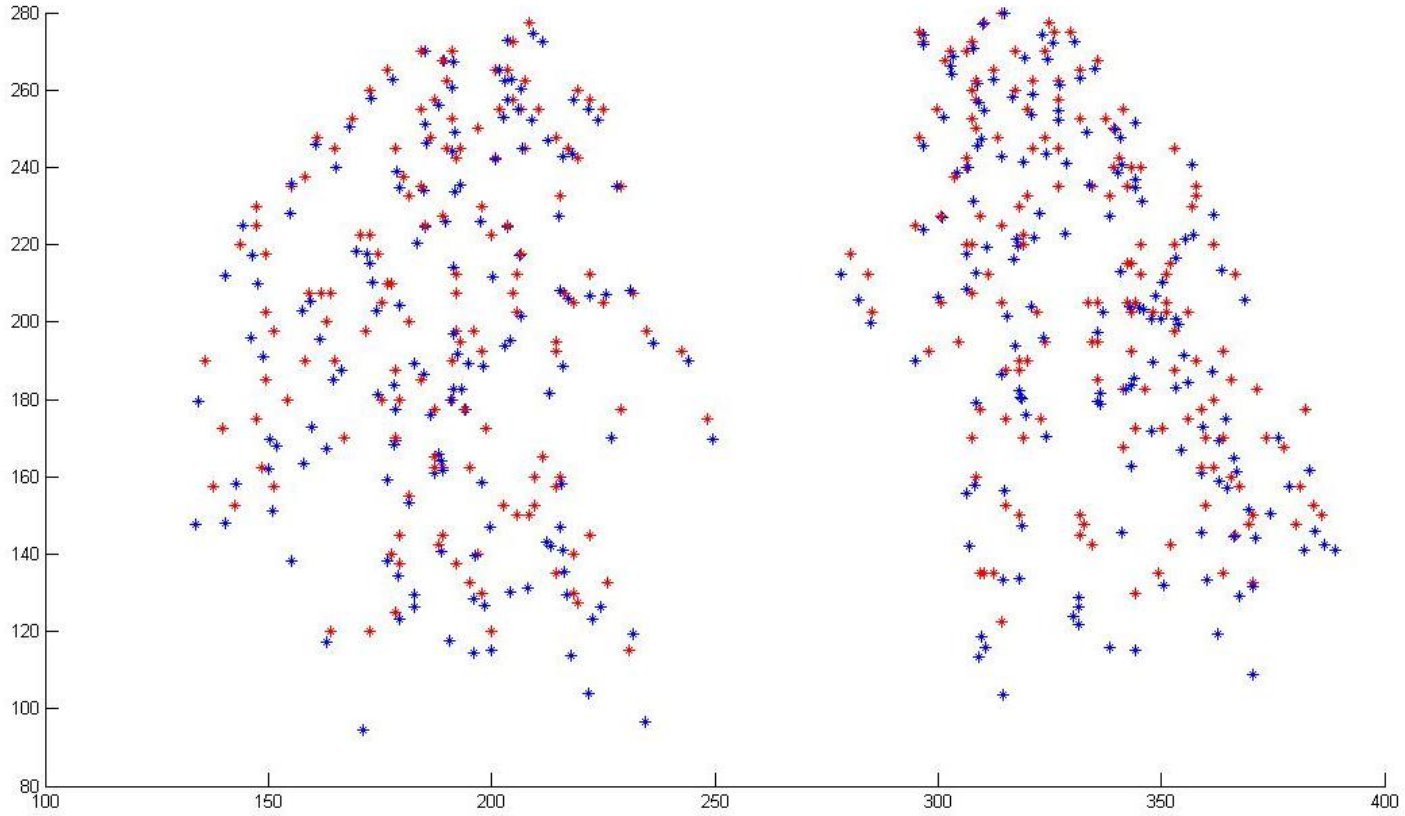


Figure 3-22. Warped landmarks are overlaid on original landmarks of the moving image for case 7. X axis shows the physical coordinates of each landmark in X direction and Y axis denotes the physical coordinates in Z direction.

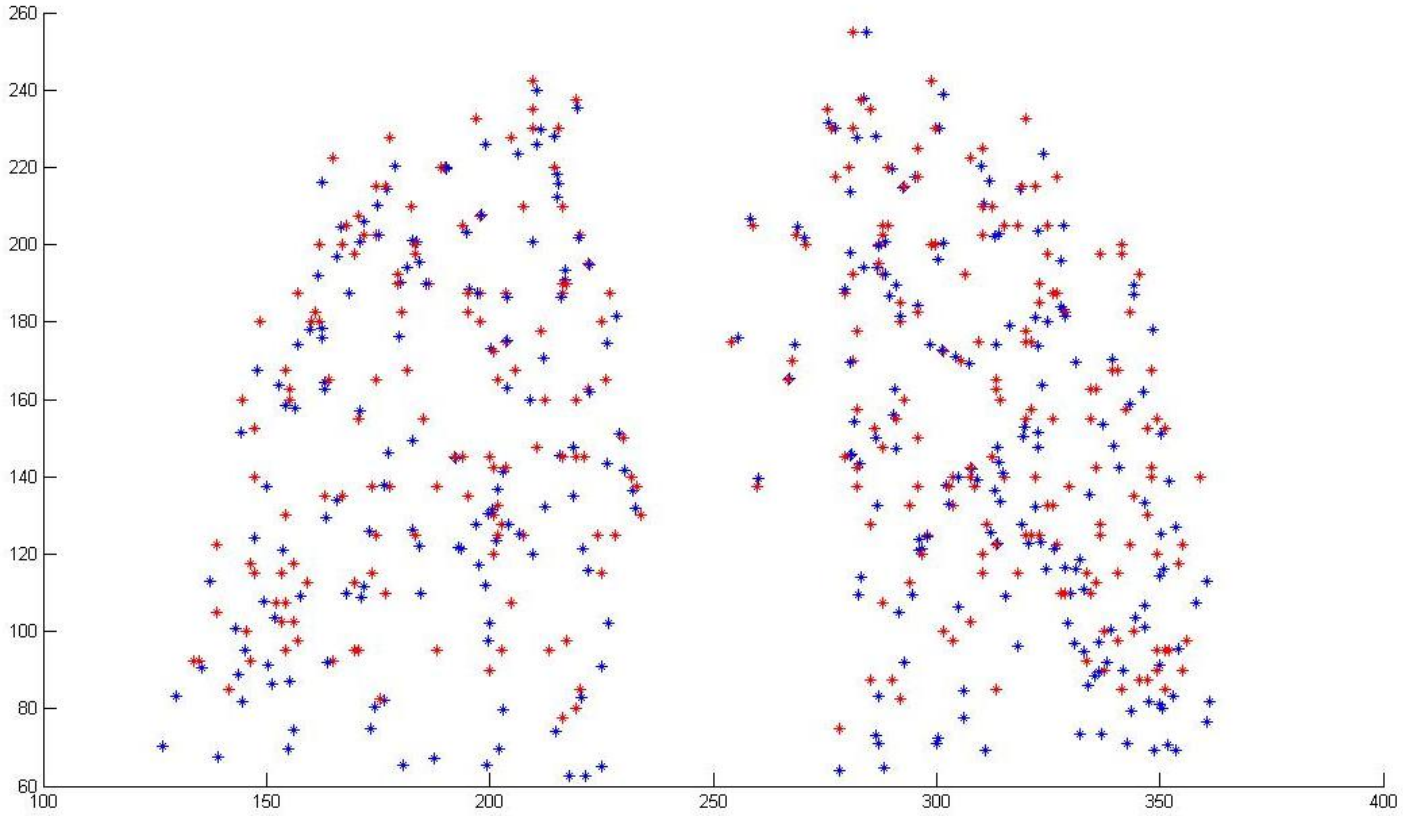


Figure 3-23. Warped landmarks are overlaid on original landmarks of the moving image for case 8. X axis shows the physical coordinates of each landmark in X direction and Y axis denotes the physical coordinates in Z direction.

For each case considerable landmarks are annotated on the images. As it can be seen from the figure 3-21, in Case 1 which has less deformation between the fixed and moving image in terms of inhale and exhale phase of the patient, we have little disparate positions comparing our landmarks (red) and the original landmarks (blue). For case 7 and especially

case 8 with the most deformation, some points have a slight different comparing original landmarks and our landmarks, but as the annotated landmarks show on image 7 and 8 (figure 3-23 and 3-24) some other points have small different. However the major differences are presented in table 3-10. The latter facts prove that our method could yield to an acceptable result for the cases with a reasonable deformation, but it has potential of improvement for the cases which have large deformation between fixed and moving images.

3-5-Optimal Experimental Registration Parameters

3-5-1- Introduction

In the last part of this chapter we will use our registration application and method to infer and obtain optimal registration parameters. These parameters are completely dependent upon the method of registration one is using and also mainly the type of cost function and images, modalities are influential criteria for choosing registration parameters. In our method we have chosen three types of pre-registration parameters to obtain their optimum domain. These parameters are

- Number of iterations
- Number of BSpline grid points
- Subsampling resolution

Predicting these parameters before the registration helps us to drive the registration procedure to a good alignment of warped image and a smooth DF. Also it helps the procedure to converge sooner and save the computational time.

Providing a mathematical framework for the prediction of these parameters by using the non-linear regression and classification method are suggested in chapter 4 as a future work.

In the next part we have used the first dataset of synthetically obtained phantom images which we used in part of 3-3 to obtain the optimal domain of the latter registration parameters. Hence for the evaluations we used the same evaluators to assess the image similarities and the synthetic DF to assess the DF obtained for each case. In figure 3-24 the schema to obtain optimal registration parameters is depicted.

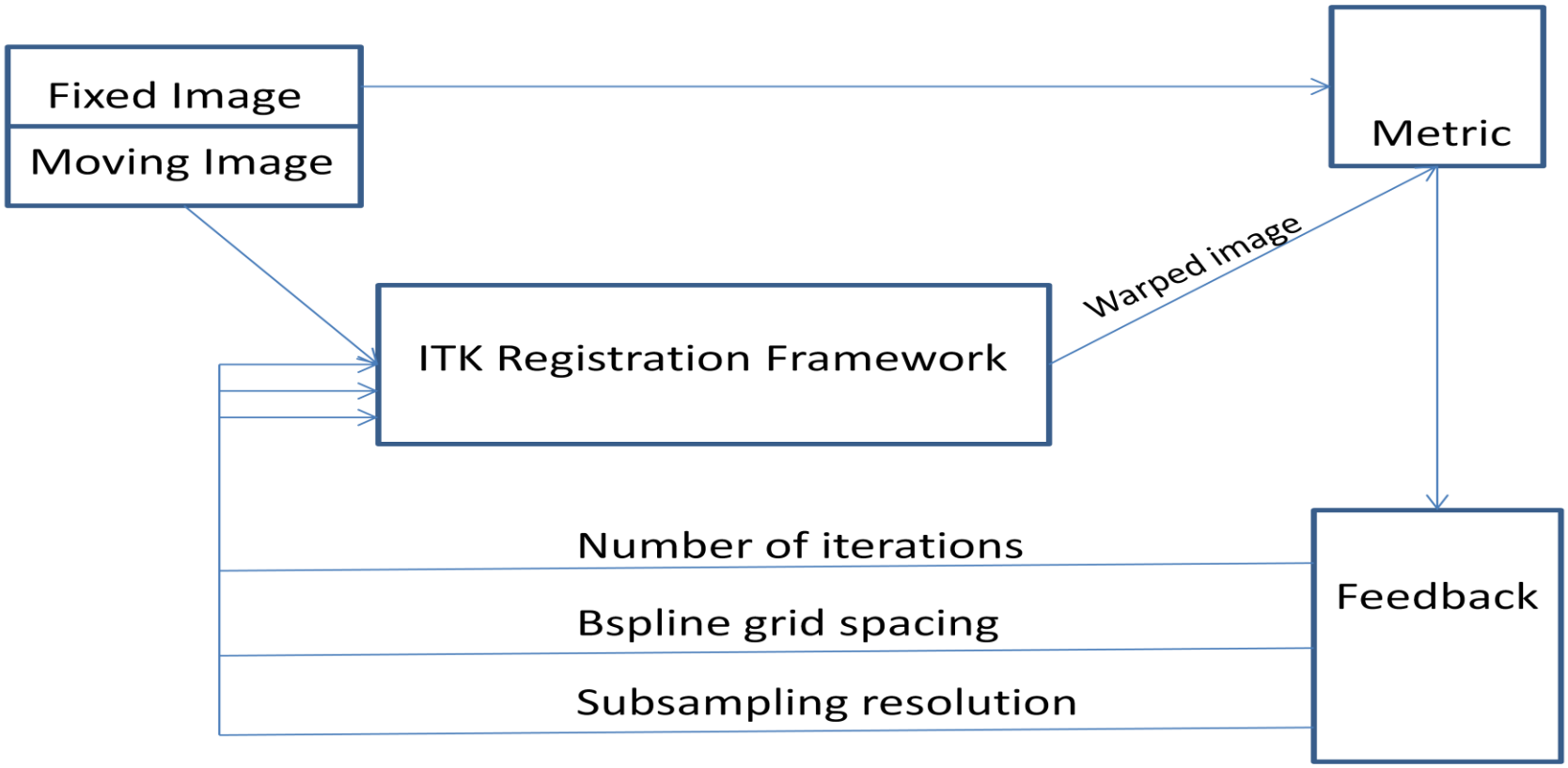


Figure 3-24. The major schema to obtain optimal registration parameters.

3-5-2-Optimum Number of iterations

Number of iterations in a registration procedure is considered to be one of the most crucial parameter which completely depends on the stopping criteria of the optimizer. Once the optimizer reaches to convergence, the process is stopped and the number of iterations is obtained.

In our cases the number of iterations is changed by giving different weights. By looking at the previous table of cases, the obtained optimal number of iteration for the weights of [1 1 1 1] has been estimated as 250 (Table 3-11).

Coarse Grid Size	5 5 5	12 12 12	20 20 20
Fine Grid Size	20 20 20	30 30 30	40 40 40
Coarse Grid spacing (mm)	119 119 90	43 43 32	25 25 18
Fine Grid spacing (mm)	25 25 18	16 16 12	12 12 9
Number of iterations	558	250	229
Time	11354 sec	8215 sec	15466 sec

Table 3-11. Optimum number of iteration by changing the number of grid points.

Table 3-11 shows the number of iterations needed to reach convergence as a function of the number of grid points. The smaller the grid size (larger number of bspline grid points) the smaller is the number of iterations, though it globally takes more time. That's because the time spent for each iteration takes more time.

Resolution	1 1 1	2 2 1	4 4 1
Number of iteration	711	250	190

Table 3-12. Optimum number of iteration by regulating the subsampling resolution

Table 3-12 shows, depending on the desired sub-sampling resolution, the most optimum number of iterations for this particular cost function.

3-5-3-Optimum Number of grid points

With the weights of [1 1 1 1], among the latter 3 number of grid points [5,12,30] for coarse grid and [20,30,40] for fine grid stage, and the specifications previously mentioned, the number of grid points of [12, 30] has the best result as can be seen from the following table. Only the best results are denoted by three stars (***) from now on. The subsampling resolution is [2 2 1]. In table 3-13 and 3-14 the similarities in terms of image statistics and the difference between the synthetic deformation field and the obtained final deformation field are shown.

Image statistics

Gird size	[5 5 5], [20 20 20]	*** [12 12 12], [30 30 30]	[20 20 20], [40 40 40]
Number if iterations	558	250	229
Mutual Information	0.905124	0.908364	0.886361
RMS Intensity Difference	37.2582	37.2091	38.1429
Edge overlap	0.0211785	0.0255223	0.037737

Table 3-13. Image metric statistics in term of similarities between the fixed and warped image when we try different number of grid points.

Deformation field statistics

Grid size	*** [5 5 5], [20 20 20]	[12 12 12], [30 30 30]	[20 20 20], [40 40 40]
RMS displacement field ground truth	4.6711	4.71556	4.61966
Robust Max displacement field difference	11.1554	11.3014	11.2441
Median displacement field difference	1.75981	1.83897	1.81432

Table 3-14. The difference of synthetic DF and the DF obtained from each of the cases above

However it must be mentioned that the default grid size which ITK considers is [5 ,20].

3-5-4-Optimum Subsampling resolution

The most optimized subsampling resolution in terms of both image statistics and deformation field with the weights of [1 1 1 1], grid spacing of [12, 30] and with the previously mentioned registration specifications is as follows:

Image statistics

Resolution	*** 1 1 1	2 2 1	4 4 1
Number if iterations	711	250	190
Mutual Information	0.921241	0.908364	0.831214
RMS Intensity Difference	20.3522	37.2091	53.6506
Edge overlap	0.0146348	0.0255223	0.0472356

Table 3-15. Image statistics for regulating the subsampling resolution to obtain the most optimum resolution.

Deformation field statistics

Resolution	*** 1 1 1	2 2 1	4 4 1
RMS displacement field	0.580043	4.71556	4.94684
Robust Max displacement field difference	9.43157	11.3014	12.0336
Median displacement field difference	0.146679	1.83897	1.41625

Table 3-16. Differences of synthetic DF and DF obtained for each above cases to obtain the most smoothed DF in term of regulating the subsampling resolution.

As shown by the table the more sample we have, the better result we get. The result obtained by the 512x512x123 resolution is dramatically better than 256x256x61, in terms of both image statistics between fixed and output warped image and deformation field statistics between synthetic and final deformation field. Thus here the optimum subsampling resolution is [1 1 1].

However both results of resolutions [1 1 1] and [2 2 1] reached to a completely better convergence comparing with only mutual information or sum of squared differences cost function registration, which will be discussed in chapter 4. Also the optimum maximum and minimum step length, relaxation factor and tolerance of the optimizer could be evaluated as well as a future work.

Chapter 4
Discussion and Summary

In this chapter, we will discuss about several specifications that are behind the results we obtained in chapter 3. In the first part, we will have a look at the results of synthetic cases and clinical cases both at the same time, in order to find some rules and regulations for choosing the optimal weights. Since the main influential component of our method is the weights that will be assigned to the div/curl parameters before the registration, we will try to give guidelines on how to choose the weights (W_{dc}, W_{cc}, W_{df}, W_{cf}), their domain and barrier based on results.
We will also try to focus on each selected case to see how they affect the similarities of final warped image and smoothness of final DF. Then we will compare our method and the cost function which contains only MI and only sum of squared differences (SSD) as its similarity metric to figure out the added benefits and critical differences of embedding div/curl of the DF as a regularization function, both for synthetic and real cases.
Lastly the whole summary of our method is presented and future works are suggested.

4-1- Choice of weights and Comparisons

As presented in chapter 2 regarding our cost function, it contains only two types of weights (w_d and w_c) to balance the scale imposed between the terms inside the cost function. These weights represent a trade-off between the recoveries of dissimilarities exist between the fixed and deformed images and irregularities emerging in the final DF in terms of expansion/compression. In a multi-resolution scheme, the weights are defined for each stage from coarsest to finest resolutions, thus leading in our case to the need of assigning four of them in total.

The first task was to define reasonable ranges for w_d and w_c.
Finding a confidence interval in image registration means to have a high percentage of overlap between fixed and deformed image, along with a smooth and regular DF. Since no ground-truth and/or literature indication was available, we experimentally determined the weights using a restricted set of images.
The method for the evaluation in this part follows by merely presenting the image statistics obtained from the evaluation of similarities between the fixed and warped images in each case and also the differences of deformation fields and the jacobian of the transformation.
In the synthetic dataset, the images are not much sensitive on regulation of the weights and forcing a large deviation between the weights in the cases may not show evident impact on the evaluation metrics. The reason of this fact is due to the type of phantom images we have been using to evaluate our method. The results we obtained for the clinical cases are clearer and more discriminative than synthetic cases. For this reason, in some parts of the discussion clinical cases help us reach to more accurate consequences for choosing the weights and decide the rules.
The method we used to exploit the relationship among weights is mostly based on Design Of Experiments (DOE) method, which states that if we keep one coefficient stay constant, and change the others, the trends ruling the whole experiment can be highlighted. Therefore in each relationship, we tried to keep one or two weights constant (e.g. W_{dc}, W_{cc}), and change the others (e.g. W_{df}, W_{cf}) to evaluate the results, and vice-versa. Finally based on the outcome of the metrics, we infer the underlying trend to the whole dataset.

4-1-1-Relationship between the stages

The aim of this evaluation is to study the relationship between div/curl weights in the coarser stage vs. the finer stage, i.e. the relationship between (W_{dc}, W_{cc}) and (W_{df}, W_{cf}).
We selected the following cases among all the experiments in real patients to see which stage has a larger impact on the final result in terms of both image similarity and DF regularity (table 4-1).

Cases	**Case1_3**	**Case1_13**	**Case1_12**	**Case1_11**	**Case1_15**
Weights	**0** **0** **1** **1**	**1** **1** **0** **0**	**0.5** **0.5** **1** **1**	**0.5** **0.5** **0.1** **0.1**	**0.5** **0.5** **0.01** **0.01**
NMI	0.57518	0.593535	0.571485	0.595221	0.595741
RMS Intensity differences	39.224	36.7891	39.3991	36.6778	36.6647
Min Jacobian	-0.450691	0.296989	-0.459611	0.303755	0.306231

Table 4-1. Cases chosen to evaluate the rule among the coarse stage and fine stage.

If we compare case1_3 and case1_13, we see that case1_13 has higher MI and lower RMS differences with a positive minimum jacobian value, whereas case1_3, which has larger weights in fine grid stage, has worse results including a negative jacobian value. Likewise, the analysis of case1_12, featuring larger weights to div/curl of the fine stage, shows worse results than case1_11 and case1_15, where the coefficients values in coarse grid are higher than in the fine grid. Further lowering of the weights in the finer stage (case1_15 vs. case1_11) results in no evident benefit in terms of the evaluation metrics.

The reason of this fact can be explained as follows. The assignment of larger weights to the finer stage may prevent the compensation of small residual intensity differences, whereas the use of large weights for the coarse stage is helpful to obtain a final regular deformation field. Therefore the first and obvious regulation is to consider larger weights for coarse grid stage than finer grid stage.

4-1-2- Relation between divergence and curl

Besides analyzing the relationship between finer/coarser weights, we look for connections between divergence and curl weights, i.e. between (W_{dc}, W_{df}) and (W_{cc}, W_{cf}). Therefore, keeping W_{dc}, W_{df} fixed, we changed the values of W_{cc}, W_{cf} (and vice-versa) exploring the evaluation metric space. We select inverse combination of divergence weights and coarse weights such as cases in table 4-2 to see which operator has a bigger impact on the registration process and final warped image and transformation.

Cases	Case 38	Case 39
Weights	0 1 0 1	1 0 1 0
Iterations(Coarse, Fine)	222 31	217 32
Time sec (Coarse, Fine)	7201 902	7079 927
NMI	0.905006	0.908678
Edge overlap	0.0255506	0.0256353
RMS intensity difference	37.1893	37.1886
RMS displacement field	4.7099	4.70593
Minimum Jacobian	0.73653	0.73659

Table 4-2. Investigation on impact of divergence and curl on the whole multi-resolution registration.

From table 4-2, we can see no appreciable difference between regularizing just with curl or divergence, whereas this is important to avoid the presence of compression/expansion and/or vortexes in the final DF.

4-1-3- Relationship between the stages and div/curl

In a further analysis, we look for a relationship between (W_{dc}, W_{cf}) and (W_{cc}, W_{df}). Results for two paradigmatic cases are presented in table 4-3.

Cases	Case 48	Case 49
Weights	1 0 0 1	0 1 1 0
Iterations(Coarse, Fine)	225 32	222 31
Time sec (Coarse, Fine)	7580 930	7540 937
MI	0.906933	0.906815
Edge overlap	0.0255962	0.0255897
RMS intensity difference	37.1901	37.1872
RMS displacement field	4.7063	4.70949
Minimum Jacobian	0.736739	0.736586

Table 4-3. Investigation on impact of absence of curl in coarse grid along with the divergence of fine grid and vice versa.

As it can be seen from the table 4-3, if we remove the influence of curl from coarse stage and div from the fine stage or vice versa, there is no strong impact on the final result.

4-1-4- Suggested weights interval

Finally, we analyzed the performance of the new cost function to establish maximum and minimum domain of the weights, as well as to study diversity and consistency among them.
We report five significant combinations of weights for a single selected real case in table 4-4 and we look at variations in the evaluation metrics.

Cases	Case1_10	Case1_6	Case1_14	Case1_8	Case1_1
Weights	**0.01** **0.01** **0.01** **0.01**	**0.5** **0.5** **0.5** **0.5**	**0.65** **0.65** **0.65** **0.65**	**1** **1** **1** **1**	**1.5** **1.5** **1.5** **1.5**
Number of iteration Coarse Fine	379 76	401 78	500 94	500 500	500 500
Time(sec) Coarse Fine	17316 3615	16908 3116	20964 3755	23166 19985	21170 19568
Min Jacobian	0.309977	0.336578	0.287362	-0.45822	-1.58048
Max Jacobian	1.65233	1.72964	1.55494	4.61871	5.93768
Percentage Jac >1 Jac<1	61% 38%	61% 38%	62% 37%	61% 38%	58% 41%
NMI	0.597866	0.598383	0.594263	0.571714	0.560407
RMS Intensity Difference	36.7051	36.62	36.8202	39.4657	40.8095

Table 4-4. Real clinical cases to evaluate the maximum/minimum domain and barrier of the weights

By looking at the above table, case1_6 is the best case in terms of NMI, RMS intensity differences and the minimum values of the Jacobian determinant. In fact, by assigning progressively increasing weights, the NMI value is decreasing and some discontinuities in the final vector field are introduced, as testified by a negative minimum Jacobian value for case1_8 and case1_1. This fact shows that if we go further than [0.5 0.5 0.5 0.5], especially values above [1 1 1 1], we are going to incur into a non-smoothed final DF, as well as large amount of misalignment.

This becomes also clearer if we look at the div/curl values reserved in the log file during the registration. For example:
Iteration number: 1 , MI: 0.844013 Div: 0.323801 Curl: 0.147235. Total: 1.31505
Iteration number:181, MI: 0.933435 Div: 8.13914 Curl: 11.1747. Total: 20.2089
Iteration number:500, MI: 0.93893 Div: 11.0001 Curl: 15.2978. Total: 27.2368

As we iterate further, the difference between the scale of three metrics of the cost function, div/curl and MI is expanding, thus progressively reducing the MI contribution to the cost function in favor of the regularization terms. This is the case for weights equal to [1 1 1 1] (case1_8), whose final output NMI is not good and it has negative Jacobian values in the final DF.
Therefore generally the suggested interval of the div/curl weights is as follows:

$$\mathbf{0.1} < W_{\boldsymbol{d}}, \boldsymbol{W_c} < 1$$

These rules and regulations can be used for any type of dataset to choose the proper weights empirically, but the final values we obtain from the rules cannot be generalized.

4-2-Comparisons and discussion

In this part we will compare the best cases among each dataset with the corresponding only-MI and only-SSD (Sum of Squared Differences) registration technique, in terms of both similarities between the fixed and warped image and smoothness of the final transformation. The goal of this part is to determine what is the percentage of improvement by applying our registration technique, compared to other techniques which utilize the same type of similarity metric with no regularization term.

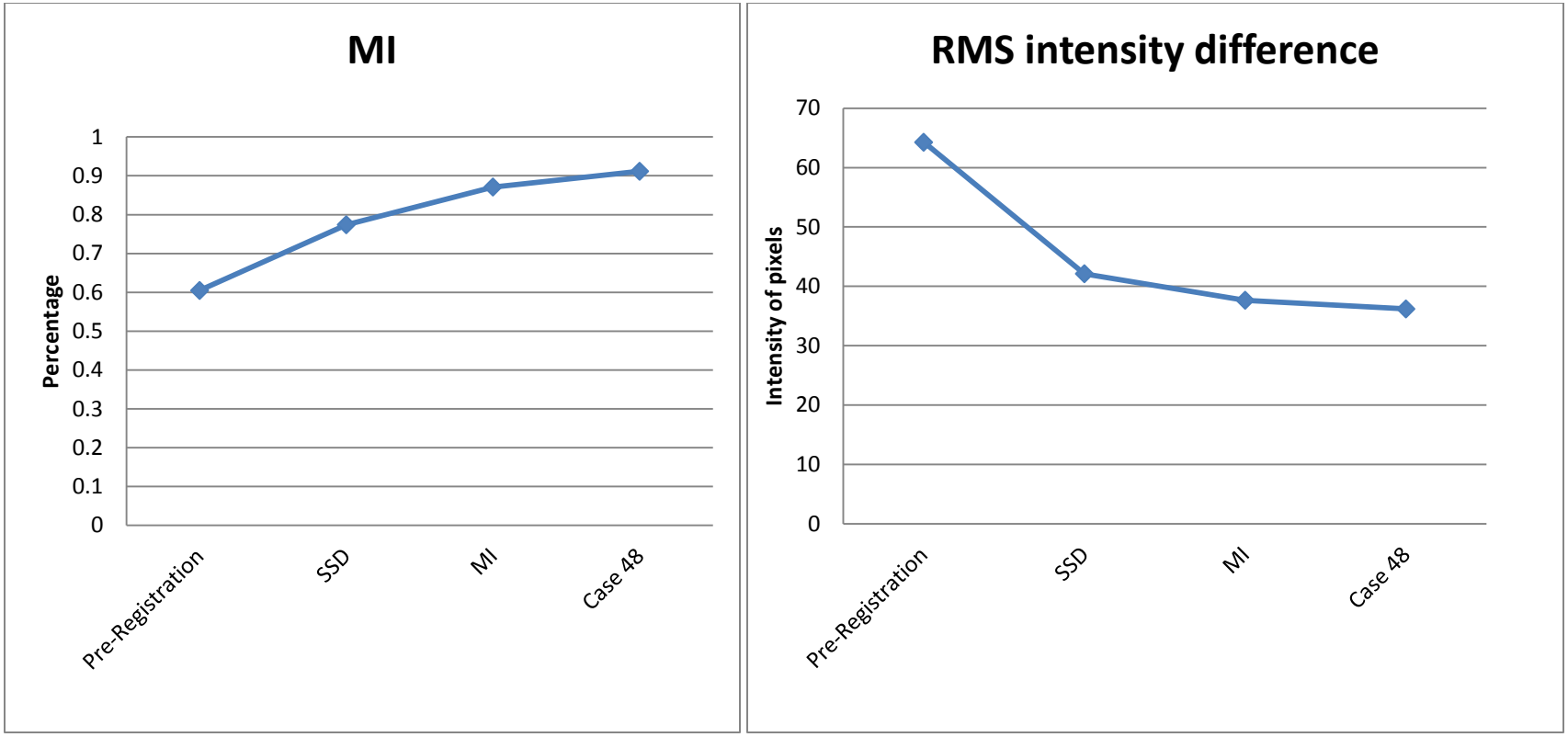
4-2-1- Comparison on synthetic cases

In order to compare our method with the registration techniques of only-MI and only-SSD, as mentioned in very first part of chapter 2, they must profit of the same pre-registration parameters as we did in our procedure. In the following table we present the results of case 45, which is the best case among the synthetic cases. Number of grid points for our registration and only MI registration are set to [12 12 12] for coarse stage and [30 30 30] for fine stage. For SSD number of grid nodes is set to [12 12 12] while (1 1 1) grid point coincides with the first pixel on the fixed image. Also alignment of the fixed and moving images is evaluated before registration, to compare it with after registration and compute the improvement. In the following table two final metric (RMS displacement field and minimum value of the jacobian) are considered for evaluation of the DF for each case, which subtract the synthetically obtained DF from final DF of each case.

Cases	Before Registration	SSD	Only-MI	Synthetic Case 45
Weights	- - - -	- - - -	- - - -	0.1 0.1 0 0
Iterations(Coarse, Fine)	-	34(*)	145 16	217 32
Time sec (Coarse, Fine)	-	396(*)	370 37	7087 958
MI (percentage)	0.604718	0.773927	0.871079	0.911669
Edge overlap (mm)	0.132376	0.0291568	0.025162	0.0246439
RMS intensity difference (intensity of pixel)	64.2719	42.121	37.6573	36.1901
RMS displacement field (mm)	6.09318 (+)	5.58623	5.09406	4.69505
Minimum Jacobian Value	0.731078 (+)	0.741036	0.779301	0.746655

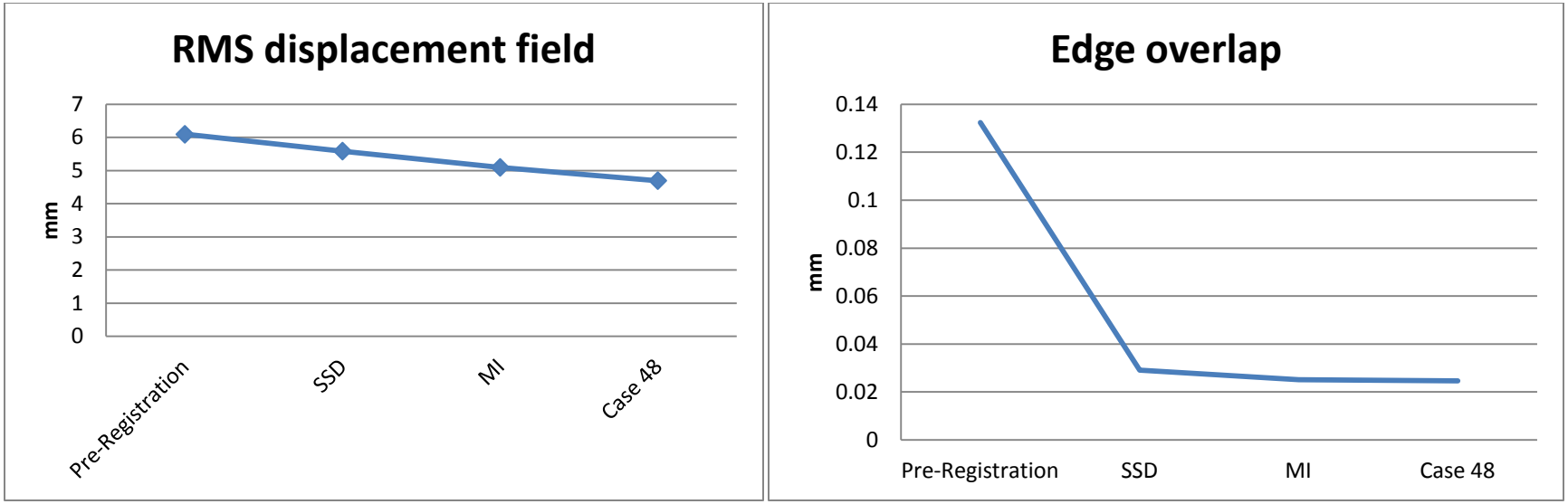
Table 4-5. Comparison of best case in synthetic dataset with only-MI and SSD registration technique. (*) For SSD registration there is no BSpline coarse and fine grid stages and all the transformation process is done in one stage.

The following graph depicts the graph of enhancement comparing two other types of registration (only-MI and only-SSD) with our method in terms of MI and RMS intensity differences between the fixed and warped image.



Graph 4-1. Left: improvement of MI between the fixed and warped image. Right: improvement of RMS intensity difference.

Also the following graph depicts the RMS displacement field differences between the synthetic DF and final DF obtained for each case as well as edge overlap.



Graph 4-2. Comparison on RMS displacement field difference and edge overlap.

In the following table, percentage of enhancement for each metric are shown comparing to our method.

Cases	Before Registration	SSD	Only-MI
MI	33%	15.1%	4.65%
Edge overlap	81%	15.4%	2.1%
RMS intensity difference	77.5%	16.3%	4%
RMS displacement field	29%	18.9%	8.5%

Table 4-6. Percentage of improvement comparing case 48 of our method with only-MI and only-SSD registration technique used with the same registration parameters.

For proving the above statistics in the following image 4-1, the comparison of the best case 48 of our registration only-MI and only-SSD are presented visually. We have overlaid the warped image on final DF in order to compare the deformations after the registration.

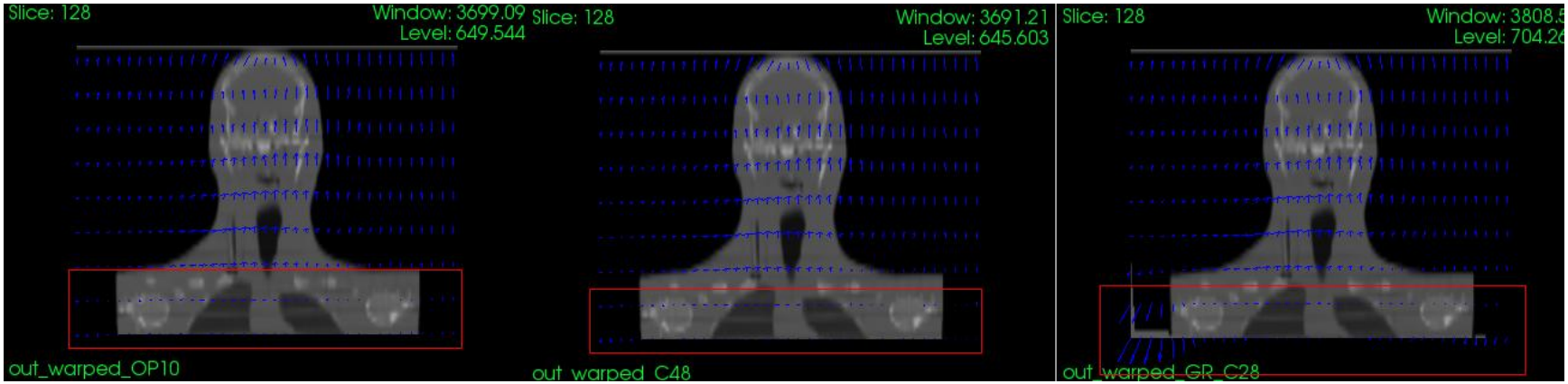


Figure 4-1. Comparison of Left: Only-MI registration with Middle: Our best case 48 and Right: only-SSD technique.

The part with the most deformation is bordered with a red rectangle. These areas are the most discriminative part of this registration with the synthetic dataset. Left image shows registration done with only-MI technique and the middle image is the case 48 of our registration method.

4-2-2- Comparison on clinical cases

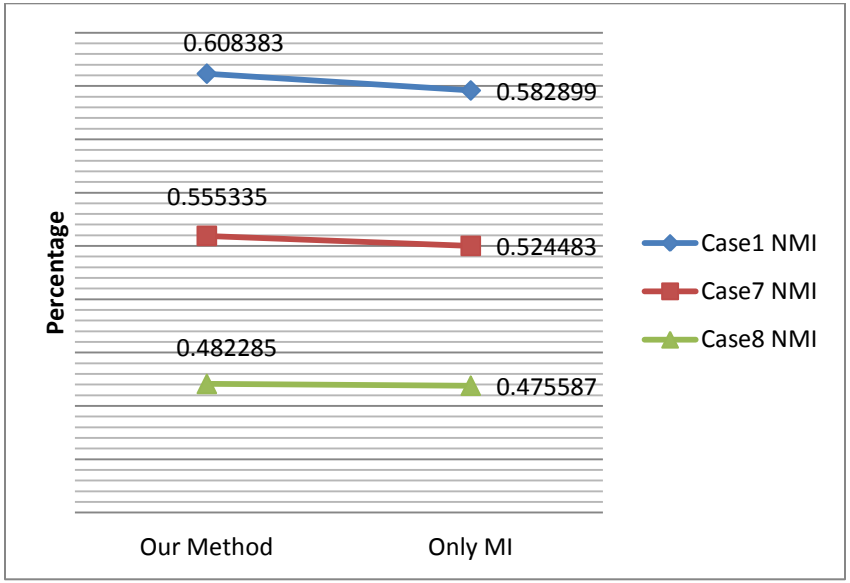
We have done the same evaluation in terms of enhancement of alignment and smoothness of DF for the clinical cases, comparing the best weights of our registration method with only-MI method. The experiment is done for case1, case 7 and 8 as follows.

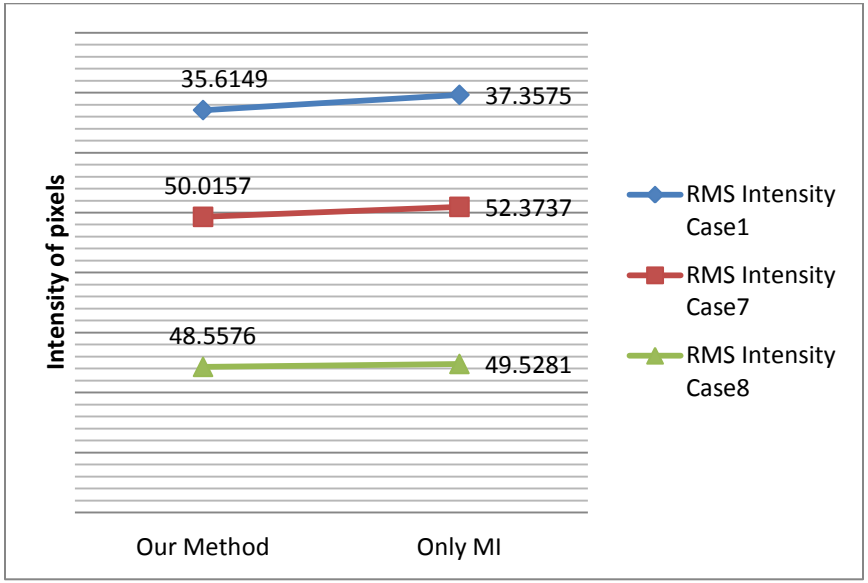
Cases	Case1	Case1_MI	Case 7	Case 7_MI	Case 8	Case 8_MI
Weights	0.5 0.5 0.5 0.5	- - - -	0.5 0.5 0.5 0.5	- - - -	0.1 0.1 0 0	- - - -
Number of iteration Coarse Fine	401 78	291 41	500 451	500 234	500 500	500 346
Time(sec) Coarse Fine	16908 3116	761 95	17128 13834	1074 446	18672 15109	1212 608
Min Jacobian	0.336578	0.435365	0.260167	0.309173	-0.222756	-0.209695
Max Jacobian	1.72964	1.55007	2.30337	2.14641	2.54068	2.66831
Percentage Jac >1 Jac<1	61% 38%	68% 31%	60% 39%	60% 39%	55% 44%	57% 42%
NMI	0.608383	0.582899	0.555335	0.524483	0.482285	0.475587
Edge Overlap	0.0742535	0.0788681	0.0750648	0.0771207	0.088021	0.0904541
RMS Intensity Difference	35.6149	37.3575	50.0157	52.3737	48.5576	49.5281

Table 4-7. comparison of case1, case 7 and 8 of clinical data by using our registration method and only-MI method for each case.

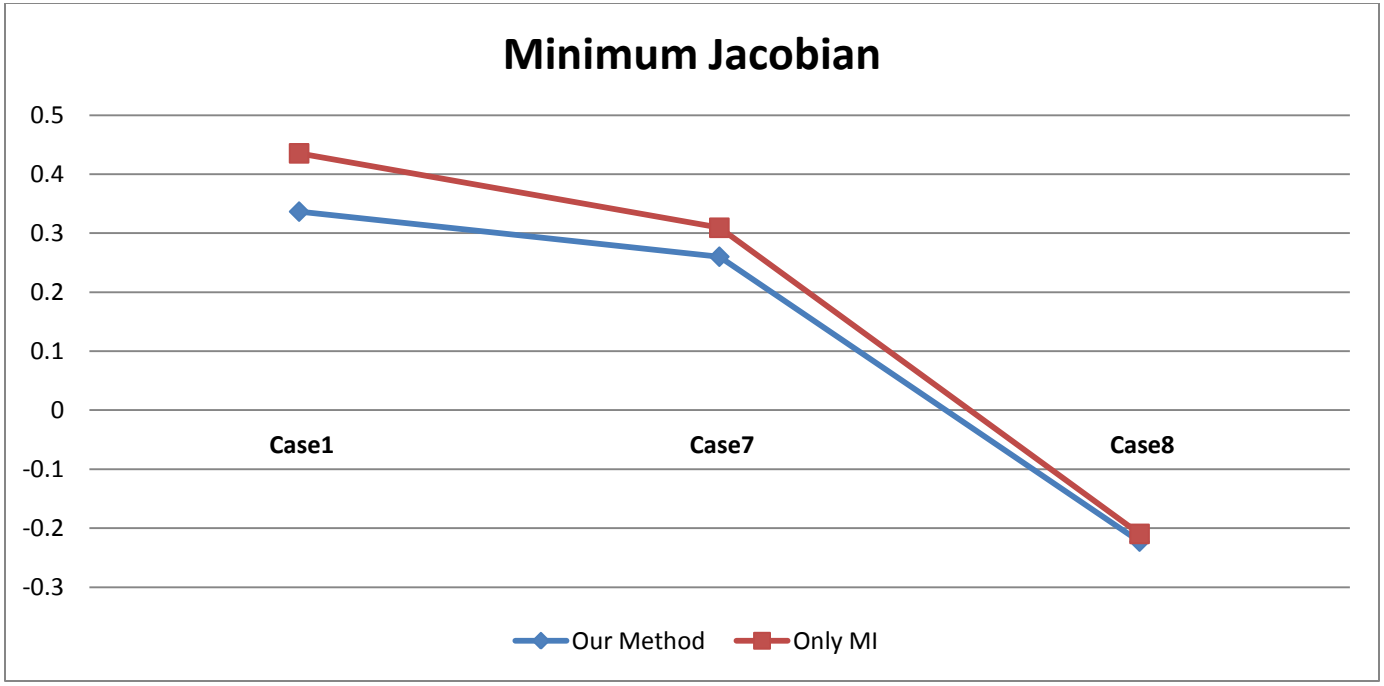
As completely demonstrated in chapter 3, for case 1 and case 7 the best weights we chose is [0.5 0.5 0.5 0.5] utilizing the regulations reported in part 1 of this chapter. But for case 8 if we set [0.5 0.5 0.5 0.5], the final DF is not acceptable and the jacobian values go to -12. (Chapter 3, part 3-4-2, table 3-8). Therefore for case 8 the best weight is [0.1 0.1 0 0]. Also for case 8 minimum jacobian has gone to a negative value for both our registration and only-MI, because of the large deformation between full inhale and full exhale.

The following graphs depict normalized mutual information (NMI), RMS intensity difference between the fixed and warped image, as well as the minimum jacobian values.





Graph 4-4. Normalized mutual information (NMI) and RMS of intensity differences between the fixed and warped images are compared between our registration method and only-MI method for each cases 1, 7 and 8.



Graph 4-5. Minimum jacobian values comparing our method and only-MI method for each cases 1,7 and 8.

In comparison to only-MI and taking all the three cases into account, NMI increase between the fixed and warped image is 4%, RMS intensity differences improvement also 4% and comparable jacobian value.

In the following image 4-2, warped images are shown with final DF overlaid on them. First image (left) shows warped image obtained by only-MI registration, the middle image includes the warped image obtained by our registration case1_6 and the right image shows case1_1 with the weights [1 1 1 1] and it has negative jacobian values. The most significant discrimination between the images are bordered by a red rectangle. Comparing only-MI and case1_6 (left and middle) we see the shorter length of the vectors in our registration than the MI method. In the right image, the DF is more chaotic, which proves the worse alignment and negative jacobian of the DF.

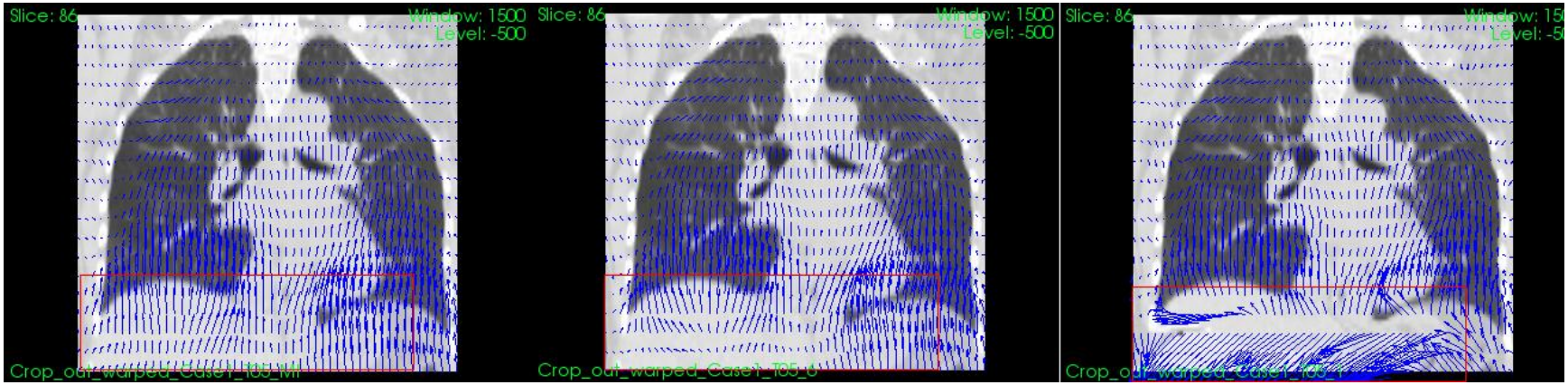


Figure 4-2. Comparison of only-MI (Left), Case1_6 of our method (Middle) and Case1_1(Right) which has the inappropriate weights assigned and has the worse deformation.

4-3-Discussion on Inverse Consistency

The point investigated here is whether the method to choose weights affects the consistency in the direct and inverse registration or not.
In the following table 4-8, we have presented the same table shown in chapter three which consists of case1, 7 and 8 to evaluate the inverse consistency of each case. However for case1, we have assessed three weights to ensure that the cases with smoother final DF have better consistency on reverse registration.

Cases	Case1_MI	Case1_2	Case1_3	Case1_6	Case7	Case8
Weights	- - - -	0.1 0.1 0 0	0 0 1 1	0.5 0.5 0.5 0.5	0.1 0.1 0 0	0.1 0.1 0 0
RMS displacement field Difference of Direct and Reverse registration DF (mm)	3.21869	2.97728	2.88039	2.87151	2.60121	3.5601
Robust Max displacement field difference (mm)	6.25829	6.31834	5.96702	6.02053	4.37401	7.41414

Table 4-8. Evaluation of inverse consistency for clinical cases 1, 7 and 8, compared with only-MI.

In the above table, case 8 which has the most deformation among the cases obviously has the worst differences between the direct and reverse registration comparing the DFs. Among the cases chosen for case1, as before case1_6 has the smallest differences in direct and reverse DFs. In comparison to only-MI registration method, case1_6 has 12% better outcome for reverse registration in terms of differences of direct-inverse registration.

In the following figure 4-3, the comparison of reverse registration of case1_6 and only-MI can be seen. The most obvious change in comparison to the direct registration is the direct of the vector fields which are reversed here. Here also we have overlaid the final DF obtained by each registration on the warped image to see the amount of deformation on each

case. The most discriminative part is shown by a red rectangle. In the left image of case1_6, the lengths of the vectors are much shorter than the only-MI case.

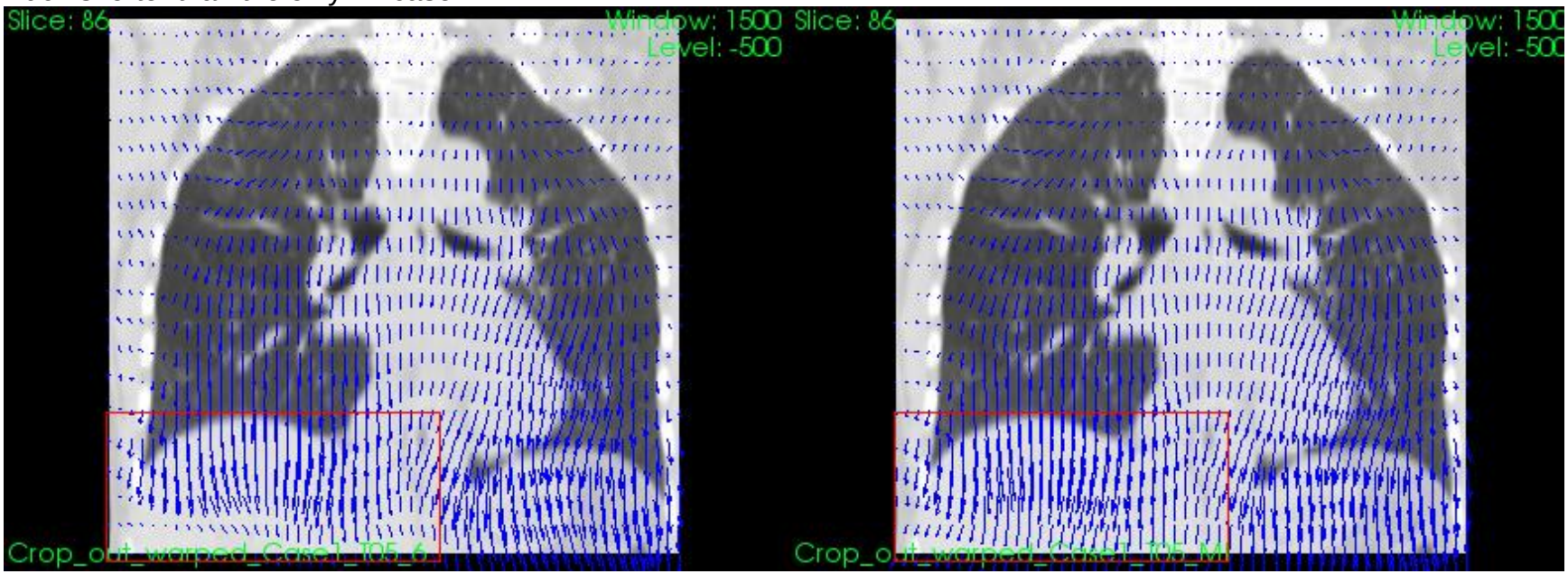


Figure 4-3. Difference of reverse registration of Left: case1_6 using our method and Right: case 1 applied only-MI registration.

4-4-Summary and conclusion

The main aim of this thesis is to supply a framework in order to design and implement an image registration method which contains vector calculus weighted cost function using ITK libraries. The framework has been tested to empirically obtain the optimal registration parameters for parametric registration algorithms applied to adaptive radiotherapy. The optimization process considers both the similarity of the images and the anatomical consistency of the transformation, which is required for radiotherapy applications.

In this thesis we have focused on implementing and designing an application of deformable image registration for adaptive radiotherapy. For this purpose, we implemented a multi-resolution image registration application with a multi-metric framework containing mutual information similarity metric and a vector spline regularization term based on divergence and curl of the deformation field. The multi-resolution technique is based on BSpline grid points, starting with a coarse grid to cover the global dissimilarities and moving to a fine to compensate local dissimilarities in the registration approach.

The discriminating spot of the application is the cost function used in the registration process. By using divergence and curl operators, we can compensate the irregularities and non-smoothness in deformable registration. Mutual information is used for recovering the alignment between fixed and moving image during the registration procedure.

In general, registration techniques that do not uniquely determine the correspondence between image volumes should benefit from the consistency constraint. This is because such techniques often rely on minimize/maximize a similarity measure which has a large number of local minima/maxima due to the correspondence ambiguity.

The problem with many image registration techniques is that the image similarity function does not uniquely determine the correspondence between two image volumes. In general, cost functions which utilize only-similarity metric without any regularization function have many local minima due to the complexity of the images being matched and the dimensionality of the transformation. It is these local minima (ambiguities) that cause the estimated transformation from image to be different from the inverse of the estimated transformation and this becomes more of a problem as the dimensionality of the transformation increases.

The major innovative point of our method is to implement the integrated divergence and curl of the deformation field into the cost function along with the mutual information as the similarity metric. Moreover we have defined weights the user can assign to the div-curl parameters before performing the registration process. Two weights are assigned for the coarse grid and two others are assigned for fine grid deformable stage, allowing to maintain the regularization terms of the cost function and obtain the desired deformation field.

For the evaluation, we used two datasets comprising synthetically deformed phantom images and real clinical patient images. The first one comes from a radio-equivalent head and neck phantom. These images are synthetically deformed and thus the synthetic transformation is considered the gold standard. Another dataset we employed includes three

patient cases, consisting of 4DCT images of the lung provided with medical reviewed landmarks, to be used as a robust benchmark to evaluate our application.

Overall we tested up to 80 cases using design of experiments method based on observational sensitivity assessment by regulating the optimal weights of div/curl to obtain the best alignment and a smooth deformation field. For the clinical cases we also justified inverse consistency of the transformation and clinical features landmark errors.

In order to evaluate our application, we used five different metrics of evaluation such as MI, RMS intensity differences, edge overlap between the fixed and warped images, RMS displacement field differences and jacobian determinant.
The statistical results show that for the synthetic images dataset comparing to only-mutual information technique we have 4.8% improvements in term of both image similarities metrics and final deformation field irregularities. We also reached 16% better recovery comparing to only-sum of squared difference (SSD) similarity metric. For the clinical patient dataset in comparison to only-MI registration method we have obtained 4% better results. Likewise, our method has 12% better consistency for inverse registration in comparison with only-MI registration.

4-5-Future work

The future work considers the following aspects:

1- In adaptive radiotherapy, a platform in which deformable image registration is optimized for the treatment is lacking. In commercial software, parameters are fixed based on previous empirical experiments, thus not guaranteeing the convergence to an optimum. The future work aims at supplying a mathematical framework to obtain optimal registration parameters for clinical applications by exploiting our implemented method. The optimization process will consider both the image similarity and anatomical consistency of the transformation. Deformable registration has several potential applications in the field of radiotherapy. However, the presence of non-optimal algorithms which only rely on fixed parameters limits the actual clinical applicability of deformable registration algorithms. Therefore the development of a platform in which the user is given the ability to select multiple transformations and registration algorithms would extensively facilitate deformable registration application at any stage of the radiotherapy treatment. This platform may also be used to estimate what are the optimal registration parameters depending on the types of chosen images, algorithms and modalities in a robust mathematical framework.

2-Since the implementation done in this thesis completely relies on open source libraries and other non-commercial academic applications, code optimization is needed. Consideration of a work plan and a group team of experts to evaluate the complexity and sensitivity of the algorithm will help the application to improve its performance and work better in terms of the goals and objectives of the adaptive radiotherapy platform.

3- As mentioned in chapter 2 of this thesis, there are several research communities which promote standardized evaluation framework for assessing the performance of an application of deformable image registration. In the future considering such a standard world wide database benchmarks to evaluate our registration results can help us to recover weakness of our program and reach to a reliable, robust and consistent registration algorithm which is useful in an adaptive radiotherapy framework.

Bibliography

1- Michael R. Kaus, Kristy K. Brock, Deformable image registration for radiation therapy planning: Algorithms and applications, BIOMECHANICAL SYSTEMS TECHNOLOGY –Vol. 1, World Scientific Publishing Co. Pte. Ltd, ISBN:978-981-270-981-3, 981-270-981-9, Chapter 1, Dec 2007.

2- Ming Chao, Yaoqin Xie and Lei Xing, Auto-propagation of contours for adaptive prostate radiation therapy, Physics in Medicine and Biology, Vo. 53, No. 17, Aug 2008.

3- Damion Shelton, Insight Segmentation and Registration Toolkit (ITK), Methods in Image Analysis, CMU Robotics Institute 16-725, U. Pitt Bioengineering 2630, Spring Term 2006.

4- Stefan Klein, Marius Staring, and Josien P. W. Pluim, Evaluation of Optimization Methods for Nonrigid Medical Image Registration Using Mutual Information and B-Splines, IEEE Transaction On Image Processing, Vol. 16, No. 12, Dec 2007.

5- Olivier Clatz, Fundamentals of medical imaging registration, Lecture 1 & 2, INRIA institute of technology, from http://www.clatz.com/download/presentations/.

6- David Sarrut, Deformable Registration for Image-Guided Radiation Therapy, Z Med Phys, Zeitschrift fur medizinische Physik (2006), Vol. 16, Issue: 4, Pages: 285-297.

7- W. Peckar, C. Schnorr, K. Rohr, and H.S. Stiehl. Non-rigid image registration using a parameter-free elastic model. In BMVC, 1998.

8- Thomas Guerrero, Geoffrey Zhang, Tzung-Chi Huang, Kang-Ping Lin, Intrathoracic tumour motion estimation from CT imaging using the 3D optical flow method, Physics in Medicine and Biology, Vol. 49, No. 17, Aug 2004.

9- D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, Nonrigid registration using free-form deformations: Application to breast MR images, IEEE Transaction on Medical Imaging, Vol. 18, No. 8, Aug 1999.

10- Paul Viola and William M.Wells, Alignment by Maximization of Mutual Information, International Journal of Computer Vision, 24(2) pg 137–154, 1997.

11- G. E. Christensen and H. J. Johnson, Consistent Image Registration, IEEE Transaction on Medical Imaging, Vol. 20, Issue. 7, 568 – 582, Aug 2002.

12- G. E. Christensen and J. He, Consistent Nonlinear Elastic Image Registration, IEEE Transaction on Mathematical Methods in Biomedical Image Analysis, 37 – 43, Aug 2002.

13- Torsten Rohlfing, Calvin R. Maurer, Jr., David A. Bluemke, Michael A. Jacobs, Volume-Preserving Nonrigid Registration of MR Breast Images Using Free-Form Deformation With an Incompressibility Constraint, IEEE Transaction on Medical Imaging, Vol. 22, Issue. 6, 730 – 741, July 2003.

14- Ch. Venkateswara Rao, K. M. M. Rao, A. S. Manjunath, and R. V. N. Srinivas. Optimization of automatic image registration algorithms and characterization, Proceedings of the ISPRS Congress, pages 698–702, Istanbul (Turkey), July 2004.

15- Hadi Rezaei, Sassan Azadi, Nonrigid Medical Image Registration using Hierarchical Particle Swarm Optimization, IEEE Transaction on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, ISBN: 978-1-4244-3429-9, 1-4, Sep 2009.

16- Arganda-Carreras, C. Ó. Sánchez Sorzano, R. Marabini, J. M. Carazo, C. Ortiz-de Solorzano, and J. Kybic, Consistent and elastic registration of histological sections using vector-spline regularization, in Computer Vision Approaches to Medical Image Analysis, ser. Lecture Notes in Computer Science, Vol. 4241. Springer Berlin / Heidelberg, May 2006, pp. 85-95.

17- Mei-Yi Chu, Hua-Mei Chen, Chih-Yao Hsieh, Ting-Hung Lin, Hsi-Yue Hsiao, Guojun Liao and Qi Peng, Adaptive Grid Generation Based Non-rigid Image Registration using Mutual Information for Breast MRI, Journal of signal processing systems, Vol. 54, No. 1-3, 45-63.

18- Akinlar, Mehmet Ali, A new method for non-rigid registration of 3D images, PhD Dissertation, University of Texas Arlington, DISS-10194, Jan 2009.

19- by Dan Ruan, S Esedoglu, J A Fessler, Discriminative sliding preserving regularization in medical image registration, IEEE International Symposium on Biomedical Imaging, Volume: 129, ISBN: 9781424439317, 430-433, 2009.

20- Martin Urschler, Stefan Kluckner and Horst Bischof, A Framework for Comparison and Evaluation of Nonlinear Intra-Subject Image Registration Algorithms, Institute of Computer Graphics and Vision, Graz, University of Technology, Austria, Insight Journal, 2008.

21- P. Hellier, C. Barillot, I. orouge, B. Gibaud, G. Le Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen, and H. J. Johnson, Retrospective Evaluation of Intersubject Brain Registration, IEEE Transaction on Medical Imaging, Vol. 22, Issue. 9, 1120 - 1130, Sep 2003.

22- Arno Klein et al, Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration, Elsevier Science Direct Publication, NeuroImage, Vol. 46, Issue 3, pp 786-802, July 2009.

23- Wei, Ying. Non-rigid image registration evaluation using common evaluation databases, Thesis-Master of Science, University of Iowa, 2009. From http://ir.uiowa.edu/etd/448.

24- Jay West, J. Michael Fitzpatrick, et al. Comparison and evaluation of retrospective inter-modality brain image registration techniques, ournal of Computer Assisted Tomography, Vol 21, Issue 4, pp 554-568, Aug 1997.

25- R. Castillo, E. Castillo, R. Guerra, V.E. Johnson, T. McPhail, A.K. Garg, T Guerrero. A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. Physics in Medicine and Biology Vol. 54 No 7, pp 1849-1870, 2009.

26- Ibanez L., Schroeder W., Ng L., Cates J., Consortium T.I.S., Hamming R., The ITK Software Guide, Kitware, Inc., Jan-2003.

29- Se Young Chun and Jeffrey A. Fessler, Regularized methods for topology-preserving smooth nonrigid image registration using B-spline basis, 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBN: 978-1-4244-2002-5, pp 1099 –1102, May 2008.

30- John Ashburner, A fast diffeomorphic image registration algorithm, Elsevier Science Direct Publication, NeuroImage, Vol. 38, Issue 1, pp 95-113, Oct 2007.

31- M. F. Spadea, P. Zaffino, M. Peroni, M. Riboldi, G. Sharp, G. Baroni, Quantitative evaluation of a deformable registration toolkit, Proceedings of the XVIth ICCR.

32- Marius Staring, Regularization in Deformable Registration, 13th International Conference on Medical Image Computing and Computer Assisted Intervention, MACCI Tutorial, Sep 2010.

33- G. C. Sharp, R. Li, J. Wolfgang, G. TY. Chen, M. Peroni, M. F. Spadea, S. Mori, J. Zhang, J. Shackleford, N. Kandasamy, Plastimatch- An open source software suite for radiotherapy image processing, Proceedings of the XVIth ICCR.