Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2017-0141

Vol. **23**, No. 3, pp. 877-898
March 2018

COMPUTATIONAL SOFTWARE

DTHB3D_Reg: Dynamic Truncated Hierarchical B-Spline Based 3D Nonrigid Image Registration

Aishwarya Pawar[1], Yongjie Jessica Zhang[1,*], Cosmin Anitescu[2], Yue Jia[3] and Timon Rabczuk[2]

[1] *Department of Mechanical Engineering, Carnegie Mellon University, PA 15213, United States.*
[2] *Institute of Structural Mechanics, Bauhaus-Universität Weimar, Germany.*
[3] *School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, China.*

Received 29 June 2017; Accepted (in revised version) 12 August 2017

Abstract. We present a robust approach to perform 3D nonrigid image registration suitable for large deformation and topology change, and develop a software package named DTHB3D_Reg (Dynamic Truncated Hierarchical B-spline based 3D Image Registration). The optimum spatial transformation, defined using truncated hierarchical B-splines, is obtained through the minimization of an energy functional. The optimization process minimizes sum of squared difference in the intensity values of the grayscale images. Control points are dynamically updated without constructing large matrices as in finite element method. To improve the computational efficiency, an adaptive strategy carries out refinement only in the regions with large deformation. The proposed method is demonstrated on 3D synthetic and medical images to show robustness on topology change as compared to other image registration methods.

AMS subject classifications: 65K10, 65K05

Key words: 3D nonrigid image registration, dynamic scheme, truncated hierarchical B-spline, adaptive refinement, topology change.

Program summary

Program title: DTHB3D_Reg
Nature of problem: This package performs 3D non-rigid image registration for medical and synthetic images using truncated hierarchical B-splines (THB-Splines).

*Corresponding author. *Email addresses:* jessicaz@andrew.cmu.edu (Y. Zhang), arpawar@andrew.cmu.edu (A. Pawar), cosmin.anitescu@uni-weimar.de (C. Anitescu), yuejia@nwpu.edu.cn (Y. Jia), timon.rabczuk@uni-weimar.de (T. Rabczuk)

Software licence: BSD 3-Clause License

CiCP scientific software URL: `https://github.com/arpawar/DTHB3D_Reg`

Programming language(s): Matlab and C++.

Computer platform: x86-64.

Operating system: Linux, Windows and Mac OS X.

Compilers: Supported and compatible compilers for MATLAB 2017a and previous releases:

- `http://www.mathworks.com/support/compilers`
- `http://www.mathworks.com/support/sysreq/previous_releases.html`

RAM: 16 GB and higher recommended.

External routines/libraries: None

Running time: Running time will depend upon the size of the images and the computational resources allocated to run the software.

Restrictions:

Supplementary material and references:

Additional comments:

1 Introduction

Image registration is the process of computing optimum correspondence between two images through a spatial transformation mapping. One of its most prominent applications is the alignment of medical images, where spatial correspondence is obtained between salient features of different images. These images can be obtained at different time frames, subjects or imaging modalities. Image registration plays a crucial role in obtaining a more comprehensive knowledge from separate images, thus having potential applications in aiding medical diagnosis [16, 19, 24, 31].

Image registration methods can be classified based on whether the images are obtained from the same or different imaging modalities [22, 40]. In monomodal image registration, analysis can be carried out between images obtained for the same subject (intra-subject) or different subjects (inter-subject). In intra-subject registration, changes in certain features over time are measured. These feature changes can be computed through the alignment of preoperative and postoperative images from the same patient. Moreover, motion quantification of certain features can be achieved. This plays a significant role in studying disease progression such as tumor movement and growth [39]. In inter-subject registration, images are taken from several subjects to study the variation of features between normal subjects and patients. This registration process is more challenging in terms of increased complexity of shape and topology of the features [21, 37]. The complexity of the registration problem also increases while registering image sequences

with respect to both space and time [23]. In multimodal registration, images are obtained from different imaging modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Single-photon Emission Computed Tomography (SPECT) and ultrasound. Important information from different imaging modalities is combined to achieve comprehensive knowledge which helps in improving clinical diagnosis [7].

Image registration methods can also be classified as rigid and nonrigid. In rigid registration, only global transformations such as translation, rotation, shearing and scaling can be modeled. In contrast, nonrigid registration can model complex and localized deformations in a more flexible manner. For example, spline-based image registration methods [9,28,30,34] are suitable to carry out nonrigid deformation as they exploit the advantageous properties of B-splines in building the spatial transformation function, such as local control, smoothness and compact support. Image deformation can be measured by the difference of spatial coordinates of the voxels, or the voxel intensities [32,35]. In [9], a level set method was proposed to carry out large deformation based diffeomorphic nonrigid registration. In this method, the image is represented by B-splines and is manipulated in a composition framework via level set propagation. Constraints are imposed on the maximum deformation of B-spline control points to ensure that each transformation mapping is diffeomorphic [9, 27, 36], where topology change is not allowed during deformation.

In this paper, we evaluate the optimal spatial transformation function through the minimization of an energy functional, and develop a software package named DTHB3D_Reg (Dynamic Truncated Hierarchical B-spline based 3D Image Registration). The energy functional is minimized using an L^2-gradient flow method, and is constrained to ensure a smooth and realistic deformation. Unlike the finite element method [20], our approach solves the strong form directly [18] and dynamically updates the spatial transformation without constructing large matrices, with a focus on modeling large deformation to capture topology change. Here, diffeomorphism is not a crucial requirement [11, 27]. B-spline based local refinement [17, 26, 38] has been adopted to improve the computational efficiency and accuracy for 2D synthetic and medical images [25, 26]. In this work, we extend the approach to implement 3D image registration using trivariate truncated hierarchical B-splines (THB-splines). We support adaptive refinement and enable only certain regions to undergo fine-scale deformations, while keeping the rest of the control grid coarse. As compared to the level set method [9] and the optical flow method [36], we improve the accuracy of the registration process by achieving topology change and preserving important features. The implementation of the proposed method has been submitted as a software package to the public domain (`https://github.com/arpawar/DTHB3D_Reg`) for reference.

The rest of the paper is organized as follows. In Section 2, we describe the spatial transformation construction and review THB-splines, followed by the introduction of the registration framework in Section 3. We also provide details of software installation and parallel implementation in MATLAB/C++. In Section 4, we carry out 3D synthetic

and medical image registration and demonstrate the main contributions of the method. The paper ends with a brief conclusion and the future work in Section 5.

2 Spatial transformation based on THB-Splines

In this section we describe the construction of the spatial transformation function. Local refinement based on THB-splines is also reviewed.

2.1 Spatial transformation function

Given the source image $I_1(\mathbf{x})$ and the target image $I_2(\mathbf{x})$ at $t=0$, the image registration process computes an optimum $f(\mathbf{x},t)$ that corresponds to the best possible match between the deformed source image $I_1(f(\mathbf{x},t))$ and the target image $I_2(\mathbf{x})$ at a particular instance t. This process is also known as forward registration. $f(\mathbf{x},t)$ is defined using trivariate B-splines as

$$f(\mathbf{x},t)=\sum_{m=1}^{N_b}\mathbf{P}_m(t)\phi_m(\mathbf{x})=\sum_{k=1}^{n_3}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\mathbf{P}_{i,j,k}(t)N_{i,p}(u)N_{j,q}(v)N_{k,r}(w), \tag{2.1}$$

where $\mathbf{P}_m(t)$ are the control points associated with the trivariate basis functions $\phi_m(\mathbf{x})$, and N_b represents the total number of basis functions. The parametric domain $\mathbf{x}=(u,v,w)$. Each trivariate B-spline $\phi_m(\mathbf{x})$ is represented as a tensor product of three univariate B-splines $N_{i,p}(u)$, $N_{j,q}(v)$ and $N_{k,r}(w)$, which are defined on the knot vectors $U=\{u_1,\cdots,u_{n_1+p+1}\}$, $V=\{v_1,\cdots,v_{n_2+q+1}\}$ and $W=\{w_1,\cdots,w_{n_3+r+1}\}$ in u, v and w directions, respectively. These open knot vectors span the entire image domain. n_1, n_2 and n_3 are the number of univariate basis functions, and p, q and r are the degree of polynomials, respectively. Here we choose $p=q=r=2$ or triquadratic basis functions. We can use higher order spline basis functions if a higher continuity is required. The control grid is initialized as an identity map (or $f(\mathbf{x})=\mathbf{x}$) to represent the initial source image $I_1(\mathbf{x})$ at the beginning of the optimization process. In Eq. (2.1), $f(\mathbf{x},t)$ is described in a parametric form where the control points $\mathbf{P}_m$(t) are dynamically updated through the registration process. We want to obtain $f(\mathbf{x},t)$ such that $I_1(f(\mathbf{x},t))\approx I_2(\mathbf{x})$. The control points at the beginning of the registration process are defined using Greville Abscissae [13] in the image domain.

2.2 A Review of THB-splines

We now explain the representation of the spatial transformation function using THB-splines. Methods for adaptive refinement of splines have been introduced using PHT-splines [12], T-splines [29] and hierarchical B-splines (HB-splines) [8, 14] to enable local surface editing. Refinement for two consecutive levels using univariate B-splines is illustrated in Fig. 1. Here the parametric domain is defined as $U\in[0,8]$. The knot vectors

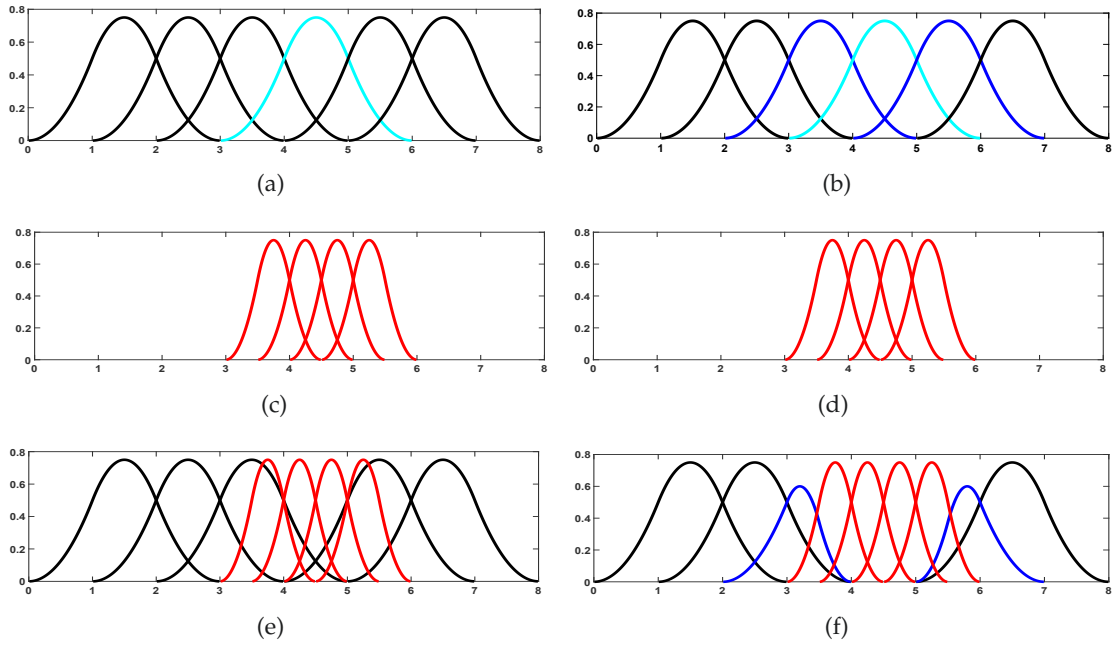


Figure 1: Construction of HB-splines versus THB-splines. First column: B-splines at coarser level (a), B-splines at finer level (c), collecting active splines to construct HB-splines (e). Second column: B-splines at coarser level (b), B-splines at finer level (d), collecting active splines to construct THB-splines (f). The active splines are shown in black, red and blue colors. The passive splines are shown in cyan color.

defined at level $l+1$ are obtained by bisecting the knot vectors at level l. The local support of the i^{th} univariate B-spline at level l ($N_i^l(u)$) is defined in the parametric space as $supp(N_i^l(u)) = [u_i, \cdots, u_{i+p+1}]$, where p is the degree of the B-spline basis function. According to the principle of local refinement, the basis function $N_i^l(u)$ at a given refinement level l (shown in cyan in Fig. 1 (a)) can be represented as a linear combination of a subset of the basis functions at the next refinement level ($l+1$) (four red curves in Fig. 1 (c)), which are termed as the children basis functions of $N_i^l(u)$. We have

$$N_i^l(u) = \sum_{k=1}^{N_c} S_{i,k} N_k^{l+1}(u), \tag{2.2}$$

where $N_k^{l+1}(u)$ are the children basis functions, and N_c is the number of the children basis functions which are completely contained in the local support of $N_i^l(u)$.

The refinement coefficients $S_{i,k}$ are computed using the Oslo knot insertion algorithm [10]. The steps explaining local refinement using HB-splines are given as follows:

1. To-be-refined B-splines are set as passive ($N_p^l(u)$); see the cyan curve in Fig. 1 (a). The remaining B-splines are set as active ($N_a^l(u)$); see the black curves in Fig. 1 (a).

2. The children of the passive B-spline are set as active ($N_a^{l+1}(u)$); see the four red

curves in Fig. 1 (c). The passive B-spline is replaced by its children using the refinability equation (2.2). The total support of the active children B-spline basis functions is denoted as $U^{l+1} \in [3,6]$.

3. The hierarchical structure is constructed by collecting all the active B-splines at levels l and $l+1$ (shown in Fig. 1 (e)). We obtain

$$N_{hb}^{l+1}(u) = N_a^l(u) \cup N_a^{l+1}(u). \tag{2.3}$$

This process is continued recursively until we reach the maximum refinement level $l_{\max}$.

THB-splines were proposed in [15] to modify the construction of HB-splines in order to satisfy partition of unity without rationalization and reduce the overlapping of B-splines between different levels. By introducing THB-splines into the spatial transformation function, we can make the image registration computation more efficient. Among the active B-splines at level l (shown in black and blue curves in Fig. 1 (b)), those having partial support from the active B-splines at level $l+1$ (the blue curves in Fig. 1 (b)) are modified using the truncation mechanism. The refinability equation Eq. (2.2) is modified to obtain

$$\text{trunc}(N_i^l(u)) = \sum_{supp(N_k^{l+1}(u)) \not\subset U^{l+1}} S_{i,k} N_k^{l+1}(u). \tag{2.4}$$

The truncated basis functions are shown in blue in Fig. 1 (f), where only the non-active children B-splines contribute to the construction of the THB-splines. The THB-splines constitute all the active B-splines at levels l and $l+1$. We obtain

$$N_{thb}^{l+1}(u) = N_a^l(u) \cup N_a^{l+1}(u). \tag{2.5}$$

The construction of trivariate THB-splines from univariate THB-splines is straightforward. Each trivariate B-spline ($\phi_m^l(\mathbf{x})$) (defined in Eq. (2.1)) at the refinement level l has local support defined as $supp(\phi_m^l(\mathbf{x})) = [u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}] \times [w_k, w_{k+r+1}]$. $\phi_m^l(\mathbf{x})$ is replaced by its non-zero children basis functions in $supp(\phi_m^l(\mathbf{x}))$.

3 Registration framework

In this section, we describe how to construct the energy functional and the optimization process. Details of refinement based on the image deformation is also discussed. Finally, the details on the implementation in MATLAB/C++ and code optimization based on parallel computation are provided.

3.1 Energy functional

The energy functional for 2D image registration was defined in [18, 20], and here we extend it to incorporate the registration in 3D space. The optimization is carried out by minimizing an energy functional $E(f(\mathbf{x},t))$ with the L^2 gradient flow method, which measures the amount of mismatch between two input images and converts the convex optimization problem into an ordinary differential equation. The goal of optimization is to iteratively decrease the dissimilarity in the intensities of the images. In 3D, the energy functional is written as

$$\begin{aligned} E(f(\mathbf{x},t)) = &\int_\Omega g(\mathbf{x},t)(I_2(\mathbf{x}) - I_1(f(\mathbf{x},t)))^2 \mathrm{d}\Omega \\ &+\lambda_1 \int_\Omega \left(\|f_{,u}(\mathbf{x},t)\|_2^2 + \|f_{,v}(\mathbf{x},t)\|_2^2 + \|f_{,w}(\mathbf{x},t)\|_2^2 \right) \mathrm{d}\Omega \\ &+\lambda_2 \int_\Omega \left(\|f_{,u}(\mathbf{x},t)\|_2^2 \|f_{,v}(\mathbf{x},t)\|_2^2 - (<f_{,u}(\mathbf{x},t), f_{,v}(\mathbf{x},t)>)^2 \right) \\ &+ \left(\|f_{,v}(\mathbf{x},t)\|_2^2 \|f_{,w}(\mathbf{x},t)\|_2^2 - (<f_{,v}(\mathbf{x},t), f_{,w}(\mathbf{x},t)>)^2 \right) \\ &+ \left(\|f_{,u}(\mathbf{x},t)\|_2^2 \|f_{,w}(\mathbf{x},t)\|_2^2 - (<f_{,u}(\mathbf{x},t), f_{,w}(\mathbf{x},t)>)^2 \right) \mathrm{d}\Omega, \end{aligned} \tag{3.1}$$

where λ_1 and λ_2 are two regularization parameters chosen according to the amount of regularization needed for the registration process. Here we set λ_1 and λ_2 to 0.0001 during the entire registration process. $f_{,u}(\mathbf{x},t)$, $f_{,v}(\mathbf{x},t)$ and $f_{,w}(\mathbf{x},t)$ are the first derivatives of $f(\mathbf{x},t)$ with respect to u, v and w, respectively. $<\cdot,\cdot>$ is the inner product operator. The term $g(\mathbf{x},t)$ is defined as

$$g(\mathbf{x},t) = \frac{1}{\sqrt{\gamma + (\frac{\partial I_1(f(\mathbf{x},t))}{\partial u})^2 + (\frac{\partial I_1(f(\mathbf{x},t))}{\partial v})^2 + (\frac{\partial I_1(f(\mathbf{x},t))}{\partial w})^2}}, \tag{3.2}$$

where γ is a small non-zero quantity set to the value of 10^{-12} to prevent division by zero. The first term in Eq. (3.1) measures the sum of squared difference (SSD) in the intensity between the target image and the evolving source image. To prevent unrealistic changes within the image while carrying out complex deformation, we introduce additional regularization terms in the energy functional. The second and third terms associated with λ_1 and λ_2 respectively, impose regularization constraints to ensure the smooth deformation of $I_1(f(\mathbf{x},t))$ and generate a continuous mapping. The second term (first-order regularization) ensures smooth variation of $f(\mathbf{x},t)$ in u, v and w directions. The third term (second-order regularization) ensures the consistency of the area of each face of the 3D control grid element during deformation.

During the minimization process, the energy functional is differentiated with respect to control points $\mathbf{P}_m(t)$ (Eq. (2.1)) and is converted to an ordinary differential equation by introducing a pseudo timestep, we have

$$\frac{\mathrm{d}\mathbf{P}_m(t)}{\mathrm{d}t} = -\delta E_m(f(\mathbf{x},t)). \tag{3.3}$$

The left hand side of the equation is approximated using the Euler method to obtain

$$\frac{\mathbf{P}_m^{s+1}-\mathbf{P}_m^s}{\epsilon}=-\delta E_m^s(f(\mathbf{x},t)), \tag{3.4}$$

where ϵ is the timestep introduced for the iterative process and the control points are dynamically updated. $\mathbf{P}_m^{s+1}$ and $\mathbf{P}_m^s$ are the control points obtained at the current and previous timestep. $\delta E_m^s(f(\mathbf{x},t))$ can be written as

$$\begin{aligned}\delta E_m^s(f(\mathbf{x},t))=&-2\int_\Omega g(I_2(\mathbf{x})-I_1(f(\mathbf{x},t)))\nabla I_1(f(\mathbf{x},t))\phi_m(\mathbf{x})\mathrm{d}\Omega\\&+\lambda_1\int_\Omega(2f_{,u}(\mathbf{x},t)\phi_{m,u}(\mathbf{x})+2f_{,v}(\mathbf{x},t)\phi_{m,v}(\mathbf{x})+2f_{,w}(\mathbf{x},t)\phi_{m,w}(\mathbf{x}))\mathrm{d}\Omega\\&+\lambda_2\int_\Omega 2\|f_{,u}(\mathbf{x},t)\times f_{,v}(\mathbf{x},t)\|(\phi_{m,u}(\mathbf{x})f_{,v}(\mathbf{x},t)+\phi_{m,v}(\mathbf{x})f_{,u}(\mathbf{x},t))\\&+2\|f_{,v}(\mathbf{x},t)\times f_{,w}(\mathbf{x},t)\|(\phi_{m,v}(\mathbf{x})f_{,w}(\mathbf{x},t)+\phi_{m,w}(\mathbf{x})f_{,v}(\mathbf{x},t))\\&+2\|f_{,w}(\mathbf{x},t)\times f_{,u}(\mathbf{x},t)\|(\phi_{m,u}(\mathbf{x})f_{,w}(\mathbf{x},t)+\phi_{m,w}(\mathbf{x})f_{,u}(\mathbf{x},t))\mathrm{d}\Omega.\end{aligned} \tag{3.5}$$

The control points are updated dynamically at every time step to generate a new spatial transformation function that corresponds to a better match between the images. We have

$$\mathbf{P}^{s+1}=\mathbf{P}^s-\epsilon\delta E^s(f(\mathbf{x},t)), \tag{3.6}$$

where $\mathbf{P}^{s+1}$ and $\mathbf{P}^s$ are control points evaluated at the current and previous iterations respectively. The time step ϵ can be determined using a line search. We empirically set the value of ϵ based on the convergence of the registration metric. $\delta E^s(f(\mathbf{x},t))$ is computed at the previous time step using a Gaussian quadrature rule. Upon updating the control points, we compute the updated $f(\mathbf{x},t)$ and the evolving source image $I_1(f(\mathbf{x},t))$ at the end of each iteration. For each refinement level, we compute a pre-defined error, e.g. the similarity ratio (RS) explained later in Eq. (4.2) in Section 4.1. The iteration loop is terminated if the change in similarity falls below a given tolerance (e.g., 10^{-4}).

3.2 Adaptive refinement

Here, we describe the procedure to implement adaptive refinement using THB-splines in image registration. In uniform refinement methods [18, 20], when the registration accuracy does not improve further at a certain refinement level, the solution is carried forward on a uniformly refined grid to capture localized deformations. In these methods, the computational cost increases as the number of refinement levels is increased. During image registration, there are only certain regions of the image that undergo large deformation. We can use the image difference (I_g) to detect these regions. We compute I_g for all the elements of the control grid, and the mean value of $I_g(G_{mean})$. We set a threshold parameter ρ that controls the amount of refinement required. For the elements within the

support of each active B-spline, if the local mean value of I_g exceeds ρG_{mean}, we refine this active B-spline basis function. This process is carried out at each refinement level, where the control points are added in the localized regions with large deformation.

Unlike 2D registration using finite element method in [25], here the control points are dynamically updated without constructing large matrices, thus improving the computational efficiency of the 3D image registration process. The THB-spline based adaptive refinement algorithm is shown in Algorithm 1.

Algorithm 1: Adaptive refinement using THB-splines

Input : Source image $I_1(\mathbf{x})$, target image $I_2(\mathbf{x})$, Δ, $MAXITER$, ρ, $l_{\max}$, ϵ, $\mathbf{P}^1_{thb}(t)$ are the B-spline control points such that $f(\mathbf{x},t)=\mathbf{P}^1_{thb}(t)\phi(\mathbf{x})=\mathbf{x}$. Here $t=0$ and the iteration counter $s=0$.

Output: $f(\mathbf{x},t)$ at a particular instance t, such that $I_1(f(\mathbf{x}))\approx I_2(\mathbf{x})$

for level l = 1 to l_{max} **do**
  if $l>1$ **then**
    Compute $I_g=|\nabla(I_1^{l-1}(f(\mathbf{x},t))-I_2(\mathbf{x}))|$ where $I_1^{l-1}(f(\mathbf{x},t))$ is the evolving image obtained at the end of the previous refinement level at the centroid of all active elements of the control grid.
    Compute $G_{mean}=\text{mean}(I_g)$
    for All the active B-splines ϕ_a^{l-1} at $level=l-1$ **do**
      Compute G_j which is the average I_g value for elements within the support of each active B-spline ϕ_a^{l-1}
      if $G_j>\rho G_{mean}$ **then**
        REFINE
      end if
    end for
  end if
  Collect all the active B-spline basis functions, control points and cells from levels l and $l+1$ to obtain the hierarchical structure.
  While $(RS(s+1)-RS(s))>\Delta$ AND $iteration<MAXITER$:

  1. Evaluate the integral $\delta E^s(f(\mathbf{x},t))$.
  2. Update the control points $\mathbf{P}^{l,s+1}=\mathbf{P}^{l,s}-\epsilon\delta E^{l,s}(f(\mathbf{x},t))$.
  3. Compute the spatial transformation function $f(\mathbf{x})=\mathbf{P}^{l,s+1}_{thb}\phi^l_{thb}$.
  4. Compute the transformed image $I_1^l(f(\mathbf{x},t))$ where $t=\epsilon(s+1)$.
  5. $s=s+1$.

end for

The setting of the size of the initial uniform grid at the first refinement level depends on the image complexity. For images containing complex features, the solution is computed at the full image resolution. The setting of the parameters ρ and ϵ is done empirically. These parameters are set constant for a particular refinement level and vary for different refinement levels. ρ depends on the image complexity and the level of refinement. A small value of ρ is suitable for images of higher image complexity in order to capture the complex features accurately. The value of ρ is increased as the refinement levels are increased to prevent too much refinement in the domain. The value of ϵ is set according to the rate of convergence of the solution. For faster convergence, a higher ϵ value is set and if the accuracy does not improve or decreases, the value of ϵ is decreased. It is observed that the value of ϵ is not that sensitive to the solution convergence at a particular refinement level, but needs to be changed for different refinement levels.

3.3 Parallel computation and MATLAB/C++ implementation

The companion software package consists of several sub folders to aid in modularizing the code implementation and can be found at the link (`https://github.com/arpawar/DTHB3D_Reg`). The code implementation has been carried out on a computer of 2.5 GHz quad-core Intel Core i7 processor and 16 GB RAM. The parallel implementation requires Parallel Computing Toolbox to be installed along with the MATLAB 2017a software. For large datasets, the code was run on the XSEDE (Extreme Science and Engineering Discovery Environment) supercomputer called Bridges [33] in Pittsburgh Supercomputer Center. Here the code was run on 2 nodes with 28 cores each having a RAM capacity of 128 GB per node. We need to set the values of the parameters ϵ, ρ, Δ, the maximum number of levels, maximum number of iterations and the number of elements for the first refinement level at the beginning of the registration. These parameters are required as an input for running the code.

For large image data sets, the proposed algorithm can be demanding in terms of memory consumption and the total CPU time required for the entire registration process. Moreover there can be some additional overhead caused due to the construction of the hierarchical adaptive control grids using THB-splines. Thus we have improved the implementation using parallelization. We have utilized the *parfor* loops wherever possible to improve the speed to carry out the numerical integration over the adaptively refined grids. The implementation of the *parfor* for computing numerical integration uses multiple cores to compute the element-wise integration in the parallel framework. In our code, the implementation of the subroutines pertaining to the construction of THB-splines has not yet been parallelized. Note that adaptive refinement of B-splines cannot be easily implemented in parallel due to a transfer of information carried out between B-splines.

To further improve the computational efficiency, we utilized the *MEX* function implementation to convert certain MATLAB functions to C++. These C++ functions can be directly called using *MEX* functions. Through the use of *MEX* functions, the speed of the code can be accelerated drastically. We have used MEX functions during the refinement

step of B-splines and during the numerical integration over the image domain. Details of the software installation and the implementation of each function can be found in the *README* file attached with the code.

4 Numerical examples and discussions

In this section we show the testing results of synthetic and medical images. We also compare our results with the level set method [9] and the optical flow method [36]. The implementation of the level set method [4] and the optical flow method [5] has been carried out in MATLAB. The comparison of the computational efficiency between the proposed method and uniform grid refinement is shown. The main contributions are highlighted through the examples.

4.1 Synthetic images

For the first set of the numerical examples, we demonstrate the results of two pairs of synthetic images. The registration accuracy is measured using similarity metrics, such as the mean squared difference (MSD) and the similarity ratio (RS) [18]. MSD is defined as

$$MSD = \frac{1}{N}\sum_{n=1}^{N}(I_2(n)-I_1(n))^2, \tag{4.1}$$

where N denotes the total number of voxels of the image. RS is defined as

$$RS(I_2(\mathbf{x}),I_1(f(\mathbf{x},t))) = \left(1-\frac{||I_2(\mathbf{x})-I_1(f(\mathbf{x},t))||_2}{||I_2(\mathbf{x})-I_1(\mathbf{x})||_2}\right)\times 100\%, \tag{4.2}$$

where $I_1(f(\mathbf{x},t))$ is the evolving image obtained at a particular instance t. A higher value of RS corresponds to a better registration result, with $RS=100\%$ corresponding to a perfectly registered image.

In Fig. 2, we perform the registration on two levels where the value of ρ is set to be 2.5 for the second level and ϵ is set to 0.00015 for both the refinement levels. The RS metric at the end of the registration process reaches 91.87%. We compare the computational time (wall time) with uniform grid refinement in Table 1. It is obvious that our method can achieve the same accuracy more efficiently.

In Fig. 3, the process is carried out on three refinement levels. The refinement parameter ρ is set to 2.5 and 3 for the second and third refinement levels. The time step ϵ is set as 0.00015 for the first and second refinement levels and 0.0001 for the third refinement level. The value of RS at the end of registration is 92.28%. In comparison, the registration accuracy using the level set method [9] is observed to reach 84.99% for Fig. 2 and 88.51% for Fig. 3. For both examples in Figs. 2 and 3, the maximum iteration number for each refinement level is set as 50 and the error tolerance is set as 10^{-4}. Note that there is a

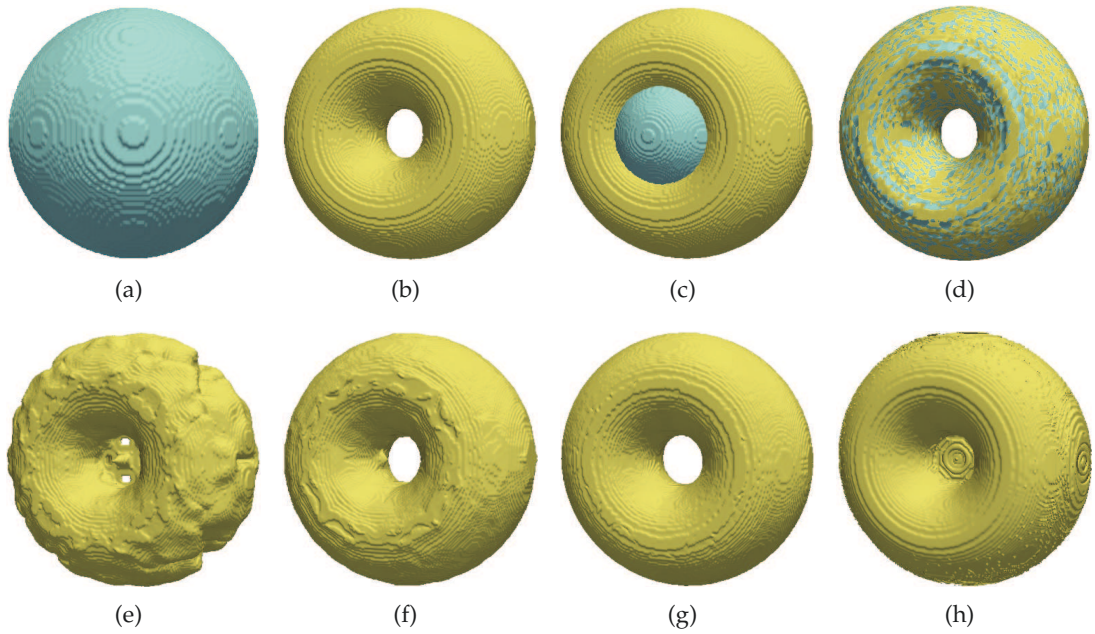


Figure 2: Registration of Sphere to Torus (size: $200\times200\times200$): the source image (a), the target image (b), initial image difference (c), final image difference (d), evolving image at level 1 (e), evolving image at level 2 (f), final registered image using DTHB3D_Reg (g) and final registered image using the level set method [9] (h).

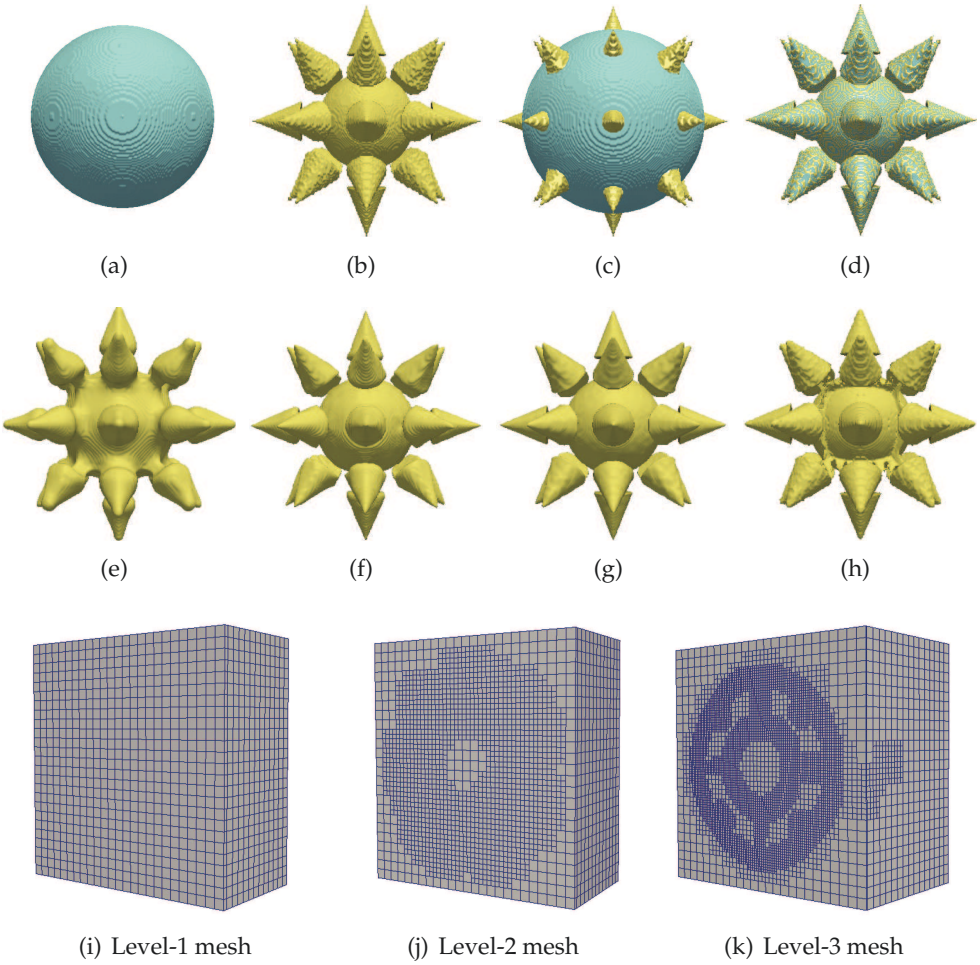
Table 1: Comparison of our adaptive refinement method with uniform B-spline refinement for the Sphere to Torus example in Fig. 2.

L *	Iterations		Control Points		RS (%)		CPU Time (s)	
	Uniform	Adaptive	Uniform	Adaptive	Uniform	Adaptive	Uniform	Adaptive
1	11	11	19,683	19,683	89.16	89.16	39.5	39.5
2	3	3	140,608	56,468	91.87	91.87	203.5	92.7

* Refinement levels

topology change in these two examples. Our algorithm is able to capture these topology changes and result in a more accurate match with the target image. In addition, sharp features of the registered image in Fig. 3 are also captured accurately as can be seen in Fig. 4.

Table 2 shows the number of iterations, control points, the values of MSD and RS obtained at each refinement level. We observe that larger-scale deformation is captured on coarser grids and fine-scale deformations are captured on finer grids. By observing the convergence of the RS value, we can see that when the accuracy does not improve further at a refinement level, the process is stopped at that level and continues on the next level. In this way, we prevent unrealistic deformations and stabilize the registration process.

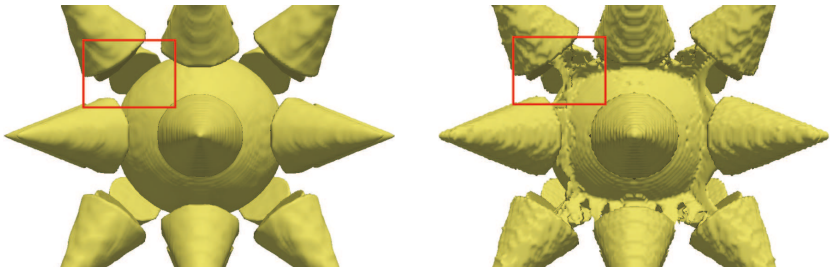


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) Level-1 mesh (j) Level-2 mesh (k) Level-3 mesh

Figure 3: Registration of Sphere to Sun (size: $200\times200\times200$): the source image (a), the target image (b), initial image difference (c), final image difference (d), evolving image at level 1 (e), evolving image at level 2 (f), final registered image using DTHB3D_Reg (g) and final registered image using the level set method [9] (h). The adaptively refined control grids at each refinement level are shown in (i-k).



(a) (b)

Figure 4: Comparison of the registered images obtained from the DTHB3D_Reg method (a) and the level set method [9] (b) for the example shown in Fig. 3.

Table 2: MSD and RS computed using the proposed adaptive refinement algorithm.

Images	Level	Iterations	Control Points	MSD	$RS(\%)$
Sphere-Torus (Fig. 2)	1	11	19,683	169.99	89.16
	2	3	56,468	95.62	91.87
Sphere-Sun (Fig. 3)	1	14	19,683	205.86	86.07
	2	30	49,868	73.07	91.70
	3	10	230,434	63.15	92.28

4.2 Medical images

We illustrate medical image registration between images taken from different subjects. Inter-subject registration is challenging and complex. The anatomical features may vary with respect to size and location, and it requires a robust registration algorithm to perform large deformation to capture these changes accurately.

In Fig. 5, we perform registration of human brain MRI images [1] on four refinement levels. The registration process performs highly localized image deformation to accurately register certain complex features of the brain. The refinement parameter is set to be 1, 4 and 6 for the second, third and fourth levels, respectively. The time step ϵ is set as 0.0002 for the first level, 0.0005 for the next two refinement levels, and 0.0002 for the fourth refinement level. Fig. 5 (d, h and l) show the adaptive grids obtained at each refinement level, where control points are only added in the regions with large deformation and the rest of the control grid is kept coarse. This improves the computational efficiency while maintaining the same registration accuracy as compared with uniform refinement. To validate the registration result, we evaluate the Dice Similarity (DS) metric for the registered images to validate the registration process and compare the segmentation accuracy with the level set method [9] and the optical flow method [36]. DS of two image volumes is given by

$$DS(label) = \frac{2d_c}{d_1+d_2}, \tag{4.3}$$

where d_1, d_2 and d_c are the number of voxels corresponding to a particular class label in the registered image, target image and the regions common to both the registered and target image, respectively.

In Fig. 6, the DS metric is computed for 12 materials using the DTHB3D_Reg method, the level set method and the optical flow method. The average segmentation accuracy is computed for ten pairs of brain MRI datasets taken from [1]. It can be seen that both the DTHB3D_Reg method and the level set method show comparable results and have better segmentation accuracy than the optical flow method. The gray matter and the white matter constitute larger fraction of the total brain volume. For these two materials we obtained a high segmentation accuracy. The average values of *DS (gray matter)* and *DS (white matter)* are 80.5% and 83.8%, respectively. In this example, there is no obvious

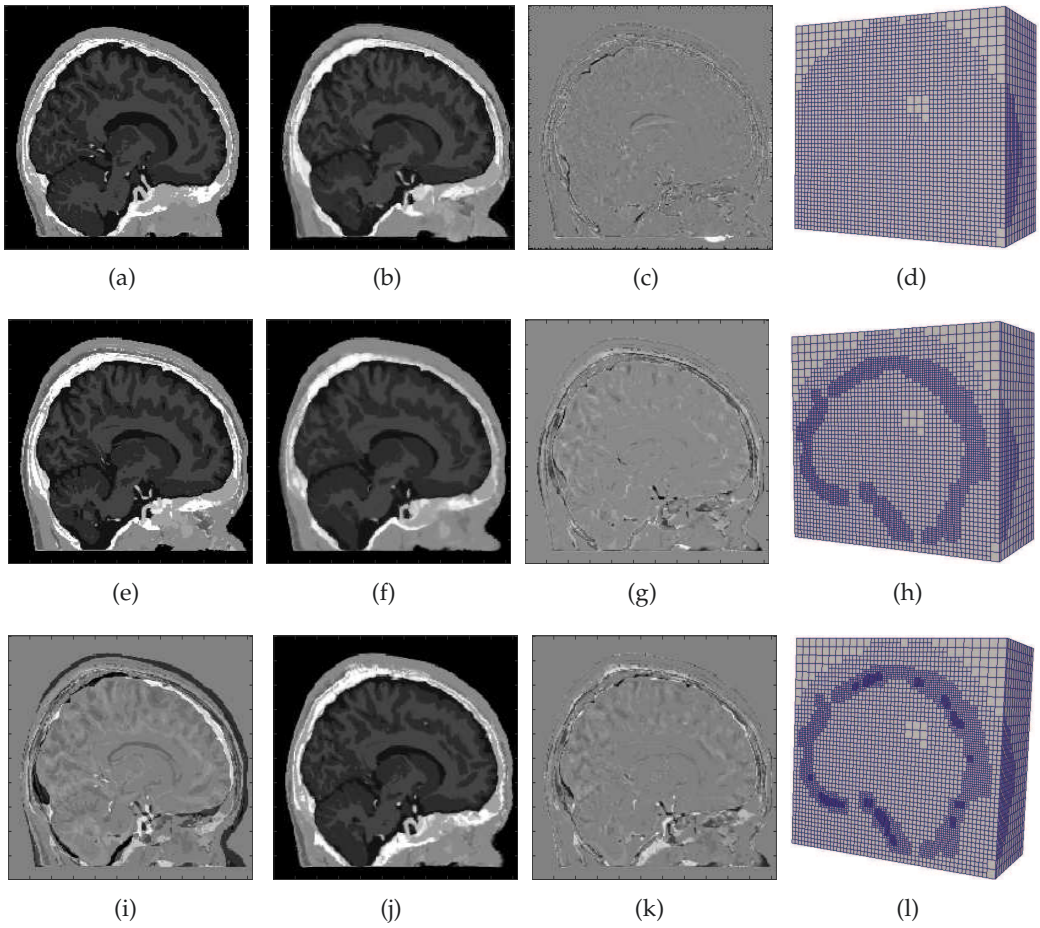


Figure 5: Registration of brain MRI image (size: $200\times224\times200$): the sagittal MRI of the source image (a), the target image (e) and the initial image difference (i). Second column: the registered image using DTHB3D_Reg (b), the level set method (f) and the optical flow method (j). Third column: the final image difference between the registered and target images using DTHB3D_Reg (c), the level set method (g) and the optical flow method (k). Fourth column: the adaptively refined grids at levels 2 (d), 3 (h) and 4 (l).

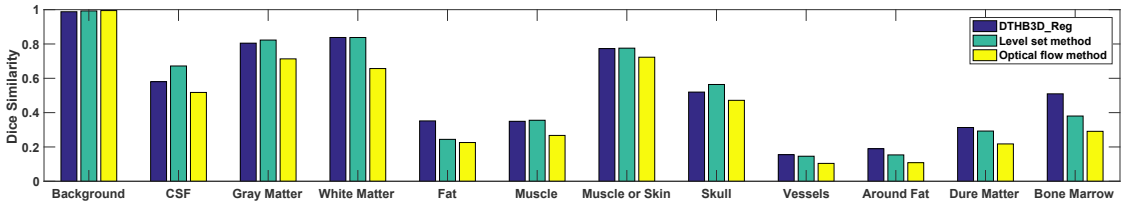


Figure 6: Average segmentation accuracy between the registered images and the target images of 10 pairs of brain MRI evaluated using the DTHB3D_Reg method, the level set method and the optical flow method, respectively.

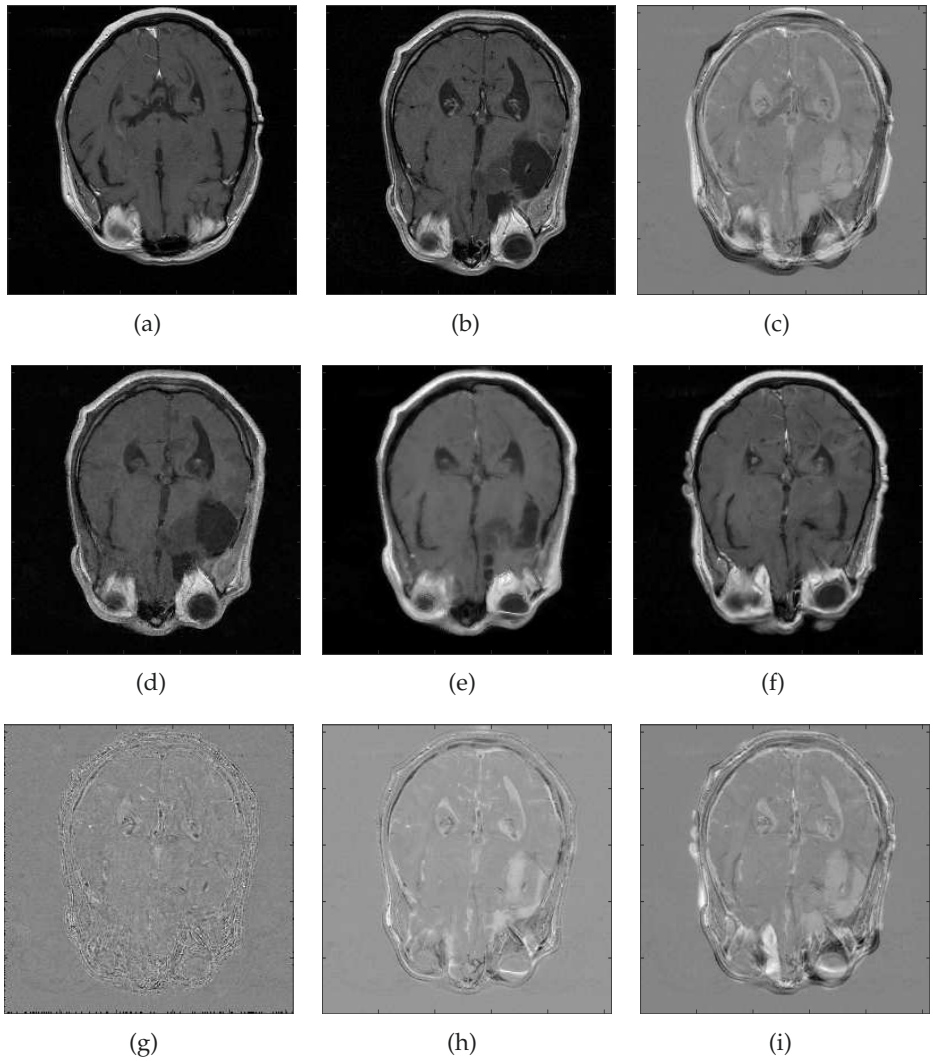


Figure 7: Inter-subject registration of brain MRI volume (size: $256\times256\times52$): the axial MRI at $z=25mm$ of the source image (a), the target image (b) and the initial image difference (c). Second row: the registered image using DTHB3D_Reg (d), the level set method (e) and the optical flow method (f). Third row: the final image difference between the registered and target images using DTHB3D_Reg (g), the level set method (h) and the optical flow method (i).

topology change between the images, and our method gives similar or better results as the level set method [9] for most types of tissue.

In the following examples we perform registration on medical images with significant changes in the anatomical features. For a better visualization, we have only shown the axial slices of the image volumes. In the case of the inter-subject registration as shown in Figs. 7 and 8, the change in feature location and size is significant. In Fig. 7, the two brain MRI volumes were obtained from two individuals from the RIRE database [2]. The

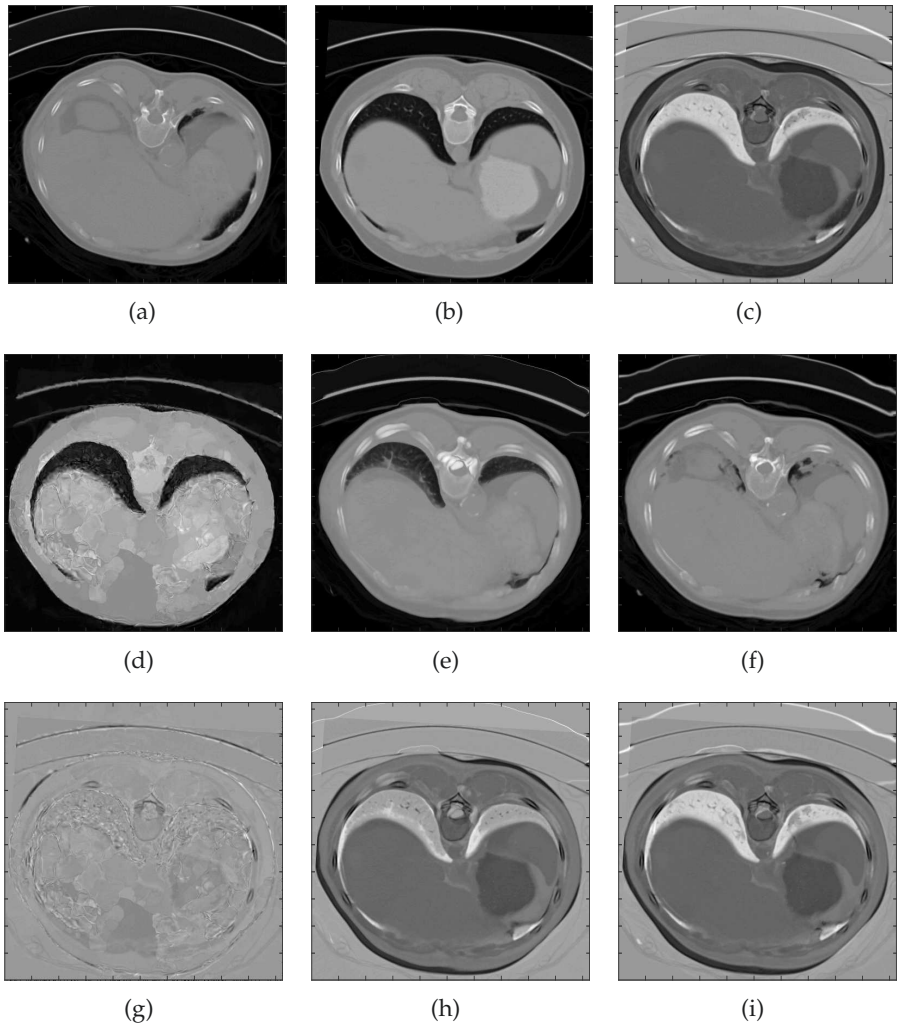


Figure 8: Inter-subject registration of the liver CT images (size: $512\times512\times35$): the axial MRI at $z=25mm$ of the source image (a), the target image (b) and the initial image difference (c). Second row: the registered image using DTHB3D_Reg (d), the level set method (e) and the optical flow method (f). Third row: the final image difference between the registered and target images using DTHB3D_Reg (g), the level set method (h) and the optical flow method (i).

registration process is performed on four refinement levels where ρ is set as 1, 1.5 and 5 for the second, third and fourth levels, respectively. ϵ is set as 0.0002 for the first two levels and 0.0005 for the next two refinement levels. The maximum iteration number for each refinement level is set as 30 and the error tolerance is set as 10^{-4}. In this example one can clearly identify that there is topology change in the target image due to resection operation. Compared with the level set method and the optical flow method, our DTHB3D_Reg method captures these significant changes in the anatomical features more accurately.

In Fig. 8, the liver CT images [6] are initially aligned using rigid transformation. Here we use the command *imregister()* with the option of "rigid" transform. We then perform nonrigid registration to capture the localized and complex deformations. In this example, we perform registration on four refinement levels with ρ set as 1 for the second and third levels and 3 for the last level. ϵ is set as 0.00005 for the first level and 0.000075 for the next three refinement levels. The maximum iteration number for each refinement level is set as 100 and the error tolerance is set as 10^{-4}. It is observed for all numerical examples that higher image complexity corresponds to a smaller timestep value for optimum convergence of the registration result.

In Fig. 9, we show an example of pre-operative and post-operative brain MRI where there is large topology change due to the removal of tumor. The patient underwent surgical procedure to remove glioma in the left frontal region of the brain [3]. The registration process is carried out on four refinement levels, with the values of ρ set as 1, 3 and 6 for the second, third and fourth levels, respectively. In this example, ϵ is set as 0.0003 for the first level, 0.0008 for the second and third levels and 0.0005 for the fourth level. The maximum iteration number for each refinement level is set as 50 and the error tolerance is set as 10^{-4}. From the corresponding slices at the same axial location, we can see that the topology change due to the surgical procedure is captured accurately using our DTHB3D_Reg method as compared to other methods. Note that for four refinement levels, we need run our code on a computer with higher RAM and higher number of cores to fully implement the parallel framework. The default value of the maximum refinement levels is set as three for easy implementation on a computer with 16 GB RAM.

Table 3 shows the MSD values computed for the above examples using the DTHB3D_Reg, level set and optical flow methods. It is obvious that our DTHB3D_Reg method yields the most accurate results indicated by the lowest MSD values. We also compare the total number of control points used in the DTHB3D_Reg method with the level set method which performs uniform refinement. Based on the tabulated results for all examples in Tables 3 and 4, we can see that we achieve higher accuracy per degree of freedom. In addition, the total number of degrees of freedom used for the registration process are much lower than the other methods.

Table 3: Comparison of the registration error (MSD) using DTHB3D_Reg, level set and optical flow methods.

Images	Initial MSD	DTHB3D_Reg	Level set	Optical flow
Brain MRI (Fig. 5)	1.94×10^3	161.25	263.80	489.19
Brain MRI (Fig. 7)	1.86×10^3	68.41	478.76	917.26
Liver CT (Fig. 8)	1.98×10^3	244.49	1.56×10^3	1.57×10^3
Brain tumor MRI (Fig. 9)	886.19	61.76	218.46	515.03

Although the proposed method is more efficient as compared to uniform refinement methods, the computational cost is still high and the process is time consuming for large datasets. Parallelization has been introduced in the implementation to make it more ef-

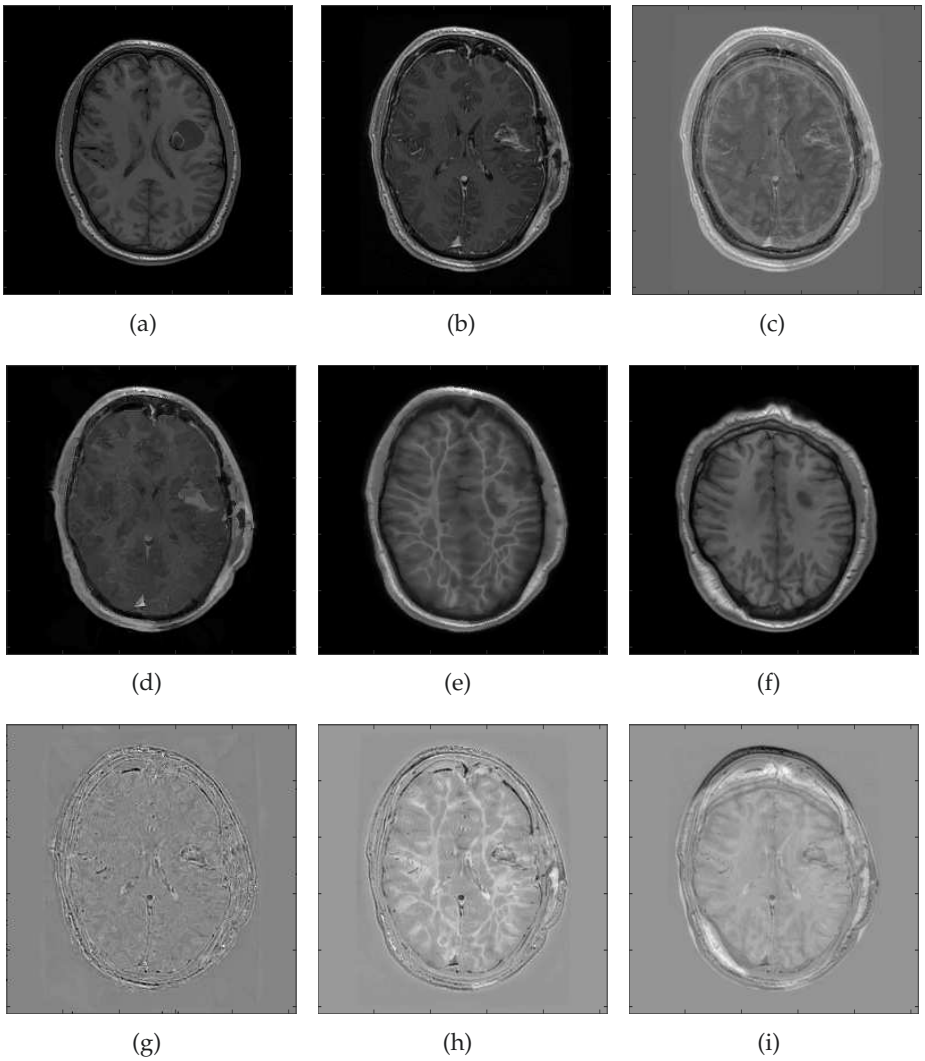


Figure 9: Pre-operative and post-operative image registration of brain MRI image (size: $256\times256\times130$): the axial MRI at $z=64mm$ of the source image (a), at $z=79mm$ of the target image (b) and the initial image difference (c). Second row: the registered image using DTHB3D_Reg (d), the level set method (e) and the optical flow method (f). Third row: the final image difference between the registered and target images using DTHB3D_Reg (g), the level set method (h) and the optical flow method (i).

ficient. However, the implementation of the adaptive refinement using THB-splines is yet to be parallelized, thus limiting the speed up of the code. For very large datasets, the memory usage increases drastically, thus limiting the maximum refinement levels that can be used. The framework can only be used for the registration of grayscale images of the same imaging modality and having the same range of intensity values for the pair of images. This is because we use SSD as the metric in the energy functional. Furthermore, the registration process can be only be applied for the pair of images of equal sizes.

Table 4: Comparison of the total number of B-spline control points used by the DTHB3D_Reg (adaptive refinement) and the level set (uniform refinement) methods.

Images	DTHB3D_Reg	Level Set
Brain MRI (Fig. 5)	307,830	9,354,443
Brain MRI (Fig. 7)	185,714	3,689,455
Liver CT (Fig. 8)	188,793	10,078,550
Brain tumor MRI (Fig. 9))	234,407	8,921,773

5 Conclusions and future work

A robust method to perform 3D nonrigid image registration suitable for large image deformation and topology change is proposed. The optimum spatial transformation is defined using THB-splines. Adaptive refinement is introduced by locally detecting and refining only those regions that undergo fine-scale deformations. Dynamic implementation of the optimization is carried out where the energy functional is minimized using the strong formulation directly. Based on the numerical results, we have shown that our method is more accurate in capturing large deformation and is robust to topology change. We have developed a software package named DTHB3D_Reg which is freely available in the public domain. In the future work, we will focus on further applications of our algorithm to carry out joint segmentation and registration of medical images. Image registration via hp-refinement and coarsening is another area we will study. We will also focus on the parallel implementation of the THB-splines in our software package in order to further improve the code efficiency.

Acknowledgments

The research at Carnegie Mellon University was supported in part by the PECASE Award N00014-16-1-2254 and NSF CAREER Award OCI-1149591. The research at Bauhaus University Weimar was supported in part by the ITN-INSIST and ERC-COMBAT funded by the EU-FP7 (PITN-GA-2011-289361). This work used the XSEDE in Pittsburgh Supercomputer Center, which is supported by NSF grant ACI-1548562.

References

[1] Brainweb: Simulated brain database. `http://www.bic.mni.mcgill.ca/brainweb/`.
[2] RIRE: The retrospective image registration evaluation project. `http://www.insight-journal.org/rire/download_data.php`.
[3] Surgical planning laboratory. `http://www.spl.harvard.edu/publications/item/view/1915`.
[4] Composed Directly Manipulated Free Form Deformation Image Registration. `https://github.com/stellaccl/cdmffd-image-registration`.

[5] Diffeomorphic Log Demons Image Registration. `http://www.mathworks.com/matlabcentral/fileexchange/39194-diffeomorphic-log-demons-image-registration`.
[6] MIDAS. `http://www.insight-journal.org/midas/collection/view/38`.
[7] R. S. Alves and J. M. R. S. Tavares. Computer image registration techniques applied to nuclear medicine images. In *Computational and Experimental Biomedical Sciences: Methods and Applications*, pages 173–191. Springer, 2015.
[8] P. B. Bornemann and F. Cirak. A subdivision-based implementation of the hierarchical B-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013.
[9] C.L. Chan, C. Anitescu, Y. Zhang, and T. Rabczuk. Two and three dimensional image registration based on B-spline composition and level sets. *Communications in Computational Physics*, 21(2):600–622, 2017.
[10] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14(2):87–111, 1980.
[11] W. Crum, D. Rueckert, M. Jenkinson, D. Kennedy, and S. Smith. A framework for detailed objective comparison of non-rigid registration algorithms in neuroimaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 679–686. Springer, 2004.
[12] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 70(4):76–86, 2008.
[13] G. Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.
[14] D. Forsey and R. Bartels. Hierarchical B-spline refinement. *ACM SIGGRAPH Computer Graphics*, 22(4):205–212, 1988.
[15] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: the truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.
[16] A. A. Goshtasby. *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.
[17] M. S. Hansen, R. Larsen, B. Glocker, and N. Navab. Adaptive parametrization of multivariate B-splines for image registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
[18] Y. Jia, Y. Zhang, and T. Rabczuk. A novel dynamic multilevel technique for image registration. *Computers and Mathematics with Applications*, 69(9):909–925, 2015.
[19] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M. C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *NeuroImage*, 46(3):786–802, 2009.
[20] J. Leng, G. Xu, and Y. Zhang. Medical image interpolation based on multi-resolution registration. *Computers and Mathematics with Applications*, 66(1):1–18, 2013.
[21] X. Li, X. Long, P. Laurienti, and C. Wyatt. Registration of images with varying topology using embedded maps. *IEEE transactions on medical imaging*, 31(3):749–765, 2012.
[22] J. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
[23] F. P. M. Oliveira and J. M. R. S. Tavares. Enhanced spatio-temporal alignment of plantar

pressure image sequences using B-splines. *Medical & Biological Engineering & Computing*, 51(3):267–276, 2013.
[24] F. P. M. Oliveira and J. M. R. S. Tavares. Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(2):73–93, 2014.
[25] A. Pawar, Y. Zhang, Y. Jia, X. Wei, T. Rabczuk, C. L. Chan, and C. Anitescu. Adaptive FEM-based nonrigid image registration using truncated hierarchical B-splines. *Computers & Mathematics with Applications*, 72:2028–2040, 2016.
[26] A. Pawar, Y. Zhang, X. Wei, Y. Jia, T. Rabczuk, C. L. Chan, and C. Anitescu. An adaptive non-rigid image registration technique using hierarchical B-splines. In *5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing, VipIMAGE 2015*. CRC Press, 2016.
[27] D. Rueckert, P. Aljabar, R. Heckemann, J. Hajnal, and A. Hammers. Diffeomorphic registration using B-splines. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 702–709. Springer, 2006.
[28] 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 Transactions on Medical Imaging*, 18(8):712–721, 1999.
[29] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM Transactions on Graphics*, volume 23, pages 276–283. ACM, 2004.
[30] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 194–201, 1994.
[31] J. M. R. S. Tavares. Analysis of biomedical images based on automated methods of image registration. In *International Symposium on Visual Computing*, pages 21–30. Springer, 2014.
[32] J. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical image analysis*, 2(3):243–260, 1998.
[33] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, and G. D. Peterson. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74, 2014.
[34] N. J. Tustison, B. A. Avants, and J. C. Gee. Improved FFD B-spline image registration. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
[35] B. Vemuri, J. Ye, Y. Chen, and C. Leonard. Image registration via level-set motion: applications to atlas-based segmentation. *Medical image analysis*, 7(1):1–20, 2003.
[36] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: efficient non-parametric image registration. *NeuroImage*, 45(1):61–72, 2009.
[37] S. Warfield, F. Jolesz, and R. Kikinis. A high performance computing approach to the registration of medical imaging data. *Parallel Computing*, 24(9):1345–1368, 1998.
[38] Z. Xie and G. E. Farin. Image registration using hierarchical B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):85–94, 2004.
[39] Y. Zhang, Y. Jing, X. Liang, G. Xu, and L. Dong. Dynamic lung modeling and tumor tracking using deformable image registration and geometric smoothing. *Molecular & Cellular Biomechanics*, 9(3):213–226, 2012.
[40] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.