Deep Learning Real-Time Adaptive Physics-based Non-Rigid Registration for Accurate Geometry Representation of Brain in Modeling Deformation During Glioma Resection

Angelos Angelopoulos
CRTC

Nikos Chrisochoides
CRTC

Abstract

The Physics-based Non-Rigid Registration (PBNRR) framework allows for accurate real-time medical image registration and geometry representation of the brain in modeling deformation during glioma resection. Existing adaptive PBNRR (APBNRR) shows promise in being able to be utilized in time-constrained image-guided neurosurgery operations, but the issue of determining patient-specific input parameters to allow for optimal registration remains an open problem. We present a deep feedforward neural network that can predict sets of possible optimal or suboptimal input parameters that lead to a low Hausdorff distance of the registered image from the preoperative image. The neural network is trained on output produced by over 2.6 million retrospective APBNRR executions consisting of an almost exhaustive parameter study using cloud computing on 13 patient cases spanning from partial to excessive tumor resection. By utilizing the neural network, we can greatly reduce the parameter space that needs to be evaluated with APBNRR in order to achieve optimal results, and initial experiments have been very promising.

1. Introduction

The Physics Based Non-Rigid Registration (PBNRR) formulation of brain shift and deformation problem during brain tumor resection can be decomposed into three sub-problems: (i) compute an accurate approximation of brain geometry in the context of tissue removal, (ii) maintain topology of the deformed brain in iMRI and (iii) last but not least, preserve the brain anatomy i.e., geometry and topology of individual brain tissues and their relationship to each other. In this paper we focus on the first sub-problem i.e., accurate representation of the brain geometry which is necessary for the solution of both subsequent problems. Accurate representation of brain geometry is critical for the accurate numerical computations (i.e., integration) in the least-square sense used in PBNRR. Both matching and regularization terms rely on accurate brain geometry representation. In addition, accurate brain geometry presentation is critical for outlier rejection. Other than tissue resection, there are other challenges like tissue retraction but are outside the scope of this study.

In this paper, we also introduce a deep learning component for the adaptive (APBNRR) medical image registration framework [1] which can predict registration parameters that can lead to more accurate registration in the time-constrained environment of a neurosurgery session. The results have been promising, with deep learning APBNRR being ~8.45 times more accurate than rigid registration and ~6.71 times more accurate than B-Spline registration.

2. Related Work

As with many other areas, machine learning has been applied to medical image registration. In this section we will take a closer look at two major applications, VoxelMorph [3] and the method proposed by J. Krebs et al [14]. A brief comparison of the problems addressed, the proposed solutions, and usage of machine learning in these two applications, as well as in APBNRR, is in Table 1.

VoxelMorph [3] addresses the problem of fast deformable medical image registration with a focus on brain MRI, but it can be used for other tissues as well. VoxelMorph uses a solution formulated by an unsupervised learning convolutional neural network for computing a registration field and a spatial transformation function for warping the preoperative image. The application can also use instance specific optimization by fine-tuning the network parameters for each MR image. As noted in the paper [3], it runs in 0.45 seconds on a top-tier GPU and 57 seconds on a CPU, with an average DICE score of 75.3%.

The method proposed by J. Krebs et al [14] tries to solve the problem of probabilistic deformation modeling for diffeomorphic registration for cardiac MRI. The proposed solution utilizes an unsupervised learning conditional variational autoencoder (CVAE) network with an exponentiation layer for creating diffeomorphic transformations. The average execution time is 0.32 seconds on a top-tier GPU with an average DICE score of 79.9% and a mean Hausdorff distance of 7.9 mm.

Application	**Problem**	**Solution**	**Machine Learning Usage**
VoxelMorph [3]	Fast deformable medical image registration (focus on brain MRI, but can be used for other tissues as well)	Unsupervised learning convolutional neural network with spatial transformation function and instance-specific optimization	Computation of registration field
Deep Learning APBNRR	Deformable medical image registration with tumor resection (for brain MRI with focus on accurate geometry representation of brain in modeling deformation)	Adaptive physics-based non-rigid registration with supervised learning deep feedforward neural network and patient-specific features	Prediction of optimal input parameters for APBNRR
J. Krebs et al Method [14]	Probabilistic deformation modeling for diffeomorphic registration (for cardiac MRI)	Unsupervised learning conditional variational autoencoder (CVAE) network with exponentiation layer	Learning of a probabilistic deformation encoding.

Table 1. Shows a comparison of the problems, solutions, and usage of machine learning of three major applications in medical image registration, including APBNRR.

3. Learning-Based Medical Image Registration

Machine learning (ML) has been applied to many areas to help solve problems, and one of them is in medical image computing [2]. Specifically, for medical image registration, many new methods currently utilize convolutional neural networks, which operate directly on the preoperative and intraoperative MRI images to produce a registered image. A notable example is MIT's VoxelMorph model [3], which was presented last year in CVPR. VoxelMorph utilizes a CNN which computes a registration field and warps the preoperative image with the registration field using a spatial transformation function [3]. In VoxelMorph, ML is used directly for image registration, as the image registration functionality is implemented using a CNN, and the goal of the project is to enable fast medical image registration.

In our experimental method of utilizing deep learning in medical image registration, we take a different path, by utilizing a supervised deep feedforward neural network trained to predict optimal input parameter sets for APBNRR, in order to be used in the application to produce registered images with a low Hausdorff distance value. We chose not to build an entirely new image registration framework based on a CNN, as this has been done before, but rather enhance the APBNRR medical image registration framework, because, as discussed later, APBNRR is robust, and can produce registered images with a Hausdorff distance under the boundary of 2 mm. Most importantly, APBNRR also performs tumor resection, which is not currently done by any surveyed CNN-based method, including VoxelMorph [3].

The application of deep learning on APBNRR aims to make it more suitable for use in the constrained time-period of an image-guided neurosurgery session because normally APBNRR is expensive both time-wise and resource-wise and requires a sweep over thousands of parameters to produce optimal results. Our proposed neural network model aims to greatly reduce this parameter space, by predicting the most optimal registration parameters specific to each patient case. Our goal is to enable APBNRR to be used for medical image registration and tumor resection in environments such as the Advanced Multimodality Image Guided Operating suite (AMIGO) of Brigham and Women's Hospital [14].

4. Method

4.1. Adaptive Physics-Based Non-Rigid Registration (APBNRR)

APBNRR takes as input a preoperative (moving) MRI and an intraoperative (fixed) MRI, a preoperative segmented MRI, and a range of twenty-seven registration and mesh generation parameters (indicated in table 1). APBNRR [1] augments PBNRR [2] to accommodate soft-tissue deformation caused by tumor resection. This iterative method adaptively modifies a heterogeneous finite element model to optimize non-rigid registration in the presence of tissue resection. Using the segmented tumor and the registration error at each iteration, APBNRR gradually excludes the resection volume from the model. During each iteration, registration is performed, the registration error is estimated, the mesh is deformed to a predicted resection volume, and the brain model (minus the predicted resection volume) is re-tessellated. Re-tessellation is required to ensure high-quality mesh elements, which is important for the convergence of the linear solver. Fig. 1 illustrates five iterations of APBNRR on a brain with a significant resection volume.

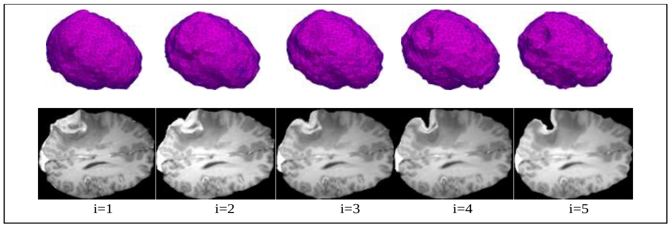

Figure 1: Non-rigid registration using 5 iterations of APBNRR reflects the changes in brain morphology caused by tumor resection. Each column depicts a brain mesh model (top row) and an axial slice of the preoperative image after the non-rigid registration (bottom row), at iteration i.

APBNRR incorporates various algorithms to perform a fast and accurate registration: (1) parallel feature selection, (2) parallel block matching, (3) parallel mesh generation, and (4) finite element solver:

Parallel feature selection. APBNRR identifies image features algorithmically by analyzing voxel intensity variation across the intracranial cavity. For each feature candidate, it computes the variance within a block of size B_S. It then selects F_S features with the highest variance. Experimental evaluation has shown that when $B_S = 3 \; or \; 5$ and $3\% \leq F_S \leq 10\%$, a sufficient number of image blocks ($> 3 \times 10^5$) can be selected. The method also uses a connectivity pattern to avoid selecting blocks that are too close to each other, thereby influencing the distribution of selected blocks in the image. Three simplex-patterns are available: "vertex" (i.e., zero-order simplex implies 26 connectivity), "edge" (i.e., first-order simplex implies 18 connectivity), and "face" (i.e., second-order simplex implies 6 connectivity). The higher the order of the simplex-pattern the higher the density of the selected blocks. Fig. 2 depicts feature selection results using the three patterns. Since the "face" pattern results in a higher density of blocks near the boundaries/interfaces of anatomical structures, features which are expected to be most persistent between pre-operative and intraoperative image acquisitions, it is most suitable for IGNS-based Glioma resection. The parallel implementation partitions the preoperative image into k sub-regions, where k is the number of threads. Each thread computes a variance value and an image index for each feature inside the sub-region. Then, the computed pairs are sorted in parallel based on their variance and merged into a global vector. The size of the global vector is equal to the total number of computed blocks. Finally, $0.5 + N\text{Features} \times Fs$ blocks are selected from the global vector.

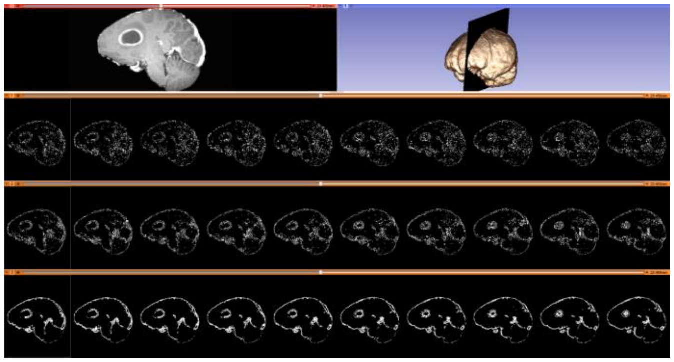

Figure 2: Selected blocks from an MRI volume using various connectivity patterns. Blocks are depicted on ten consecutive sagittal slices. From top to bottom row: sagittal slice (left) and volumetric MRI rendering (right); selected blocks with a "vertex" pattern; selected blocks with an "edge" pattern; selected blocks with a "face" pattern. Number of selected blocks for all patterns: 322060.

Parallel block matching. Given a block in the preoperative image and a block matching window W of size W_S in the intraoperative image, this module searches for a block in W that maximizes the similarity between the block in the preoperative image and the block in W. Displacements between corresponding blocks in the preoperative and intraoperative images are used to drive the finite element solver. W_S may be different in the axial, coronal, and sagittal directions due to anisotropic image data. The parallelization of the block matching algorithm is based on image partitioning and exhibits excellent performance [9]. However, in tumor resection, the problem of blocks with a missing correspondence may compromise the accuracy of the non-rigid registration. A large window size cannot solve the problem because: (i) it is opposed to the assumption that a complex non-rigid transformation can be approximated by point-wise translations of small image regions, (ii) it may lead to

Parameter	Description	Parameter (cont.)	Description (cont.)
1. Half Block Size X	Half block size (in voxels) for the X dimension.	**15. Mesh Method**	Meshing method (BCC, Delaunay, LD, Hex)
2. Half Block Size Y	Half block size (in voxels) for the Y dimension.	**16. Trade-off**	Trade-off between mechanical energy and matching energy in energy minimization equation. The larger the trade-off, the more weight is given to the matching energy term.
3. Half Block Size Z	Half block size (in voxels) for the Z dimension.	**17. Shape Function Type**	FEM shape function type (Linear, Linear with ESF, Quadratic, Quadratic with ES)
4. Half Window Size X	Half block matching window size (in voxels) for the X dimension.	**18. Linear Solver Type**	Linear solver type (LSQR, ITPACK, LU, BICGSTAB)
5. Half Window Size Y	Half block matching window size (in voxels) for the Y dimension.	**19. Young's Modulus (parenchyma)**	Young's modulus for brain parenchyma.
6. Half Window Size Z	Half block matching window size (in voxels) for the Z dimension.	**20. Poisson Ratio (parenchyma)**	Poisson ratio for brain parenchyma.
7. Selection Fraction	Percentage of selected blocks from total number of blocks.	**21. Young's Modulus (tumor)**	Young's modulus for tumor.
8. Approximation Steps	Number of approximation iterations.	**22. Poisson Ratio (tumor)**	Poisson ratio for tumor.
9. Interpolation Steps	Number of interpolation iterations.	**23. CBC3D Spacing**	Lattice spacing (in mm) for CBC3D.
10. Adaptive Iterations	Maximum number of adaptive iterations	**24. CBC3D Fidelity**	Mesh fidelity for CBC3D.
11. Zero Blocks Fraction	Minimum number of blocks with zero correspondence.	**25. CBC3D Iterations**	Smoothing iterations for CBC3D.
12. Rejection Fraction	Percentage of rejected outlier blocks.	**26. CBC3D Multiple Fidelities**	Multiple fidelities for CBC3D.
13. Similarity Metric	Similarity metric for block matching (NCC or NMI).	**27. Delaunay Delta**	Tetrahedron size for Delaunay meshing.
14. Non-connectivity	The pattern for feature selection.		

Table 2. Shows the tunable parameters utilized by APBNRR. Parameters 1-12 are utilized in the deep learning model, while the rest are fixed. I/O parameters are not included in this table.

unrealistic matches that deteriorate the quality of the transformation, and (iii) it imposes performance overheads due to an exhaustive search on a large number of blocks. APBNRR introduces a heuristic two-step block matching algorithm to eliminate the blocks with a missing correspondence. First, a traditional block matching [9] maximizes the similarity between blocks in the preoperative image and blocks in W. Next, a parallel closest block matching pairs the blocks with a missing correspondence with their closest point on the surface S of the brain in the intraoperative image.

Parallel mesh generation. The image segmentation is used to generate a patient-specific finite element mesh for APBNRR. The quality of the tetrahedral mesh influences the numerical accuracy of the solution and the correctness of the estimated transformation. The higher the quality of the elements (i.e., the larger the minimum dihedral angle), the better the convergence of the linear solver. A parallel Delaunay meshing method is employed to tessellate the segmented brain with high-quality tetrahedral elements and to model the brain surface with geometric and topological guarantees [10]. Both single-tissue (i.e., brain parenchyma) and multi-tissue (i.e., brain parenchyma and tumor) meshes are generated. Fig. 3 depicts one of the multi-tissue meshes. Parameter δ determines the size of the mesh, where a smaller $\delta > 0$ generates a larger mesh.

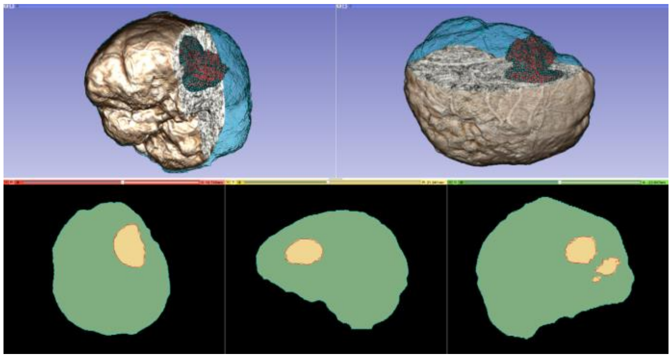

Figure 3: A multi-tissue (brain parenchyma, tumor) finite element mesh used for APBNRR (number of tetrahedral elements: 160179; minimum dihedral angle: 4.41˚). Top row: the mesh superimposed on a volume rendering of the MRI data. Cyan and red represent the brain parenchyma and tumor meshes, respectively. Bottom row: mesh fidelity illustrated on an axial, sagittal, and coronal slices. Each slice depicts a 2D cross-section of the mesh surface (cyan and red lines) and the segmented volume (green and yellow regions). The closer the mesh surface is to the segmented boundaries, the higher the mesh fidelity.

Finite element solver. This module assembles a $N \times N$ global stiffness matrix $K_g = K_m + K_b$ for the biomechanical brain model, where $N = 3 \times n$ is the number of degrees of freedom in the model, n is the number of mesh vertices, and K_m, K_b are the stiffness matrices of the mesh and the selected blocks, respectively. The stiffness matrix depends on the geometry/quality of the mesh, the interpolation polynomial-order (e.g., linear), and the mechanical properties (E_b, v_b) of the underlying materials. Matrix K_m is assembled from the individual stiffness matrices of mesh elements [11]. Matrix K_b represents the stiffness of the blocks. After associating each block with an element, a 3×3 stiffness tensor is computed by interpolating the element stiffness at the block position [12]. Matrix K_b is assembled from the individual 3×3 block tensors. The FEM solver constrains the mesh model by applying displacements determined during block matching, represented by external force vector F, to the associated mesh elements. The displacements U of unconstrained mesh vertices are then estimated by solving an $N \times N$ linear system of equations $F = K_g \times U$. The FEM solver employs a BICGSTAB (bi-conjugate gradient stabilized) iterative solver [13] to compute U. Mesh deformations are estimated using an approximation of an interpolation-based formulation which rejects feature outliers (i.e., blocks with the largest error between U and the block matching displacements). In each outlier rejection step, $(N_b \times F_r)/N_{rej}$ outliers are removed, where N_b is the number of selected blocks, F_r is the fraction of the rejected blocks, and N_{rej} is the number of outlier rejection iterations (Table ?). Previous experiments have shown that $F_r = 25\%$ is sufficient to reject all significant outliers without rejecting relevant matches. Deformation at each image voxel can be estimated by interpolating the final solution U across mesh elements.

Non-rigid registration of medical resonance images that compensates for brain shift and tumor resection is a difficult problem, and APBNRR offers a solution [1]. MRIs are registered iteratively, and the finite element model changes in every iteration to reflect the current morphology of the brain (fig.1). However, APBNRR cannot currently be effectively utilized in a clinical setting, due to the large parameter space that needs to be evaluated for every individual patient so that optimal registration can take place. Before APBNRR can be used in IGNS, the parameter space needs to be greatly reduced. This can take place with the proposed deep learning model, which is trained on millions of past APBNRR executions and predicts a set of optimal parameters specifically tuned for each patient.

4.2. Deep Learning Model Architecture

Our proposed model consists of a supervised deep feedforward neural network, which is trained on data produced from past APBNRR executions. The model takes as input fourteen parameters: twelve APBNRR parameters that have an infinite range of possible values and two extra, patient-specific parameters. It produces as output a predicted Hausdorff distance of the registered preoperative image. The twelve APBNRR parameters are: the half block size $\mathbf{B_s}$, and the half window size $\mathbf{W_s}$ in the axial, coronal, and sagittal direction, the fraction $\mathbf{F_s}$ to select the image blocks, the number of approximation (outlier rejection) steps $\mathbf{N_{rej}}$, the number of interpolation steps $\mathbf{N_{int}}$, the maximum number of adaptive iterations $\mathbf{N_{iter,.max}}$, the minimum number of blocks with a zero correspondence $\mathbf{N_{b0,min}}$, and the percentage of rejected outlier blocks $\mathbf{F_r}$. The two patient-specific parameters are the location of the tumor in the brain (lobe-wise) and the degree of brain deformation, which can be directly inferred from the rigid registration error. These two parameters are used to improve model performance by providing additional cues for the neural network to learn, as well as to fine-tune the model for a specific patient during an IGNS session.

The neural network was implemented using Keras [6] on a TensorFlow [7] backend. It consists of 4 hidden fully connected layers, each comprised of 128 neurons. We used ReLU [4] as the activation function and stochastic gradient descent with Nesterov momentum [5] for optimization. We chose to use SGD because it has been shown to lead to better

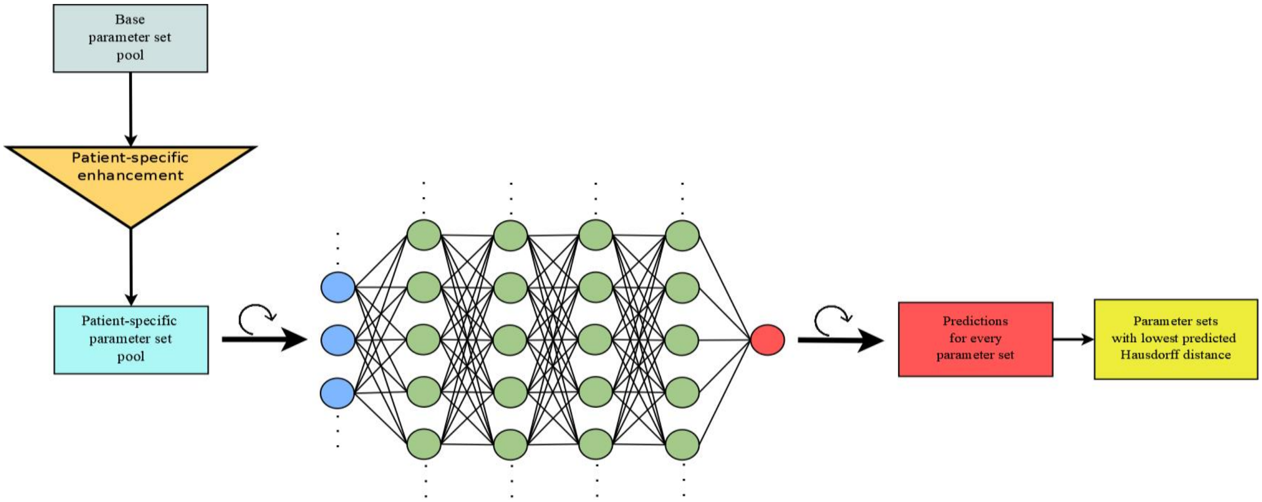
Base parameter set pool
Patient-specific enhancement
Patient-specific parameter set pool
Predictions for every parameter set
Parameter sets with lowest predicted Hausdorff distance

Figure 4. Deep learning portion of APBNRR registration.

generalization in comparison to adaptive gradient optimizers such as Adam, due to its tendency to converge to better global minima [8]. No dropout layers or methods of preventing overfitting were used, as the training data is complex and highly variated. Furthermore, the feature selection in parallel mode is non-deterministic, meaning that the same input parameters can yield different outputs in every execution, further reducing the ability of the model to "memorize" the training data and overfit to the training set. The neural network was built in an iterative fashion, wherein each iteration a new patient case was added (consisting of data from about ~100,000 APBNRR executions) and the network's hyperparameters and architecture were tuned to reduce the loss marked in the previous iteration.

4.3. Deep Learning APBNRR

The deep learning portion of APBNRR takes place before the actual execution of APBNRR. APBNRR utilizes as its input the parameter sets predicted by the deep learning model to result in the lowest Hausdorff distances. A visualization of how the deep learning model works is shown in fig. 4. The neural network is given as input each parameter set in a pool consisting of patient-specific parameter sets, which was produced by augmenting a base, general parameter set pool. We have created tools that do this automatically. The neural network iterates through each parameter set and outputs the Hausdorff distance of the registered image that would be produced by APBNRR if this parameter set was utilized. Out of all those predictions, the lowest ones are compiled in a file, and can then be used as input to APBNRR. This process takes about 15 seconds.

4.3.1 Patient-Specific Parameter Pool. The deep learning model takes as input parameter sets from a pool that is specific to each patient. This is an augmented version of a base, general parameter pool, which has been enhanced to include two patient-specific features, the location of the tumor in the brain, and the size of the deformation as can be derived from the rigid registration error. In our experiments, the patient-specific features have yielded significantly better results.

4.4. Challenges

There were a couple of important challenges we encountered while constructing the deep learning model. The first one is that APBNRR is non-deterministic due to the parallel feature selection algorithm. This means that given two identical parameter sets, APBNRR will yield two different registered images, with different Hausdorff distances. This non-determinism leads to a degree of "randomness" in results, which hinders the ability of the neural network to correctly predict the Hausdorff distances. Unfortunately, there is no practical way of solving this issue, other than amassing more training data to "average out" this randomness, which leads us to the second challenge: collecting training data takes a long time. To gather training data, we must run APBNRR for every parameter set in an exhaustive pool for every additional patient case we want to include in our training set. Evaluating a single case exhaustively (~1 million executions) takes about 1-2 months, depending on the severity and size of the brain tumor.

5. Experimental Evaluation

5.1. Data Set

Our data set for the model consists of medical resonance images from thirteen patient cases provided by the Huashan Worldwide Medical Center and Brigham and Women's Hospital of Harvard University, as well as output data from over 2.6 million executions of APBNRR, collected over the

Case	Type	Tumor Location	Rigid Registration	B-Spline Registration	PBNRR	APBNRR (Default Parameters)	APBNRR (Deep Learning)	APBNRR (Parameter Sweep)
1	PTR	Left frontal lobe	17.00	17.00	16.49	5.69	2.78	2.78
2	PTR	Left frontal lobe	10.59	5.28	10.76	2.30	2.64	1.77
3	PTR	Left parietal lobe	16.15	13.78	15.12	4.60	2.40	1.70
4	PTR (deep)	Left parietal-occipital lobes	29.93	21.34	27.76	2.83	1.84	1.46
5	CTR	Frontal-temporal lobes	25.51	25.18	22.50	4.97	2.64	2.64
6	CTR	Right frontal lobe	5.59	5.59	3.43	3.09	2.78	2.64
7	CTR	Right temporal lobe	17.90	16.94	15.56	4.11	2.94	2.00
8	CTR	Left posterior-temporal lobe	18.85	17.49	17.38	3.57	3.29	3.24
9	CTR	Left frontal lobe	17.14	7.48	15.41	4.25	2.40	1.84
10	ETR (deep)	Left frontal lobe	35.72	27.77	33.57	3.71	2.06	1.77
11	CTR	Left frontal lobe	25.72	25.72	23.90	3.42	2.64	2.30
12	PTR	Right frontal lobe	26.89	15.86	26.89	4.00	2.85	2.21
13	CTR	Left frontal lobe	19.24	14.61	19.89	2.40	2.00	1.48

Table 3. Shows the thirteen patient cases that consist our data set and the results (measured as the Hausdorff distance in mm) achieved with various methods of registration, including with APBNRR using deep learning and APBNRR using a parameter sweep. Cases with numbers 1-11 were used for training, and 12, 13 were used for evaluation.

course of five months. Out of the thirteen cases, eleven were used for training (~2.4 million parameter sets) and two for evaluation (~200,000 parameter sets). APBNRR was executed with arrays of 120 parameter sets on a supercomputing cluster, utilizing in total 450 CPU cores and over 4 TB of RAM. The total compute time that was required for the executions was approximately 3600 hours

5.2. Deep Learning Model Results

Using the deep learning model, we achieved a training root mean squared error (RMSE) of 1.41 and an evaluation RMSE of 1.21 for predicted Hausdorff distances. For further evaluation, utilizing the trained model, we executed APBNRR for the 13 cases again, using the top 120 parameter sets predicted by the model for each case to yield the lowest Hausdorff distances. The best results of those executions are displayed in Table 2, along with the results from several other registration methods as a comparison. The best results from the APBNRR parameter sweeps that were used for generating our dataset are also displayed as a reference of what could have been the lowest Hausdorff distance value.

On average, APBNRR with deep learning is ~8.45 times better than rigid registration, ~6.71 times better than B-Spline registration, and ~7.9 times better than PBNRR, which is an older version of APBNRR. It should also be noted that APBNRR works very well with deep tumors which result in great brain deformation in comparison to the other registration methods, leading to results that are on average ~16.8 times better. Overall, APBNRR with deep learning leads to superior results than any of the registration methods noted above. Choosing the correct parameters for medical image registration is a difficult task, as there are many (usually infinite) possible values and combinations of parameters that can lead to better or worse results. The deep learning portion of the APBNRR framework makes this easier by greatly limiting the set of possible optimal parameters for each individual patient, moving APBNRR one step closer to being able to be utilized in a real-time setting, where registration accuracy and speed are critical.

6. Performance Evaluation

In this section, we will evaluate deep learning APBNRR on how it would perform in a real-world setting. Specifically, we will look at how APBNRR would perform in the brain tumor resection workflow in AMIGO [15], which represents an ideal setting for the usage of APBNRR.

Referring to the AMIGO workflow, APBNRR execution takes place after the intraoperative MR phase, and is meant to assist in tumor and residual tumor assessment due to the ability of APBNRR to perform tumor resection. The registered intraoperative image with the tumor resected, allows the neurosurgeon to better evaluate how well the tumor has been resected. The time window for APBNRR to execute as part of this workflow is a few minutes. As such, on top of accuracy, speed is also critical.

A key element for evaluating the performance of APBNRR is determining optimal input parameters. Optimal input parameters are usually in the first 800 – 39,000 parameter sets in our base parameter pool. Determining optimal input parameters by doing a parameter sweep utilizing arrays of 120 jobs would take an estimated 33 minutes – 27 hours based on time data collected from previous executions with an average APBNRR execution of 5 minutes. This high variability is not acceptable in a neurosurgery session. Deep learning offers very significant speedups in this area, by allowing the user to limit the parameter pool to a custom value. For example, by utilizing deep learning, we could reduce the number of parameter sets that would need to be evaluated to 120 (the size of the job array mentioned before), potentially allowing APBNRR to finish executing in 5 minutes while simultaneously yielding good results, which is much more acceptable.

Deep learning solves the problem of limiting the parameter pool that needs to be evaluated, but there are still some issues with APBNRR that need to be resolved. One of the most important is the dependence of APBNRR performance on the size of the tumor. The larger the tumor, the more time it takes for APBNRR to execute. Furthermore, the more adaptive iterations APBNRR goes through, the longer the execution. We have seen an execution time ranging from ~5 to ~30 minutes, with parameter sets having more adaptive iterations slower execution than those with lower ones, but better registration accuracy on average.

Finally, it should be noted that the overall performance of deep learning APBNRR is dependent on the amount of computational resources available. In particular, the more computational resources available, the larger the parameter pool of the best parameter sets predicted by the neural network can be. As a result, there is a greater chance of achieving better accuracy and performance.

7. Summary

The accurate geometrical representation of the brain for modelling deformation during glioma resection is a difficult problem that can be solved with APBNRR. Deep learning can improve the performance of APBNRR towards solving this problem, by predicting a pool of input parameters that can lead to the most optimal representations, and ultimately, to more accurate, and faster image registration that is one step closer to being able to be used in real-world scenarios.

8. Future Work

Deep learning APBNRR takes us a step closer to enabling APBNRR to be used in real-world scenarios. However, there are still some issues to be resolved before that can happen. First, more training data needs to be collected to allow the deep learning model to offer more accurate predictions. Second, work needs to be done to enable the deep learning model to generate a parameter pool that is not only limited in size but can also be evaluated rapidly. Finally, the performance of APBNRR needs to be further improved, especially the performance of APBNRR regarding deep tumors which involve very large brain deformation.

Acknowledgements

We would like to acknowledge Dr. Fotis Drakopoulos and the authors of the media paper for their work on PBNRR and APBNRR.

References

[1] F. Drakopoulos et al. Toward a real-time multi-tissue Adaptive Physics-Based Non-Rigid Registration framework for brain tumor resection. Frontiers in Neuroinformatics, 11(8), 2014.

[2] J. Ker et al. Deep Learning Applications in Medical Image Analysis. IEEE Access. 6(1):9375-9389, 2018.

[3] G. Balakrishnan et al. An Unsupervised Learning Model for Deformable Medical Image Registration. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 9252-9260, 2018.

[4] R. Arora et al. Understanding Deep Neural Networks with Rectified Linear Units. arXiv. 2018

[5] S. Ruder. An overview of gradient descent optimization algorithms. arXiv. 2017

[6] F. Chollet. Keras. https://keras.io/. 2015.

[7] Google Inc. TensorFlow. https://www.tensorflow.org/. 2015.

[8] N. Keskar. Improving Generalization Performance by Switching from Adam to SGD. arXiv. 2017.

[9] Liu Y, Kot A, Drakopoulos F, Yao C, Fedorov A, Enquobahrie A, Clatz O, Chrisochoides NP. An ITK implementation of a physics-based non-rigid registration method for brain deformation in image-guided neurosurgery. Frontiers in Neuroinformatics. 8(33), 2014.

[10] Foteinos PA, Chrisochoides NP. High quality real-time image-to-mesh conversion for finite element simulations. Journal of Parallel and Distributed Computing. 74(2):2123-2140, 2014,

[11] Bathe KJ. 1996. Finite element procedures. New Jersey: Prentice-Hall.

[12] Clatz O, Delingette H, Talos IF, Golby A, Kikinis R, Jolesz F, Ayache N, Warfield S. Robust non-rigid registration to capture brain shift from intraoperative MRI. Medical Imaging, IEEE Transactions on, 24(11):1417-1427, 2005.

[13] Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, der Vorst HV. Templates for the solution of linear systems: building blocks for iterative methods. 2nd ed. Philadelphia (PA): SIAM, 1994.

[14] Brigham and Women's Hospital. Advanced Multimodality Image Guided Operating (AMIGO) Suite. https://www.brighamandwomens.org/research/amigo/advanced-multimodality-image-guided-operating-suite

[15] The Ferenc Jolesz National Center for Image Guided Therapy, Brigham and Women's Hospital. Brain Tumor Resection Workflow in AMIGO. https://ncigt.org/brain-tumor-resection

[16] J. Krebs et al. Unsupervised Probabilistic Deformation Modeling for Robust Diffeomorphic Registration. arXiv. 2018.