ATHENA: a GPU-based Framework for Biomedical 3D Rigid Image Registration

Giuseppe Sorrentino, Marco Venere, Eleonora D'Arnese, Davide Conficconi, Isabella Poles, Marco Santambrogio
Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy
{giuseppe.sorrentino, marco.venere}@mail.polimi.it
{eleonora.darnese, davide.conficconi, isabella.poles, marco.santambrogio}@polimi.it

***Abstract*—3D image registration is one of the most complex algorithms employed in medical imaging applications. Software solution struggles to reach high accuracy in a reasonable time, therefore this work presents ATHENA, a framework for rigid 3D Image Registration, exploiting heterogeneous architecture for acceleration, and also providing support for memory-constrained devices. Moreover, ATHENA presents a tool for automatically generating misalignent between volumes, to perform a robustness analysis on different kinds of distortions. We compared ATHENA with SimpleITK, a well-known software library, and with a software version of the proposed algorithm, achieving a top speedup of 18.1× and 53.7× respectively.**

***Index Terms*—3D Image Registration, GPU, Acceleration**

I. Introduction

Nowadays, an increasing number of companies [1]–[7] give their contribution to find innovative and fast solutions for very complex algorithms employed in biomedical applications [8], [9]. Among the huge spectrum of applications, the ones related to medical image analysis are the most explored due to the growing number of use case scenarios, such as disease diagnosis [7] and robotic aided surgery [10]. In particular, given the limitations of software solutions, researchers investigated new architectural and algorithmic alternatives employing heterogeneous systems to accelerate compute-bound procedures.

One of the most computationally expensive applications is Image Registration (IR), which involves aligning two images to fix their relative distortion. This task can be categorized into *rigid/affine* registration and *deformable* registration [11]. The former handles a subset of possible deformations since it exploits transformation that preserve the parallelism of lines. Conversely, the latter can correct all kinds of transformations and mainly exploits deep learning techniques [12]. However, the two are not completely unrelated, since deformable registration exploits, as its initial step, the rigid one. Thus, the acceleration of rigid IR will also benefit the deformable one [13]. Additionally, depending on the acquisition technique, we can also distinguish between mono-modal and multi-modal IR. The former registers images acquired in different time instants by the same protocol and sensor type. The latter, common in clinical practice, deals with heterogeneous image types such as Computed Tomography (CT), Positron Emission Tomography (PET), or Magnetic Resonance Imaging (MRI) [14].

Since most medical images, like CTs and PETs, are acquired as 2D slices stacked in a 3D volume, 2D IR focuses on registering a couple of images at a time. Conversely, 3D IR also considers information about the volume itself, thus dealing with all the 2D slices as a single 3D object. More specifically, 3D IR aligns a *floating* volume against a *reference* to fix distortion.

In the literature of 3D image registration there are two main trends: software-based and hardware-based solutions. Software frameworks rely entirely on software libraries, while accelerated frameworks use hardware acceleration to achieve higher performance. On this path, GPUs have proven effective in accelerating image registration procedures [13], [15]–[18].

For these reasons, this work presents ATHENA, an open-source 3D IR framework for multi-modal rigid procedure exploiting heuristic procedures and heterogeneous architecture to enhance performance. As a similarity metric for the employed heuristic algorithm, ATHENA uses mutual information, which already proved effective to achieve remarkable results in accuracy [19]. Our main contributions are:

- An open-source framework[1] to perform 3D IR on heterogeneous architecture, achieving state-or-the-art accuracy (Section IV-A).
- A novel methodology to exploit such a complex procedure on memory-constrained devices, reaching the same accuracy as less constrained architecture (Section IV-B).
- A tool for the automatic generation of misaligned volumes, to perform robustness analysis while exploring the space of possible deformations (Section IV-C).

Following, we present the background of 3D IR (Section II) and a comparison with other related work (Section III) as groundwork to deeply understand the ATHENA's design process (Section IV) and results (Section V).

II. Background on Image Registration

This Section describes the main theoretical concepts behind 3D rigid image registration (IR), to provide the groundwork for fully understanding the development process. Rigid IR can be carried out using landmark points or voxels intensity [11].

[1]https://github.com/necst/athena

DOI 10.1109/BioCASXXXXX.2023.XXXXXXX

While the former requires some a priori knowledge of the image features, the latter does not and, therefore, is more general and commonly employed. Moreover, the IR is structured as an iterative heuristic process aiming at maximizing a similarity metric (Sim) among the *reference* (R) and the *floating* (F) volume, after the application of the transformation T to F (F^T), and giving, as output, the optimal parameters for *floating* volume transformation ($\hat{T}$) as in (Equation (1)) [20].

$$\hat{T} = \arg\max_{T} Sim(R, F^T) \tag{1}$$

The rigid IR algorithm can be divided into three main phases: the parameter search, involving the use of optimizers to explore the parameters space; the transformation, which exploits the parameters discovered in the previous phase to transform the *floating* image; and the similarity metric computation, used to evaluate the correctness of the transformation. This similarity metric allows convergence to the solution providing the best parameters to align the *floating* volume to the *reference*.

Depending on the desired result, the three phases can be customized to create several configurations of transformations, metrics, and optimizers.

III. Related Work

Current approaches dealing with 3D image registration cluster into two main categories. On the one hand, there are entirely software-based solutions aiming at performing registration, looking for the best accuracy-performance trade-off. These solutions can be closed-source, like MATLAB [21], which require licenses and achieve very high results suffering low flexibility and customizability, or open-source solutions, like SimpleITK [22] and ANTs [23], that achieve sufficient accuracy but far from negligible execution times.

On the other hand, none of these algorithms achieves perfect accuracy since it would be unfeasible in terms of execution time. Consequently, other solutions exploit hardware acceleration to enable more accurate algorithms while keeping low execution time [16], [24]–[26]. On this path, GPUs proved very effective in accelerating IR algorithms. While many solutions performed approximations to achieve significant results [26], D'Arnese et al. [16] developed a framework to accelerate 2D IR by employing heterogeneous architecture without reducing accuracy and reaching remarkable performance. However, they only focus on 2D IR, which does not take advantage of the information about the overall 3D volume. Additionally, 2D computations do not exploit all the available hardware resources and memory bandwidth, thus suggesting that higher performance can be obtained with the more general 3D IR.

In this sense, recognizing the effort of the approach proposed in [16], we present ATHENA, a framework exploiting heterogeneous architectures to accelerate multi-modal 3D rigid image registration, reaching state-of-the-art accuracy and significantly reducing execution time compared to the software. Differently from others, ATHENA uses information about the volume to register highly misaligned images, thus fixing problems of existing solutions in the state-of-the-art.

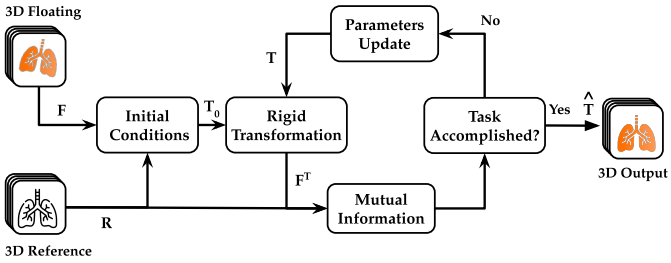

Fig. 1. High-level scheme of the 3D Image Registration heuristic.

IV. Proposed Approach and Optimization

ATHENA's solution performs 3D image registration with a GPU-accelerated framework. This Section presents the software design (Section IV-A), proposes a specific strategy to obtain high accuracy even on memory-constrained devices (Section IV-B), and devise a generator of misalignment between volumes, that will be used to evaluate the registration robustness to different degrees of distortion between the *reference* and *floating* volumes (Section IV-C).

A. 3D Image Registration Framework

We developed our framework in Python while using the PyTorch library [27] for image processing and Kornia [28] for rigid transformations. Indeed PyTorch allows for using both CPU and GPU with the same programming interface while offering support for multithreading, especially for tensorial operations. The software pipeline that performs 3D image registration receives as input two volumes, the *reference* and the *floating*, represented as two stacks of 2D images (as the volumes are split on the z axis). Its goal is to apply a transformation to the *floating* volume that maximizes the chosen similarity metric (i.e., mutual information) with respect to the *reference* volume. Therefore, it executes a search into the transformation parameters space, as Figure 1 shows.

As described in [29], a correct search highly depends on good initial conditions. Generally, 3D moments between volumes are a rigorous way to find such starting parameters, yet their computation appears highly inefficient [30]. Therefore, we introduce a lightweight computation of 3D moments that focus on the three main degrees of freedom considered for biomedical cases: the translation on the axes x and y, belonging to the plane orthogonal to the acquisition direction (z axis), and the rotation around z. We reduce the 3D moments' computation to average the 2D moments' parameters between the 2D images of the input stacks. Since it is computationally demanding, we offload its evaluation onto GPUs.

After the initial condition computation, an optimizer searches for the parameters that maximize the similarity metric between the *reference* and the *floating* volumes by considering translations and rotations over the three axes. However, specific use cases may allow for avoiding some negligible parameters. We implemented two different optimizers: 1+1 Evolutionary and Powell. The former employs an evolutionary

strategy of parent-child to optimize all the parameters simultaneously and stops when converging or after a maximum number of iterations. The latter instead optimizes each parameter at a time and converges based on a hyperparameter threshold. According to profiling analysis, these algorithms' most computationally expensive parts are the evaluation of the mutual information and the transformation of the *floating* volume. Therefore, we accelerate them by exploiting GPUs. Indeed, we load the volumes onto the GPU RAM, perform volume transformation and computation of the mutual information, and then retrieve the output. To reduce back and forth CPU-GPU data transfer we move the transformation matrix only. Furthermore, to reduce the execution time, we precompute the mutual information piece of the reference volume, being constant throughout the procedure. Moreover, we further reduce Powell's execution time by evaluating, at each iteration, two volume transformations and their corresponding mutual information concurrently. After the computation of the final parameters, they are applied to the floating volume to align it.

B. Strategy for Memory-Constrained Devices

The main bottleneck of the IR procedure in Section IV-A is the amount of memory available on the GPU devices. Indeed, since we transform 3D volumes and compute the mutual information on GPU, we need the whole reference and floating volumes to fit the available GPU memory, along with the additional data structures used by the underlying libraries. This prevents, in many cases, the adoption of low- or middle-end memory-constrained devices.

Therefore, we devised an additional strategy to tailor IR to such a class of devices: we sample a portion of the input volumes and search for the optimal registration parameters on such subvolumes. Once we obtain the final parameters, we apply them subvolume-wise to the input *floating* volume. Doing so, the amount of data that needs to be loaded onto the devices does not exceed the available memory. Indeed, we sample a restricted number of 2D slices based on the system resources, but other strategies may be explored according to an apriori knowledge of the volume's information distribution.

Once the algorithm computes the optimal transformation, we divide the floating volume into subvolumes, and apply the final transformation parameters on each subvolume separately, since memory-constrained hardware cannot apply a transformation on the whole volume. Differently from unconstrained devices that can transform the entire volume with a single transformation, the mechanism for memory-constrained ones requires to split the volume into subvolumes, move them to GPU, and transform them separately. This procedure increases the transfer time introducing a delay in the last transformation compared to the one performed on unconstrained boards.

C. Generator of Misalignment for Robustness Evaluation

We also developed a tool to generate and apply specific misalignment degrees to a 3D volume. Indeed, we aimed to estimate the robustness of the registration task compared to the misalignment between the *reference* and the *floating* volumes.

TABLE I
TARGET MACHINE CONFIGURATIONS WITH THEIR UNIQUE CODE NAME.

Code Name	CPU	Machine RAM	NVIDIA GPU	GPU RAM
A5000	AMD Ryzen 7 5800X	32GB	RTX A5000	24GB
GTX1650	Intel i7-10870H	16GB	GTX 1650 Ti	4GB
GTX1050	Intel i7-7700HQ	16GB	GTX 1050	4GB
GTX960	Intel i7-6700	32GB	GTX 960	2GB
V100	Intel Xeon Platinum 8167M	88GB	Tesla V100	32GB
Ryzen7	AMD Ryzen 7 5800X	32GB	-	-

The misalignment can be described in translations and rotations over x, y, and z. Our generator takes an initial input volume, randomly applies transformations, and ATHENA performs registration between the initial volume and the transformed one. Then, it evaluates the registration accuracy and considers it optimal if it overcomes a specific threshold. The generator applies transformations that only modify one degree of freedom at a time (e.g., a translation or a rotation) with increasing values until the IR quality exceeds the threshold, providing quantitative interval ranges for the variables.

V. EXPERIMENTAL EVALUATION

For evaluating ATHENA, we consider two main classes of devices: high-end GPUs, which can search the parameters over the whole volumes, and memory-constrained GPUs, for which we apply the procedure explained in Section IV-B. Then, we compare ATHENA's performance and accuracy against a software version of the proposed algorithm and SimpleITK. Finally, we provide results for the robustness analysis based on the generator of Section IV-C. For clarity, in Table I we show the hardware configurations used in testing, together with the codename they are referred to throughout this Section.

A. Performance Evaluation: Execution Time and Accuracy

Since we exploit a novel mechanism to perform 3D IR even on memory-constrained devices, performance speedup is evaluated in Frame per Second (FPS) exploited for searching the optimal parameters. Moreover, to reach the highest accuracy, each configuration uses the maximum number of slices that fits the available GPU resources. Indeed, the qualitative analysis in Figure 2 shows the difference between the overlap achieved between the ground truth and ATHENA, and SimpleITK.

Figure 3 and Figure 4 showcase the speedup against ATHENA's software version with 1+1 and Powell, respectively, as optimizers. All the data in Figure 3 and Figure 4 does not account for time needed to apply the final transformation to the *floating* volume since, for the low-end device, it introduces an overhead, while Table II reports the overall execution time, comprehensive of the final transformation.

Figure 3 shows that even memory-constrained devices achieve massive speedups over our software version, specifically with a top speedup of $15.1\times$ (on low-end boards) and $53.7\times$ (on high-end ones) with 1+1. Figure 4, instead, shows that SimpleITK suffers Powell's complexity, achieving a speedup of $1.88\times$, while ATHENA still achieves significant speedup over our software version, with a top speedup of $8.99\times$ and $33.7\times$, on low- and high-end devices respectively.

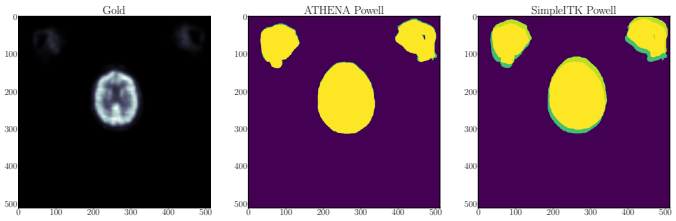

Fig. 2. Visual comparison of the overlapping achieved with ATHENA's A5000 and SimpleITK Ryzen7

.

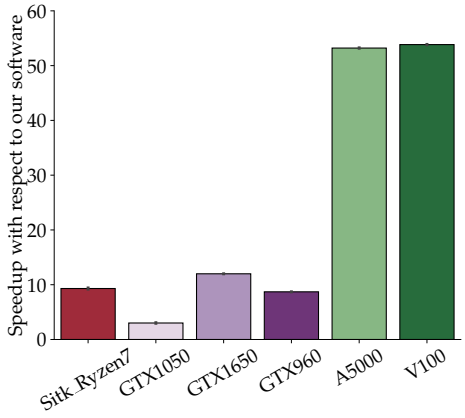

Fig. 3. Speedup against our software, exploiting 1+1 optimizer.

Table II, along with the overall the execution time, includes information about the accuracy, measured as Intersection over Union (IoU), for each solution. We can see how all GPU-based configurations are significantly more accurate than SimpleITK, reaching up to 0.986 IoU on memory-constrained devices. Moreover, Table II allows appreciating that the employed mechanism has no negative impact on accuracy, reaching comparable results than high-end devices.

Finally, almost all configurations are significantly faster than SimpleITK, with a top speedup of 18.1× and 15.8× using Powell optimizer on ATHENA's V100 and A5000.

B. ATHENA *Robustness Analysis*

The procedure defined in Section IV-C aims to evaluate the quality of the output of the registration procedure based on the intensity of the misalignment between the *floating* and the *reference* volumes. We described such misalignment in terms of translation on x and y axes and rotation around z axis since they are the most representative for our biomedical use case. From now on, we refer to these parameters as t_x, t_y, and ψ, respectively. For each of them, we select an interval to test. Table III displays the results obtained on A5000 since it is a good trade-off low-end and newer high-end devices. To evaluate the registration quality, we define three different ranges: *optimal*, for IoU ≥ 0.85, *acceptable*, for $0.75 \leq IoU < 0.85$, and *not acceptable*, for IoU < 0.75.

Therefore, we selected one parameter per time and applied the corresponding transformations. For t_x and t_y, we tested the

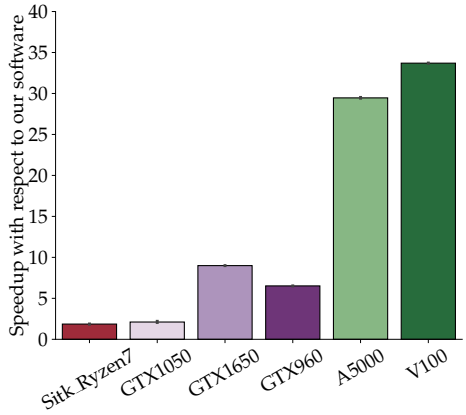

Fig. 4. Speedup against our software, exploiting Powell's optimizer.

TABLE II
VOLUME DIMENSION, EXECUTION TIME $\mu(\sigma)$, AND IOU FOR DIFFERENT ATHENA'S CONFIGURATIONS, SIMPLEITK, AND OUR SOFTWARE.

Work	Device	Optimizer	#Volume Slices	Execution Time [s]	IoU
ATHENA	GTX960	1+1	80	95.50 (0.08)	0.876 (0.079)
		Powell	80	25.43 (0.03)	0.907 (0.070)
	GTX1650	1+1	190	164.8 (2.74)	0.981 (0.015)
		Powell	160	36.81 (1.44)	**0.988 (0.006)**
	GTX1050	1+1	180	716.24 (69.42)	0.977 (0.019)
		Powell	140	128.67 (15.62)	0.981 (0.008)
	A5000	1+1	246	47.859 (0.099)	0.983 (0.008)
		Powell	246	16.97 (0.08)	0.982 (0.008)
	V100	1+1	246	47.296 (0.086)	0.982 (0.007)
		Powell	246	**14.836 (0.017)**	0.985 (0.008)
SimpleITK	Ryzen7	1+1	246	273.855 (8.542)	0.861 (0.060)
		Powell	246	269.063 (4.344)	0.873 (0.024)
Software	Ryzen7	1+1	246	2550.418 (94.4216)	0.991 (0.060)
		Powell	246	499.952 (1.224)	0.993 (0.024)

range $(-200px; +200px)$, with a $10px$ step, while for rotations, we tested values of $\psi \in (-25°; +25°)$, with a $0.5°$ step. For each variable, we show in Table III the interval ranges for *acceptable* and *optimal* performance. The robustness analysis proves that ATHENA optimally corrects translation of almost 30% of the image dimension and rotations of 8°, which is remarkable given the physical constraints of our use case.

TABLE III
ACCEPTABLE AND OPTIMAL RANGES FOR MISALIGNMENT VARIABLES.

Var.	Acceptable Rng. IoU ∈ [0.75,0.85)	Optimal Rng. IoU ∈ [0.85,1]
t_x	$(-140px; +200px)$	$(-100px; +150px)$
t_y	$(-200px; +180px)$	$(-190px; +150px)$
ψ	$(-11°; +11°)$	$(-8°; +8°)$

VI. CONCLUSIONS AND FUTURE WORK

This work introduced ATHENA, a framework for 3D rigid image registration exploiting hardware acceleration through heterogeneous devices. Compared with the state-of-the-art, ATHENA presents better results in accuracy and execution time for almost all configurations, by also introducing a novel mechanism to use memory-constrained devices.

Future work. We will extend ATHENA to other optimizers and metrics to reach software-library flexibility.

ACKNOWLEDGMENT

Data used in this publication were generated by the National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC). We thank Oracle Cloud Infrastructure and Oracle for Research and the NVIDIA Academic Hardware Grant Program for their support to this work.

REFERENCES

[1] Hati International, "The massive interest of non-healthcare companies in the healthcare industry," https://hatiintl.com/blog/the-massive-interest-of-non-healthcare-companies-in-healthcare-industry, 2019.

[2] J. Fowers, K. Ovtcharov, M. Papamichael, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman, L. Adams, M. Ghandi *et al.*, "A configurable cloud-scale dnn processor for real-time ai," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2018, pp. 1–14.

[3] E. Chung, J. Fowers, K. Ovtcharov, M. Papamichael, A. Caulfield, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman *et al.*, "Serving dnns in real time at datacenter scale with project brainwave," *IEEE Micro*, vol. 38, no. 2, pp. 8–20, 2018.

[4] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[5] C. N. Takahashi, B. H. Nguyen, K. Strauss, and L. Ceze, "Demonstration of end-to-end automation of dna data storage," *Scientific reports*, vol. 9, no. 1, p. 4998, 2019.

[6] B. W. Bögels, B. H. Nguyen, D. Ward, L. Gascoigne, D. P. Schrijver, A.-M. Makri Pistikou, A. Joesaar, S. Yang, I. K. Voets, W. J. Mulder *et al.*, "Dna storage in thermoresponsive microcapsules for repeated random multiplexed data access," *Nature Nanotechnology*, pp. 1–10, 2023.

[7] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado *et al.*, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nature medicine*, vol. 25, no. 6, pp. 954–961, 2019.

[8] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 574–584.

[9] Y. Tang, D. Yang, W. Li, H. R. Roth, B. Landman, D. Xu, V. Nath, and A. Hatamizadeh, "Self-supervised pre-training of swin transformers for 3d medical image analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 730–20 740.

[10] S. E. Salcudean, H. Moradi, D. G. Black, and N. Navab, "Robot-assisted medical imaging: A review," *Proceedings of the IEEE*, 2022.

[11] L. G. Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.

[12] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: A learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, aug 2019. [Online]. Available: https://doi.org/10.1109

[13] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley, "A survey of medical image registration on multicore and the GPU," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 50–60, 2010.

[14] Y. Hu, M. Modat, E. Gibson, W. Li, N. Ghavami, E. Bonmati, G. Wang, S. Bandula, C. M. Moore, M. Emberton, S. Ourselin, J. A. Noble, D. C. Barratt, and T. Vercauteren, "Weakly-supervised convolutional neural networks for multimodal image registration," *Medical Image Analysis*, vol. 49, pp. 1–13, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841518301051

[15] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley, "Parallel computation of mutual information on the gpu with application to real-time registration of 3d medical images," *Computer methods and programs in biomedicine*, vol. 99, no. 2, pp. 133–146, 2010.

[16] E. D'Arnese, E. Del Sozzo, D. Conficconi, and M. D. Santambrogio, "Exploiting heterogeneous architectures for rigid image registration," in *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2021, pp. 1–5.

[17] P. Bhosale, M. Staring, Z. Al-Ars, and F. F. Berendsen, "Gpu-based stochastic-gradient optimization for non-rigid medical image registration in time-critical applications," in *Medical Imaging 2018: Image Processing*, vol. 10574. International Society for Optics and Photonics, 2018, p. 105740R.

[18] Y. Liu, Y. Zhou, Y. Zhou, L. Ma, B. Wang, and F. Zhang, "Accelerating sar image registration using swarm-intelligent gpu parallelization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5694–5703, 2020.

[19] J. Pluim, J. Maintz, and M. Viergever, "Mutual-information-based registration of medical images : a survey," *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[20] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[21] I. MathWorks, "Image processing toolbox," 1994-2020. [Online]. Available: https://mathworks.com/products/image.html

[22] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, "The design of simpleitk," *Frontiers in neuroinformatics*, vol. 7, p. 45, 2013.

[23] B. B. Avants, N. Tustison, G. Song *et al.*, "Advanced normalization tools (ants)," *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.

[24] D. Conficconi, E. D'Arnese, E. Del Sozzo, D. Sciuto, and M. D. Santambrogio, "A framework for customizable fpga-based image registration accelerators," in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2021, pp. 251–261.

[25] E. D'Arnese, D. Conficconi, E. Del Sozzo, L. Fusco, D. Sciuto, and M. D. Santambrogio, "Faber: A hardware/software toolchain for image registration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 1, pp. 291–303, 2022.

[26] K. Ikeda, F. Ino, and K. Hagihara, "Efficient acceleration of mutual information computation for nonrigid registration using cuda," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 956–968, 2014.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[28] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: an open source differentiable computer vision library for pytorch," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3674–3683.

[29] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artificial intelligence review*, vol. 55, no. 1, pp. 291–322, 2022.

[30] J. Flusser, T. Suk, and B. Zitová, *2D and 3D image analysis by moments*. John Wiley & Sons, 2016.