B-spline Non Rigid Image Registration using L-BFGS Optimizer

B.Sri harsha1, PN.Pournami2, V K Govindan3

1,2 M.Tech, Department of Computer Science and Engineering, NIT Calicut, kerala, India
3Ph.D, Department of Computer Science and Engineering, NIT Calicut, kerala, India

Abstract- Image registration is a vital problem in medical imaging. It has many potential applications in clinical diagnosis (Diagnosis of cardiac, retinal, pelvic, renal, abdomen, liver, tissue etc.). It is a process of aligning them in order to monitor subtle changes between the two. There are lots of image registration techniques evolved for sooothing the image registration process. This paper proposes a B-spline non rigid image registration method using L-BFGS Optimizer.

Keywords: Non-rigid image registration, B-spline transformation, Optimizer.

I. INTRODUCTION

The frequent problem arises when images taken at different times, by different sensors or from different viewpoints need to be compared. The image needs to be aligned with one another so that differences can be detected. A similar problem occurs when searching for a prototype or template in another image. To find the optimal match for the template in the image, the proper alignment between the image and template must be found.

Image registration [1] is a process of aligning two images into a common coordinate system thus aligning them in order to monitor subtle changes between the two. Registration algorithms compute the transformations to set correspondence between the two images. Image processing methods are possibly able to visualize objects inside the human body, are of special interest. Advances in computer science have led to reliable and efficient image processing methods useful in medical diagnosis, treatment planning and medical research. In clinical diagnosis, the integration of useful data obtained from separate images using medical images is often desired. The images need to be geometrically aligned for the better observation. This procedure of mapping points from one image to corresponding points in another image is called Image Registration. In general, its applications can be divided into three main groups according to the manner of the image acquisition:

1. Different viewpoints (multi view analysis): Images of the same scene are acquired from different viewpoints. The aim is to gain larger a 2D view or a 3D representation of the scanned scene. Examples of applications: Remote sensing, mosaicing of images of the surveyed area, computer vision shape recovery.
2. Different times (multi temporal analysis): Images of the same scene are acquired at different times, often on regular basis, and possibly under different conditions. The aim is to find and evaluate changes in the scene which appeared between the consecutive images acquisition.

Examples of applications: Remote sensing monitoring of global land usage, land-scape planning, computer vision automatic change detection for security monitoring.
3. Different sensors (multimodal analysis): Images of the same scene are acquired by different sensors. The aim is to integrate the information obtained from different source streams to gain more complex and detailed scene representation. Example of application: Remote sensing.

The majority of the registration methods consist of the following four steps:

1. Feature Detection: Salient and distinctive objects (closed-boundary regions, edges, contours, line intersections, corners, etc.) are manually or, preferably, automatically detected. For further processing, these features can be represented by their point representatives (centers of gravity, line endings, distinctive points), which are called Control Points (CPs) in the literature.

2. Feature Matching: In this step, the correspondence between the features detected in the sensed image and those detected in the reference image is established. Various feature descriptors and similarity measures along with the spatial relationships among the features are used for that purpose.

3. Transform Model Estimation: The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated. The parameters of the mapping functions are computed by means of the established feature correspondence.

4. Image Resampling and Transformation: The sensed image is transformed by means of the mapping functions. Image values in non-integer coordinates are computed by the appropriate interpolation technique.

Non-rigid medical image registration refers to seek one or series of space transformation for a picture of non-rigid medical image to make it match to another picture of non-rigid medical image on the corresponding points of the space. Any medical image registration can be divided into three parts

First: Determine the space transformation of the source image and the target image.
Second: Measuring the similarity degree of source images and target image.
Third: Take some measures to make the similarity measure reach the optimal value (parameter optimization) better and fast.

www.ijcsit.com
This paper is organized into seven sections. Section 1 presents an introductory discussion on Image Registration. Section 2 introduces Mutual information, one of the similarity criteria. Section 3 is about different transformations that we are going to use in this paper. Section 4 gives a brief introduction about the Line search optimization, emphasizing more on LM-BFGS. The proposed method is described in section 5. Section 6 presents the Results and finally section 7 concludes the paper.

2. MUTUAL INFORMATION

Mutual information [2] is a basic concept from information theory, measuring the statistical dependence between two random variables or the amount of information that one variable contains about the other. The MI registration criterion states that the Mutual information of the image intensity values of corresponding voxel pairs is maximal if the images are geometrically aligned. Because no limiting constraints are imposed on the nature of the relation between the intensities in the images to be registered and no assumptions are made regarding the image content of the modalities [3], the mutual information criterion is very general and powerful. So it is used widely for multi-modal image registration techniques.

\[ I(A, B) = H(A) + H(B) - H(A, B) \]

I(A, B); The amount of information that B contains about A. H(A) and H(B) being the entropy of A and B respectively. H(A, B) is the joint entropy of A and B.

Studholmes [4] has shown that with increase in misregistration (which really coincides with the decreasing overlap) the mutual information measure may actually increase. This can occur when the relative areas of object and background even out and the sum of the marginal entropies increases, faster than the joint entropy. Studholmes proposed a normalized measure of mutual information, which is less sensitive to change in overlap.

\[ C_{\text{similarity}}(A, B) = \frac{H(A) + H(B)}{H(A, B)} \]

3. TRANSFORMATION

In medical image registration accuracy of image registration is very crucial. So in this proposed method we do both Global and Local transformation [5].

\[ T(x, y, z) = T_{\text{global}}(x, y, z) + T_{\text{local}}(x, y, z) \]

Global transformation model:

It describes the overall motion of the object. The simplest choice is a rigid transformation which is parameterized by 6 degrees of freedom, describing the rotations and translations of the object. A more general class of transformations are affine transformation, describing translation, rotation, scaling and shearing.

In 3-D, an affine transformation can be written as

\[ T_{\text{global}}(x, y, z) = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & x \\ \theta_{21} & \theta_{22} & \theta_{23} & y \\ \theta_{31} & \theta_{32} & \theta_{33} & z \end{bmatrix} + \begin{bmatrix} \theta_{41} \\ \theta_{42} \\ \theta_{43} \end{bmatrix} \]

Where the coefficients parameterize the 12 degrees of freedom of the transformation (Rotation, translation, shearing and scaling).

Local transformation model:

As the Global transformation model can’t deal with deformation of the image to be registered, we go for this transformation model.

To correct the deformation in the image we go for FFD model using B-spline [6], which is powerful tool for modeling 3-D deformable objects. The basic idea of FFD is to deform an object by manipulating an underlying mesh of control points.

In Spline based FFD, we denote the domain of the image volume as \( \Omega = \{(x, y, z) | 0 \leq x \leq X, 0 \leq y \leq Y, 0 \leq z \leq Z\} \). Let \( \Theta_{i,j,k} \) with uniform spacing \( \delta \). The FFD can be written as the 3-D tensor product of the familiar 1-D cubic B-splines.

\[ T_{\text{local}}(x, y, z) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} B_i(u) B_j(v) B_k(w) \Theta_{i,j,k} \]

Where \( i = \text{floor}(\frac{x}{n_x}) - 1 \), \( j = \text{floor}(\frac{y}{n_y}) - 1 \), \( k = \text{floor}(\frac{z}{n_z}) - 1 \).

\[ u = \frac{x}{n_x - \text{floor}(\frac{x}{n_x})}, \quad v = \frac{y}{n_y - \text{floor}(\frac{y}{n_y})}, \quad w = \frac{z}{n_z - \text{floor}(\frac{z}{n_z})} \]

where \( B_i \) represents the \( i \)th basis function of the B-spline.

\[ B_0(u) = (1 - u)^3 / 6 \]

\[ B_1(u) = (3u^3 - 6u^2 + 4) / 6 \]

\[ B_2(u) = (-3u^3 + 6u^2 - 3u + 1) / 6 \]

\[ B_3(u) = u^3 / 6 \]

In contrast to other splines methods like thin-plate splines or elastic-body splines, B-splines are locally controlled, which makes them computationally efficient even for a large number of control points. Change control point \( \Theta_{i,j,k} \) affects the transformation only in the local region of the control point.

The control points \( \Theta \) act as parameters of the B-spline Free Form Deformation degree of non-rigid deformation depends on the resolution of the mesh of control points. A large spacing of control points allows modeling of global non-rigid deformation, while a small spacing of control points allows modeling of highly local non-rigid deformation. At the same time, the resolution of the control point mesh defines the number of degrees of freedom and, consequently, the computational complexity. In general, the local deformation of the objects should be characterized by a smooth transformation. To constrain the spline-based FFD transformation to be smooth, one can introduce a penalty term which regularizes the transformation. The general form of such a penalty term has been described by Wahba [7].
4. LINE SEARCH OPTIMIZATION:
The line search strategy is to find a local minimum of an objective function \( f \). The line search approach first finds a descent direction along which the objective function \( f \) that will be reduced and then computes a step size that determines how far \( f \) should move along that direction. The descent direction can be computed by various methods, such as Gradient descent, Newton’s method and Quasi-Newton method.

Quasi-Newton methods are algorithms for finding local maxima and minima of functions. Quasi-Newton methods are based on Newton method to find the stationary point of a function, where the gradient is 0. Newton’s method uses the gradient and the Hessian matrix of second derivatives of the function to be minimized. In quasi-Newton methods the Hessian matrix does not need to be computed. The Hessian is updated by analyzing successive gradient vectors instead.

Limited-memory BFGS [8] (L-BFGS or LM-BFGS) is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm using a limited amount of computer memory, conventional BFGS stores a dense \( n \) approximation to the inverse Hessian \( (n \) being the number of variables in the problem). Unlike L-BFGS stores only a few vectors that represent the approximation implicitly. Due to its resulting linear memory requirement, the L-BFGS method is particularly well suited for optimization problems with a large number of variables.

**Basic optimization algorithm**
Input: Initial \( x_0 \) value, \( f \) (function), \( k=0 \)
Stopping criterion: \( \frac{df}{dx} = 0 \)
Output: \( x_0 \) (is the value at which the function is a minimum)

Algorithm:
1. Compute a search direction \( p_k \).
2. Compute the step length \( \alpha_k \), such that \( f(x_k + \alpha_k p_k) < f(x_k) \).
3. Update \( x_k = x_k + \alpha_k p_k \).
4. Check for convergence (Stopping criteria)

The above mentioned algorithm is the more generalized optimization algorithm.

BFGS algorithm is an iterative method for solving unconstrained nonlinear optimization problems. It approximates Newton’s method, a class of hill-climbing optimization technique that seeks a stationary point of a function. A necessary condition for optimality is that the gradient be zero.

**Broyden-Fletcher-Goldfarb-Shanno Algorithm:**
Input: Initial value - \( x_0 \), \( \delta \) - error, \( c_{ij} \) - initial hessian.
Stopping criterion: norm of gradient greater than zero.
Output: \( x_j \) (the value at which function is a minimum)

Algorithm:
1. \( j=0 \)
2. while (true) do
   a. \( d_j \leftarrow -c_j \Delta f(x_j) \)
   b. \( a_j \leftarrow \text{LineSearch}(x_j, f) \)
   c. \( x_{j+1} \leftarrow x_j + a_j d_j \)
   d. Compute \( c_{j+1} \)
   e. \( j=j+1 \)
   f. if \( \|\Delta f(x_j)\| \leq \delta \) then
      stop
   g. end if
3. end while

BFGS is currently considered the most effective and is by far the most popular quasi-Newton update formula. The success of the BFGS algorithm depends on how well the updating formula for \( C_j \) approximates the inverse of the true Hessian at the current iterate. Experiments have shown that the method has very strong self-correcting properties (when the right line search is used) so that if, at some iteration, the matrix contains bad curvature information, it often takes only a few updates to correct these inaccuracies. BFGS requires only matrix-vector multiplications which brings the computational cost at each iteration from \( O(n^2) \) for Newton’s method down to \( O(n^2) \). However, if the number of variables is very large, even \( O(n^2) \) per iteration is too expensive - both in terms of CPU time and sometimes also in terms of memory usage (a large matrix must be kept in memory at all times).

A less computationally intensive method when \( n \) is large, is the Limited-Memory BFGS method. Instead of updating and storing the entire approximated inverse Hessian \( C_j \), the LBFGS method never explicitly forms or stores this matrix. Instead it stores information from the past ‘\( m \)’ iterations and uses only this information to implicitly do operations requiring the inverse Hessian (in particular computing the next search direction). The updating in LBFGS is done using just 4mn multiplications bringing the computational cost down to \( O(mn) \) per iteration. If \( m \ll n \), this is effectively the same as \( O(n) \). In some cases the LBFGS method uses as many or even fewer function evaluations to find the minimizer. This is remarkable considering that even when using the same number of function evaluations, LBFGS runs significantly faster than full BFGS if \( n \) is large.

**Direction finding in LBFGS**
\( q \) - The present gradient
\( y \) - The gradient vector, which contains all the previous gradient values.
\( v \) - The list of x values calculated in the process of getting optimum.
\( r \) = \( \text{Transpose of } y \times \text{Transpose of } v \)
\( s \) = \( \text{Transpose of } v \).
Input: q (gradient), p = \frac{1}{\text{Transpose of } y \times v}, s = \text{Transpose of } v \times y \times r \.
Output: Direction d (\( d \rightarrow -r \))

Algorithm:
1. for \( i=(j-1): -1 : (j-m) \) do
2. \( a_i \leftarrow p_i (s^i)^T q \)
3. \( q \leftarrow q - a_i y^{(i)} \)
4. end for
5. for \( i=(j-m): 1 : (j-1) \) do
6. \( \beta \leftarrow p_i (s^i)^T r \)
7. \( r \leftarrow r + (s^i)(a_i - \beta) \)
8. end for

www.ijcsit.com
5. PROPOSED METHOD:

B-spline is more flexible transformation having high degrees of freedom. L-BFGS is an optimization algorithm, calculates it’s parameters from the previous instances with linear memory storage. By combining these two, we can get an efficient registration technique.

Algorithm:
**Input:** Target image (I1) and Reference image (I2)
**Output:** Registered image (I3)

1. Take the initial transformation matrix $\theta_0$.
2. Compute ‘$\theta_1$’ by using Global transformation.
   i.e $\theta_1 = \text{GlobalTransformation}(\theta_0)$
3. Initialize the control points $\emptyset$.
4. for i = $\emptyset$ to $\emptyset^o$
   a. Compute ‘$\theta_2$’ by using local transformation.
      i.e $\theta_2 = \text{LM-BFGS}(f, \theta_1\_opt)$
   b. Increase the control point resolution by calculating the new control points
5. Apply B-spline transformation on the target image using $\theta_2$ to obtain I3.

We can use any one of the global transformations like rigid, affine. Opt is a structure that contains information like error ($\delta$), termination tolerance of transformation matrix. Function f contains the parameters like size of control points, I1 and I2.

6. RESULTS:

Figure 1 & Figure 2 are Moving and Static image respectively. Figure 3 is the final registered image. Table1 shows the comparison results of the proposed method with affine transformation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Image difference</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affine</td>
<td>20.08</td>
<td>9.92</td>
</tr>
<tr>
<td>B-spline with L-BFGS</td>
<td>1.16</td>
<td>2.27</td>
</tr>
</tbody>
</table>

7. CONCLUSION

In this paper, we are using B-spline transformation on the image so that any deformations present in the image can be rectified. In order to perform the registration process efficiently using B-spline, we must choose an optimization process so that the registration is done efficiently. For this purpose, we have chosen L-BFGS optimizer. We have shown in section 4 how this gives better performance over the other optimizers. With results obtained, we can say that proposed method is efficient.

REFERENCES