IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 7, 2017, accepted September 12, 2017, date of publication September 26, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2755863

A GPU-Accelerated Deformable Image Registration Algorithm With Applications to Right Ventricular Segmentation

KUMARADEVAN PUNITHAKUMAR[1,2,3], (Senior Member, IEEE), PIERRE BOULANGER[1,2,3], AND MICHELLE NOGA[1,2]

[1]Department of Radiology and Diagnostic Imaging, University of Alberta, Edmonton, AB T6G 2R3, Canada
[2]Servier Virtual Cardiac Centre, Mazankowski Alberta Heart Institute, Edmonton, AB T6G 2B7, Canada
[3]Department of Computing Science, University of Alberta, Edmonton, AB T6G 2R3, Canada

Corresponding author: Kumaradevan Punithakumar (punithak@ualberta.ca)

This work was supported in part by Servier Canada Inc., in part by academic hardware donations from NVIDIA Corporation, and in part by the Research Leadership Fund from the Heart and Stroke Foundation of Alberta, NWT and Nunavut, and Mazankowski Alberta Heart Institute.

ABSTRACT Delineation of the cardiac right ventricle is essential in generating clinical measurements such as ejection fraction and stroke volume. Given manual segmentation on the first frame, one approach to segment right ventricle from all of the magnetic resonance images is to find point correspondence between the sequence of images. Finding the point correspondence with non-rigid transformation requires a deformable image registration algorithm, which often involves computationally expensive optimization. The central processing unit (CPU)-based implementation of point correspondence algorithm has been shown to be accurate in delineating organs from a sequence of images in recent studies. The purpose of this study is to develop computationally efficient approaches for deformable image registration. We propose a graphics processing unit (GPU) accelerated approach to improve the efficiency. The proposed approach consists of two parallelization components: Parallel compute unified device architecture (CUDA) version of the deformable registration algorithm; and the application of an image concatenation approach to further parallelize the algorithm. Three versions of the algorithm were implemented: 1) CPU; 2) GPU with only intra-image parallelization (sequential image registration); and 3) GPU with inter and intra-image parallelization (concatenated image registration). The proposed methods were evaluated over a data set of 16 subjects. CPU, GPU sequential image, and GPU concatenated image methods took an average of 113.13, 16.50, and 5.96 s to segment a sequence of 20 images, respectively. The proposed parallelization approach offered a computational performance improvement of around 19× in comparison to the CPU implementation while retaining the same level of segmentation accuracy. This paper demonstrated that the GPU computing could be utilized for improving the computational performance of a non-rigid image registration algorithm without compromising the accuracy.

INDEX TERMS Image registration, GPU computing, cardiac functional analysis, image segmentation, magnetic resonance imaging.

I. INTRODUCTION

Image registration is the process of obtaining a mapping between a pair of images, one of the most common applications in medical image processing [1]–[3]. During the registration process, one image known as the reference image is considered fixed and the other is moving and goes through a number of image operations such as transformation and regularization. The process of obtaining the alignment involves finding the transformation that maximizes the similarity between the reference and moving images. The final solution to the image registration approach is often achieved via many iteration loops which lead to high computational cost. Traditional single central processing unit (CPU) based implementations take a considerable amount of time to compute the registration, and therefore, limit the application for standard clinical applications. One approach for improving the performance of the registration algorithm is to use field-programmable gate array (FPGA) [4]. However, the utilization of FPGAs has been restricted by its development time and complexity for implementation.

During the past few decades, the computing ability of new heterogeneous parallel hardware such as graphics processing units (GPUs) has increased much faster than its CPU counterpart. The peak computational performance and memory bandwidth of recent GPUs exceed CPU by a large factor. Recently, there is a significant research effort to utilize GPUs to improve the computational performance of image processing algorithms [5]–[9]. Several studies have also been proposed in literature to improve the computational performance of image registration algorithms [10], [11]. In contrast to FPGAs, GPUs are less expensive and require less development time. Although recent GPU software development frameworks allow for scientific computing through general purpose graphics processing computing (GPGPU) platforms such as Compute Unified Device Architecture (CUDA) by NVIDIA Corporation, the process of developing efficient massively-parallel versions of the image processing algorithms poses a number of challenges to the scientific community. The major challenge arises from the GPU's architecture and programming model which differs radically from that of a CPU, and the traditional way of developing algorithms often results in poor performance.

This study proposes a parallel implementation of a deformable image registration algorithm using GPU computing. Recent studies have shown that the CPU implementation of the algorithm to be effective in delineating endocardium and epicardium of the right ventricle (RV) [12] as well as analyzing cardiac left ventricular regional function from MRI [13]. The registration algorithm consists of a number of computationally expensive components such as the computation of transformation, calculation of similarity metric and optimization. The registration is performed sequentially over the image series so that the difference between the reference and moving images is smaller which leads to faster convergence. To reduce the accumulated propagation errors arising from the sequential registration process, the images in the series are registered in the forward and backward directions, and a weighted average is used in computing the final transformation mapping for each image in the cardiac cycle.

Parallelization of several components of the registration algorithm is straightforward since these components perform same arithmetic operations for every pixel in the image. In this study, we propose two parallelization components: 1) Parallel implementation of the elements of the registration algorithm using CUDA; and 2) Using image concatenation to parallelize the registration of the image sequence. One of the performance bottlenecks in using the graphics processors is caused by the data transfers between the CPU and GPU memory locations. The transfer rate between these two memory locations is considerable low. By moving the entire images in a sequence to the graphics memory, the proposed approach was able to minimize the data transfer and utilize the very high bandwidth available between the GPU memory and its arithmetic unit.

Due to its complex morphology and function, RV segmentation over a sequence of MR images is acknowledged more challenging than left ventricular segmentation [14], [15]. Automating the process of segmenting RV from a sequence of MRI images has bestirred a significant research attention recently, and has been the subject of a Medical Image Computing and Computer Assisted Intervention (MICCAI) 2012 RV Segmentation Challenge (RVSC) [16]–[19]. The proposed method was evaluated over a data set of 16 subjects and took an average of 5.96 ± 0.91 seconds to segment RV from a sequence of 20 MRI images, a significant performance improvement over serial image registration approach. A preliminary conference version of this work has appeared in IEEE EMBC 2015 [20]. This journal version expands on [20] with a much broader, more informative/rigorous discussion of the subject. This includes a more detailed algorithmic description of the moving mesh generation and quantitative evaluations of the accuracy of the GPU and CPU implementations.

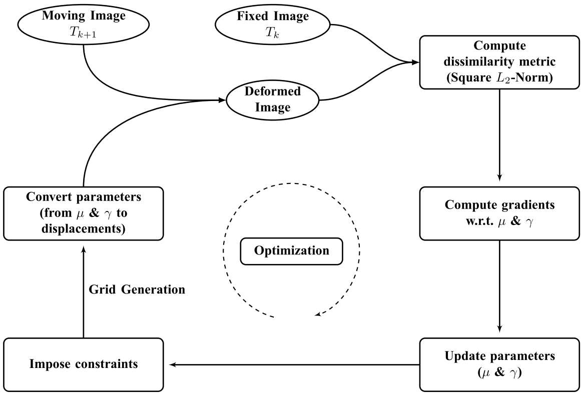

Moving Image T_{k+1}
Fixed Image T_k
Compute dissimilarity metric (Square L_2-Norm)
Deformed Image
Convert parameters (from μ & γ to displacements)
Optimization
Compute gradients w.r.t. μ & γ
Grid Generation
Impose constraints
Update parameters (μ & γ)

FIGURE 1. Computation of moving mesh correspondences between a pair of fixed and moving images.

II. METHODS

The objective of this study is to accelerate the diffeomorphic non-rigid image registration algorithm [21] which computes point-to-point mapping between two images using the squared L_2-norm as the similarity metric [1]. Fig. 1 shows the flow diagram of the process of finding point correspondences between two consecutive images T_k and T_{k+1} (for $k = 1, \ldots, K-1$) where k is the frame number of the image and K is the total number of frames in a cardiac cycle. Finding the optimum mapping between images T_k and T_{k+1} can be formulated as

$$\hat{\phi} = \arg\min_{\phi} E_{L2}(T_k, T_{k+1}, \phi(\xi)) \qquad (1)$$

where $\xi \in \Omega$ denotes pixel location, $\phi : \Omega \rightarrow \Omega$ a transformation function and $E_{L2}(\cdot)$ is the squared L_2-norm based dissimilarity measure [1]. As this problem may not have a unique solution and require more constraints, we introduce

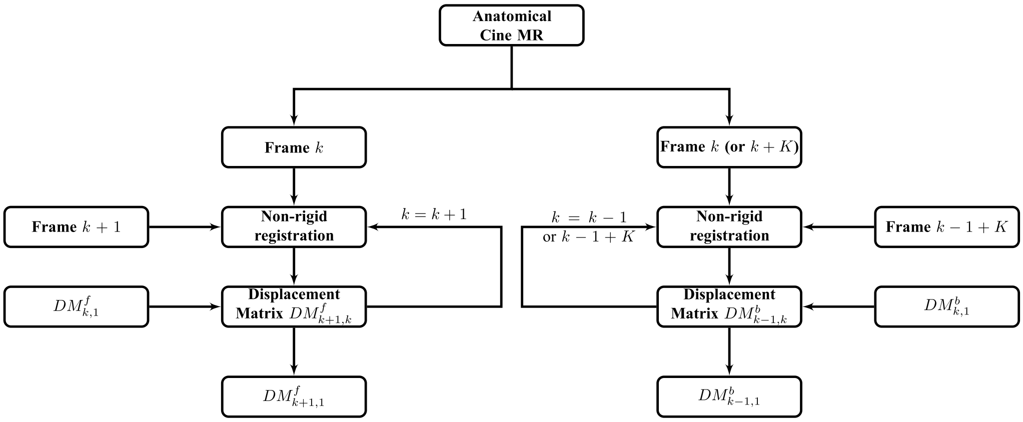


FIGURE 2. **The proposed forward-backward propagation of the moving mesh correspondences for a sequence of cine MR images.**

in the following a deformation field using a monitor function μ and curl of end velocity field γ, where $\mu : \Omega \to \mathbb{R}$ and $\gamma : \Omega \to \mathbb{R}$. Define a continuous monitor function $\mu(\xi)$ constrained by:

$$\int_\Omega \mu = |\Omega|. \tag{2}$$

The objective is to find a transformation ϕ that satisfies,

$$J_\phi(\xi) = \mu(\xi), \tag{3}$$

where J_ϕ is the Jacobian determinant of the transformation. First, we compute a vector field $\rho(\xi)$, defined by

$$\text{div}\ \rho(\xi) = \mu(\xi) - 1. \tag{4}$$

Then, we construct a velocity vector field from $\rho(\xi)$:

$$v_t(\xi) = \frac{\rho(\xi)}{t + (1-t)\mu(\xi)}, \quad t \in [0, 1], \tag{5}$$

where t is an artificially algorithmic time. Finally, we solve the following ODE to obtain ϕ that satisfies (3):

$$\frac{d\psi(\xi, t)}{dt} = v_t(\psi(\xi, t)), \quad t \in [0, 1],\ \psi(\xi, t = 0) = \xi \tag{6}$$

and setting $\phi(\xi) = \psi(\xi, t = 1)$.

We solve the div-curl system under the Dirichlet boundary condition by adding to (4) a constraint on the curl of $\rho(\xi)$, which guarantees obtaining a unique solution:

$$\begin{cases} \text{div}\ \rho(\xi) = \mu(\xi) - 1 & \text{(7a)} \\ \text{curl}\ \rho(\xi) = \gamma(\xi) & \text{(7b)} \end{cases}$$

with null boundary condition $\rho(\xi) = 0 \forall \xi \in \partial\Omega$, where $\gamma(\xi)$ is a continuous function over Ω. Hence, the transformation can be fully parameterized by $J_\phi(\xi)$ and $\gamma(\xi)$. We ensure the uniqueness of the solution using the Dirichlet boundary condition [22]. Note that a diffeomorphism corresponds to a positive transformation Jacobian determinant, which is enforced explicitly via the monitor function [23]. The above problem can be solved by a *step-then-correct* optimization strategy iteratively to find the point correspondence.

The endocardial and epicardial boundary points were tracked in all the frames of a cardiac cycle using the transformation function $\hat{\phi}$ and the manual segmentation on the first frame. The amount of deformation is smaller between neighboring frames than the deformations between frames that are further in temporal domain. Therefore, we compute the transformations between neighboring frames, for instance T_{k+1} and T_k, rather than computing the transformations between T_{k+1} and T_1. We use the above approach to improve the tracking accuracy and convergence time. However, this approach may lead to accumulation of tracking errors. In order to reduce the accumulation of tracking errors, two separate propagation algorithms along forward and backward directions performed throughout the cardiac cycles. Fig. 2 shows the steps of computing the contours for all the frames of a cardiac cycle. The displacement vectors correspond to the point-to-point correspondence are calculated using interpolation.

$$DM_{k,1} = (1 - w)DM^f_{k,1} + wDM^b_{k,1} \tag{8}$$

where $DM^f_{k,1}$ and $DM^b_{k,1}$ correspond to the displacement vectors between the k^{th} and first frames in the forward and backward directions, and $w = (k - 1)/(K - 1)$.

A. PARALLEL IMPLEMENTATION

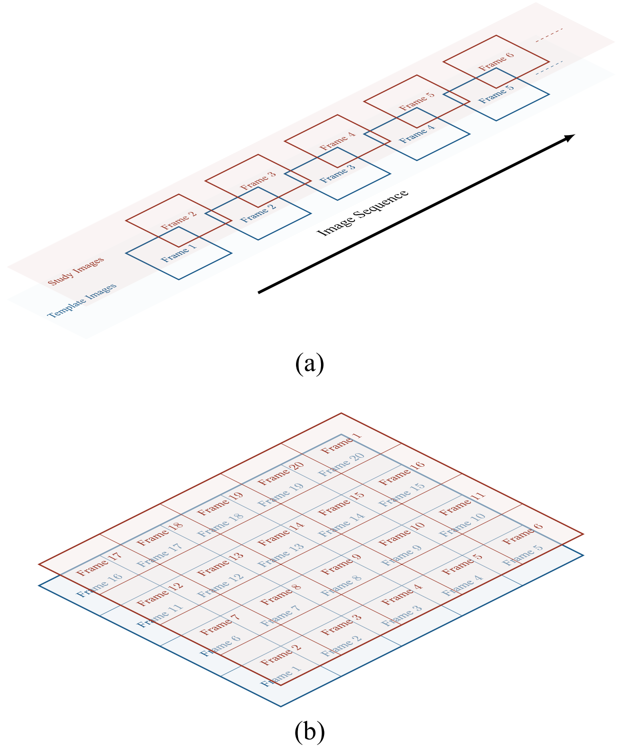
We use two approaches in parallelizing the moving mesh computation: 1) intra-image parallelism; and 2) inter-image parallelism.

1) INTRA-IMAGE PARALLELISM

The intra-image parallelism is obtained by parallelizing the computational components of the moving mesh algorithm. GPU hardware allows for parallel execution of large number of threads. The threads are grouped into blocks and blocks are grouped in a grid. The resources available in each thread is limited and shared between threads of each block. Optimum block per grid size was computed using the size of the deformation grid and the selected threads per block. Parallel executions were implemented for the computation of the transformation for each pixel in the moving mesh, the cost function using the squared L_2-norm and the optimizer. To minimize the overhead associated with the data transfer between GPU and CPU memory locations, the image information was transfered to the GPU memory and all the subsequent computations associated with nonrigid registration were executed in the GPU. Only the final results correspond to the mapping between the images were transfered back to the CPU memory. The forward and backward propagation of the moving mesh algorithm were also computed in parallel as they do not rely on each other.

2) INTER-IMAGE PARALLELISM

We used inter-image parallelism to further parallelize the moving mesh correspondences between a sequence of images. The traditional way of computing image registration is to register two images at a time as depicted in Fig. 3(a). This process requires running the image registration algorithm $K-1$ times to compute the point correspondences between a sequence of K images. The process can be parallelized by using image concatenation to create two separate images with an offset of one image as depicted in Fig. 3(b). Then, the moving mesh correspondences can be computed between the concatenated images in a single step. Similar to the intra-image parallelism, the concatenated image information was transfered to the GPU memory and all the subsequent computations associated with nonrigid registration were executed in the GPU to minimize the overhead associated with the data transfer between GPU and CPU memory locations. The moving mesh correspondences between the individual image pairs in the sequence were then be obtained by splitting the matrix computed from the concatenated images. We set the initial boundary to be equal to zero and ignored the effects of boundary condition for the inter-image parallelized version of the GPU implementation.



(a)

(b)

FIGURE 3. Inter-image parallelism: (a) Traditional way of computing image registration using a sequential approach which requires $K-1$ steps to register a sequence of K images; (b) Using image concatenation to create two image tiles with an offset of one image which allows the computation of image registration in a single step.

B. IMPLEMENTATION

Three versions of the algorithm were implemented: 1) CPU; 2) GPU with only intra-image parallelization; and 3) GPU with inter and intra-image parallelization. The algorithms were implemented using the Python programming language. A computationally expensive part of the algorithm for the CPU version was implemented in C using Cython module to improve the computational performance significantly over the pure Python version of the algorithm. The GPU CUDA version was implemented using Numbapro (Continuum Analytics, Austin, TX) with 32-bit floating-point precision. We used cufft and cublas CUDA submodules in the GPU implementation. The single thread CPU version of the algorithm was computed on 2.6 GHz Intel Core i7 processor. The GPU versions of the algorithm were computed on a NVIDIA GeForce GTX 1060 graphics card which has 10 streaming multiprocessors each with 128 streaming processors (SPs), *i.e.,* a total of 1280 SPs. We set the threads per block to 32 for all GPU implementations.

C. MAGNETIC RESONANCE IMAGING DATA

The proposed approaches were evaluated over an MR image data set composed of 16 subjects provided by the Right Ventricular Segmentation Challenge, MICCAI 2012 [15]. The data sets were acquired on 1.5T MR scanners (Symphony Tim, Siemens Medical Systems, Erlangen, Germany) with steady-state free precession acquisition mode. The MRI sequence parameters are as follows: TR = 50 ms; TE = 1.7 ms; flip angle = 55; slice thickness = 7 mm; matrix size = 256 × 216; Field of view = 360–420 mm; 20 images per cardiac cycle. For the moving mesh, grid sampling was set equal to the pixel spacing of the MR images.

D. QUANTITATIVE EVALUATION METRICS

In addition to the assessment of the computational performance, we assessed the accuracy of the GPU and CPU versions of the algorithms in comparison to manual delineation of the right ventricle. We relied on the following two metrics for the quantitative analysis of the accuracy.

1) THE DICE METRIC (DM)

The DM is a well-known metric to measure the similarity between manual and automated delineations. The DM is given by

$$DM(V_a, V_m) = \frac{2V_{am}}{V_a + V_m}, \tag{9}$$

where V_a is the volume of the automatically segmented region, V_m is the volume of the manually segmented region, and V_{am} is the intersection between the manual and automated segmented volumes. The DM is bounded by [0, 1]. The higher the value of the DM, the better the similarity.

2) THE HAUSDORFF DISTANCE (HD)

The HD [24] is another well-known metric which measures the maximum deviation between the manual and automated contours. The HD between the automated contour C_a and manual contour C_m is given by

$$HD(C_a, C_m) = \max(\max_i(\min_j(d(p_a^i, p_m^j))), \max_j(\min_i(d(p_a^i, p_m^j)))) \tag{10}$$

where $\{p_a^i\}$ denotes the set of all the points in C_a, $\{p_m^j\}$ denotes the set of all the points in C_m, and $d(\cdot)$ denotes the Euclidean distance. We computed the HD in mm using the spatial resolution obtained from the pixel spacing information of the DICOM header.

III. RESULTS

A. COMPUTATIONAL PERFORMANCE

Fig. 4 shows the computational times for the serial image registration algorithm implemented for CPU and GPU. The computational time for the CPU implementation increases linearly with the number of pixels in the image. However, the computational time for the GPU implementation is approximately constant with the number of pixels. Fig. 5 shows the runtime for kernel and overhead executions of the GPU implementation of the sequential image registration approach for different ranges of number of pixel values. The runtime was measured using NVIDIA command line profiling tool, nvprof. Although the runtime for kernel execution time was increasing approximately linearly with the number of pixel values, it was only a small fraction of the total computational time. Majority of the runtime was spent on overhead such as memory transfer, memory allocation, kernel launch and other CUDA application programming interface (API) calls. Fig. 6 shows the runtime for kernel and overhead executions of the GPU implementation of the concatenated image registration approach. The number of API calls have been reduced significantly for GPU implementation of the concatenated image registration due to reduction in the number of pair of

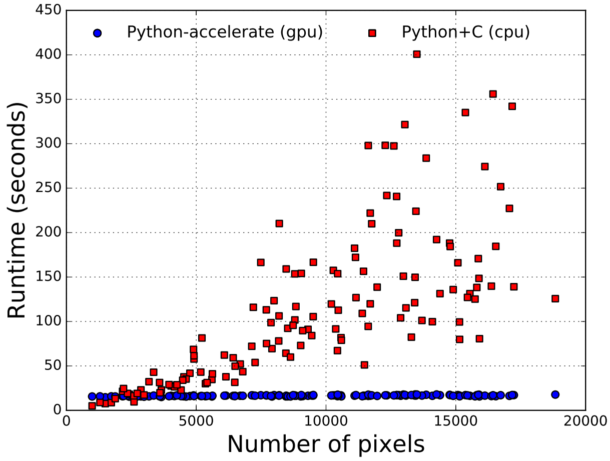


FIGURE 4. Computational times for the CPU and GPU implementations of the sequential image registration algorithm. The computational time for the CPU implementation approximately increases linearly with the number of pixels in the image, whereas the computational time is approximately constant for the GPU implementation.

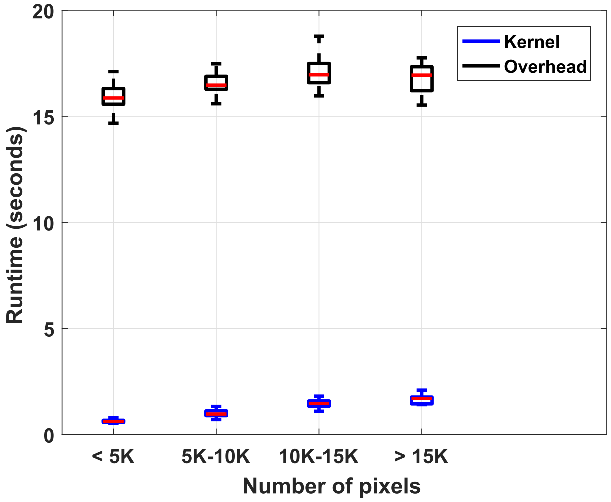


FIGURE 5. Runtime for kernel and overhead executions of the GPU implementation of the sequential image registration algorithm. The figure shows that the majority of the runtime was spent on overhead such as CUDA API calls than the execution of kernels.

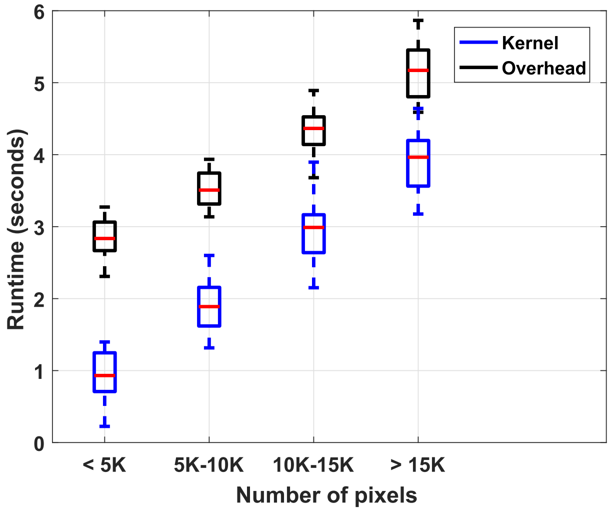


FIGURE 6. Runtime for kernel and overhead executions of the GPU implementation of the concatenated image registration approach. The number of CUDA API calls have been reduced significantly due to the reduction in number of pair of images used in the registration process which led to the reduction of the computational overhead.

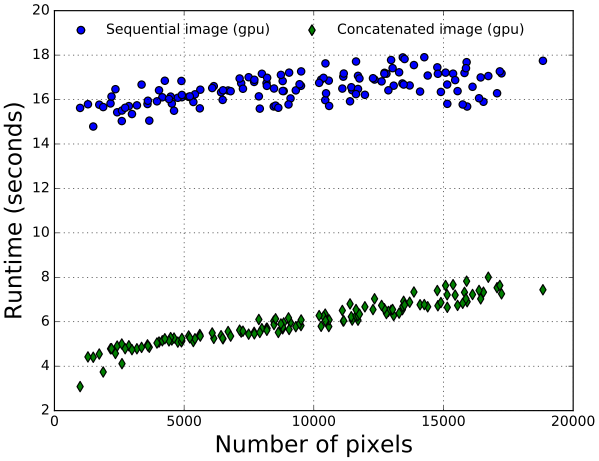


FIGURE 7. Computational times for parallel and serial image registration algorithms implemented using GPU computing. The proposed parallel image registration algorithm yielded a significant performance improvement.

images used in the registration process, *i.e.,* only one pair was used in concatenated image approach as opposed to 20 pair of images in sequential registration approach. Fig. 7 shows the computational times for the GPU implementation of serial and concatenated image registration approaches with the number of pixels.

Table 1 shows the computational time, peak GPU memory usage and GPU computational unit usage for the proposed parallel image registration algorithm, and sequential image registration algorithms implemented for the GPU and CPU. The GPU implementation of the concatenated image registration approach took an average computational time of 5.96 seconds, a three times speedup in comparison to the GPU implementation of sequential image registration. The GPU memory and computational unit usage were measured using GPU-Z tool. The GPU implementation of the concatenated image registration approach consumed a maximum GPU memory of 261 MB, around 140 MB higher than the maximum GPU memory consumption by the sequential image registration implementation. The GPU implementation of the concatenated image registration approach utilized the computational unit at an average rate of 38.7% which is more than 4.5 times the utilization by the sequential registration approach.

TABLE 1. The computational time, peak GPU memory usage and GPU computational unit usage for the proposed approach and sequential image registration algorithms. The proposed parallel registration approach yielded a significant performance improvement over the sequential approach.

Implementation	Moving mesh computation (seconds)	Peak GPU memory usage (MB)	GPU load (%)
GPU: Concatenated	5.96 ± 0.91	261	38.7 ± 13.3
GPU: Sequential	16.50 ± 0.65	112	8.5 ± 2.3
CPU (Python + C)	113.13 ± 84.59	N/A	N/A

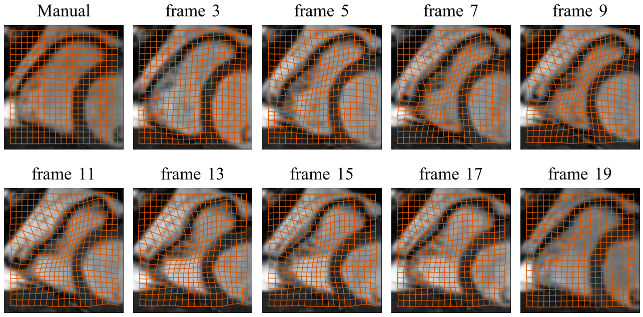


FIGURE 8. Representative examples showing RV deformations computed via grid generation using concatenated image registration approach.

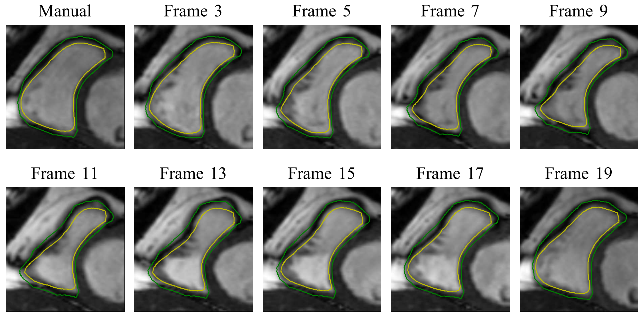


FIGURE 9. Representative examples of segmented endocardial (green) and epicardial (yellow) borders of the RV over a complete cardiac cycle using concatenated image registration approach.

B. EVALUATIONS OF SEGMENTATION ACCURACY

In Fig. 8, we give representative examples showing RV deformations computed using the proposed parallel method. We could see that grid folding is not present in any of the frames, although the amount of deformation is significant near the end-systolic phase. In Fig. 9, we give representative examples of segmented endocardial and epicardial borders of the RV over a complete cardiac cycle. These examples show that the proposed method accurately tracked both endocardial and epicardial border over the entire cardiac cycle.

1) QUANTITATIVE EVALUATION

The CPU, GPU sequential image and GPU concatenated image based approaches have been quantitatively evaluated over 16 subjects from the training set of the MICCAI 2012 RV segmentation challenge dataset. The manual segmentation for the end-systolic (ES) and end-diastolic (ED) phases of the heart were provided by the challenge organizers. Table 2 reports the DM and HD values for the training set as ES. Automated contours from all three algorithms for the endocardial and epicardial segmentations yielded average DM values of 0.82 and 0.86 in comparison to manual contours, respectively.

TABLE 2. Quantitative comparisons of the automatic contours by the CPU, GPU sequential image and GPU concatenated image implementations of the algorithm at end-systole. The higher the Dice Metric (DM) or the lower the Hausdorff Distance (HD), the better the performance.

	Endocardium		Epicardium	
Method	DM	HD	DM	HD
GPU: Concatenated image	0.82 (0.15)	7.19 (4.05)	0.86 (0.10)	7.61 (3.83)
GPU: Sequential image	0.82 (0.15)	7.00 (3.99)	0.86 (0.10)	7.69 (3.71)
CPU	0.82 (0.15)	7.05 (3.80)	0.86 (0.09)	7.87 (3.57)

IV. DISCUSSION

Although the proposed registration will be applied as a post-processing approach, it is important to obtain the results within a limited time frame for clinical applications due to the large number of images associated with the MR scans. In this study, we have demonstrated that the GPU computing can be utilized for improving the computational performance of an image registration algorithms without compromising the accuracy. We have also shown that the proposed algorithm can be used for tracking the endocardial and epicardial borders of the right ventricle over a sequence of MR images, given an initial manual segmentation on the first frame. The proposed approach does not require a time-consuming manually-built training set. Further, the method does not assume any shape constraints in the delineation process of the right ventricle.

A number of studies have been proposed in the literature to utilize non-rigid registration approaches to segment organs from a sequence of medical images, given manual segmentation on the first frame. For instance, Odille *et al.* [25] proposed a non-rigid registration approach to segment aorta from real-time velocity mapping MR images. Also, several studies have been proposed in the literature to improve the performance of the image registration algorithms using graphics hardware [26]–[31]. However, most of the existing methods are application-specific, and they do not take full advantage of the features particular to the cardiac deformation. For instance, the nonrigid registration approach proposed in this study prevents mesh folding, *i.e.,* grid lines of the same family will not cross each other, an essential attribute in tracking the cardiac deformation. In addition, the proposed registration approach allows for setting the minimum and maximum allowable deformation to imitate the deformation of cardiac tissue to improve the tracking accuracy.

The results for the CPU version of the proposed nonrigid registration algorithms for Test1 and Test2 datasets of the Right Ventricular Segmentation Challenge, MICCAI 2012 [15] is presented in [12]. The CPU version of the algorithm yielded a Dice score of 0.83 and 0.85 for endocardium for Test1 and Test2 datasets, respectively. The corresponding Dice scores for epicardium were 0.87 and 0.88. These results show that the proposed nonrigid registration based method performed significantly better than the other methods in [16] and [17] for both endocardial and epicardial segmentations. The GPU versions of the algorithm are expected to produce similar results since they generate nearly identical moving mesh correspondences. The GPU implementation of the concatenated image registration approach took around 6 seconds to segment the entire set of images in a sequence whereas most existing methods segments only the end-diastole and end-systole frames. The proposed nonrigid registration based algorithm does not utilize shape or distance priors in the segmentation process as in [18] and [19]. Also, the proposed approach does not require a training set as in [32] and [33]. Lack of reliance on shape priors and training sets led to a more robust algorithm, and the proposed nonrigid registration approach has been shown to be effective in segmenting the RV with considerable shape difference such as hypoplastic left heart syndrome hearts [12]. Another advantage of using registration based method for tracking the boundary is that it provides the ability to assess the regional function of the ventricle by retaining the point correspondence over the image sequence [13].

Most of the previous GPU computing based registration approaches mainly focus on parallelizing one or more individual components of the registration process. Kubias *et al.* proposed utilization of GPU computing for parallelizing the computation of similarity measures [26]. Ruiz *et al.* [27] used GPU implementation for the most computationally demanding part of their algorithm, the calculation of cross-correlations between images. Modat *et al.* [28] proposed a GPU implementation to compute free form deformation in parallel. Huang *et al.* [29] utilized GPU for computing the transformations required by the registration approach and relied on CPU for histogram and similarity measure computations. A few notable exceptions where the entire registration calculations were performed on the GPU include [30], [31]. In addition to performing the entire registration using GPU computing, one of the major contributions of our approach is the algorithmic changes in the registration process by concatenating the images to further accelerate the computational performance.

Parallelization of components of the registration algorithm using GPU programming yielded a computational performance improvement of 6.8× in comparison to the CPU implementation. The image concatenation approach produced a significant computational performance improvement and offered a speed up of 19× over the CPU implementation. All three versions of the algorithm yielded the same level accuracy in terms of Dice metric in comparison to expert manual contours. The peak GPU memory consumption by the implementation of the concatenated image registration approach is 261 MB, a value much smaller than the amount of memory available on most modern GPUs.

Although the algorithms presented in this study computes the automatic contours for all the images in an MRI sequence, the evaluations were performed only on ES due to the unavailability of the manual contours on all images of the cardiac cycle. The ability to compute contours over the whole cardiac cycle is advantageous in producing an in-depth set of clinical parameters. For instance, we require contours over the entire cardiac cycle to compute a filling rate curve which can be used for finding early and late ventricular filling parameters.

V. CONCLUSIONS

In this study, we proposed a parallel implementation of a moving mesh approach for performing deformable image registration using GPU computing. The proposed algorithm consist of two-fold parallelization: 1) intra-image parallelism which consists of computing the moving mesh correspondences using CUDA platform for GPU computing; and 2) inter-image parallelism which consists of concatenating a sequence of images and then compute the moving mesh correspondences as if it is one single image. The proposed approach yielded a significant performance improvement and computed the forward and backward moving mesh correspondences with an average of 5.96 ± 0.91 seconds for a sequence of 20 images.

CONFLICT OF INTEREST

MN and KP have submitted a patent application based on the proposed RV segmentation approach. Authors declare that they have no other competing interests.

ETHICAL APPROVAL

The study relied on a dataset provided by the MICCAI 2012 Right Ventricle Segmentation Challenge organizers. Institutional review board approval and patient written consent were obtained by the organizers [16].

REFERENCES

[1] A. Goshtasby, *Image registration: Principles, Tools and Methods*. London, U.K.: Springer, 2012.

[2] T. Boehler, D. van Straaten, S. Wirtz, and H.-O. Peitgen, "A robust and extendible framework for medical image registration focused on rapid clinical application deployment," *Comput. Biol. Med.*, vol. 41, no. 6, pp. 340–349, Jun. 2011.

[3] R. J. Lapeer, S. K. Shah, and R. S. Rowland, "An optimised radial basis function algorithm for fast non-rigid registration of medical images," *Comput. Biol. Med.*, vol. 40, no. 1, pp. 1–7, Jan. 2010.

[4] O. Dandekar and R. Shekhar, "FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions," *IEEE Trans. Biomed. Circuits Syst.*, vol. 1, no. 2, pp. 116–127, Jun. 2007.

[5] A. Eklund, P. Dufort, D. Forsberg, and S. M. LaConte, "Medical image processing on the GPU—Past, present and future," *Med. Image Anal.*, vol. 17, no. 8, pp. 1073–1094, Dec. 2013.

[6] K. Sidiropoulos *et al.*, "Real time decision support system for diagnosis of rare cancers, trained in parallel, on a graphics processing unit," *Comput. Biol. Med.*, vol. 42, no. 4, pp. 376–386, Apr. 2012.

[7] S. Szénási, "Segmentation of colon tissue sample images using multiple graphics accelerators," *Comput. Biol. Med.*, vol. 51, pp. 93–103, Aug. 2014.

[8] T.-H. Lee, J. Lee, H. Lee, H. Kye, Y. G. Shin, and S. H. Kim, "Fast perspective volume ray casting method using GPU-based acceleration techniques for translucency rendering in 3D endoluminal CT colonography," *Comput. Biol. Med.*, vol. 39, no. 8, pp. 657–666, Aug. 2009.

[9] K. Kwon, E.-S. Lee, and B.-S. Shin, "GPU-accelerated 3D mipmap for real-time visualization of ultrasound volume data," *Comput. Biol. Med.*, vol. 43, no. 10, pp. 1382–1389, Oct. 2013.

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

[11] O. Fluck, C. Vetter, W. Wein, A. Kamen, B. Preim, and R. Westermann, "A survey of medical image registration on graphics hardware," *Comput. Methods Programs Biomed.*, vol. 104, no. 3, pp. e45–e57, Dec. 2011.

[12] K. Punithakumar, M. Noga, I. Ben Ayed, and P. Boulanger, "Right ventricular segmentation in cardiac MRI with moving mesh correspondences," *Comput. Med. Imag. Graph.*, vol. 43, pp. 15–25, Jul. 2015.

[13] K. Punithakumar *et al.*, "Regional heart motion abnormality detection: An information theoretic approach," *Med. Image Anal.*, vol. 17, no. 3, pp. 311–324, Apr. 2013.

[14] F. Haddad, S. A. Hunt, D. N. Rosenthal, and D. J. Murphy, "Right ventricular function in cardiovascular disease, part I: Anatomy, physiology, aging, and functional assessment of the right ventricle," *Circulation*, vol. 117, no. 11, pp. 1436–1448, 2008.

[15] J. Caudron, J. Fares, V. Lefebvre, P.-H. Vivier, C. Petitjean, and J.-N. Dacher, "Cardiac MRI assessment of right ventricular function in acquired heart disease: Factors of variability," *Acad. Radiol.*, vol. 19, no. 8, pp. 991–1002, Aug. 2012.

[16] C. Petitjean *et al.*, "Right ventricle segmentation from cardiac MRI: A collation study," *Med. Image Anal.*, vol. 19, no. 1, pp. 187–202, Jan. 2015.

[17] J. Ringenberg, M. Deo, V. Devabhaktuni, O. Berenfeld, P. Boyers, and J. Gold, "Fast, accurate, and fully automatic segmentation of the right ventricle in short-axis cardiac MRI," *Comput. Med. Imag. Graph.*, vol. 38, no. 3, pp. 190–201, Apr. 2014.

[18] A. Atehortúa, M. A. Zuluaga, J. D. García, and E. Romero, "Automatic segmentation of right ventricle in cardiac cine MR images using a saliency analysis," *Med. Phys.*, vol. 43, no. 12, pp. 6270–6281, Dec. 2016.

[19] Y. Liu *et al.*, "Distance regularized two level sets for segmentation of left and right ventricles from cine-MRI," *Magn. Reson. Imag.*, vol. 34, no. 5, pp. 699–706, Jun. 2016.

[20] K. Punithakumar, M. Noga, and P. Boulanger, "A GPU accelerated moving mesh correspondence algorithm with applications to RV segmentation," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2015, pp. 4206–4209.

[21] H.-M. Chen, A. Goela, G. J. Garvin, and S. Li, "A parameterization of deformation fields for diffeomorphic image registration and its application to myocardial delineation," in *Medical Image Computing and Computer-Assisted Intervention, Part I*, vol. 6361, T. Jiang *et al.*, Eds. Berlin, Germany: Springer, 2010, pp. 340–348.

[22] X. Zhou, "On uniqueness theorem of a vector function," *Prog. Electromagn. Res.*, vol. 65, pp. 93–102, 2006.

[23] J. Liu, "New development of the deformation method," Ph.D. dissertation, Dept. Math., Univ. Texas Arlington, Arlington, TX, USA, 2006.

[24] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[25] F. Odille, J. A. Steeden, V. Muthurangu, and D. Atkinson, "Automatic segmentation propagation of the aorta in real-time phase contrast MRI using nonrigid registration," *J. Magn. Reson. Imag.*, vol. 33, no. 1, pp. 232–238, Jan. 2011.

[26] A. Kubias, F. Deinzer, T. Feldmann, D. Paulus, B. Schreiber, and T. Brunner, "2D/3D image registration on the GPU," *Pattern Recognit. Image Anal.*, vol. 18, no. 3, pp. 381–389, Sep. 2008.

[27] A. Ruiz, M. Ujaldon, L. Cooper, and K. Huang, "Non-rigid registration for large sets of microscopic images on graphics processors," *J. Signal Process. Syst.*, vol. 55, no. 1, pp. 229–250, Apr. 2009.

[28] M. Modat *et al.*, "Fast free-form deformation using graphics processing units," *Comput. Methods Programs Biomed.*, vol. 98, no. 3, pp. 278–284, Jun. 2010.

[29] T.-Y. Huang, Y.-W. Tang, and S.-Y. Ju, "Accelerating image registration of MRI by GPU-based parallel computation," *Magn. Reson. Imag.*, vol. 29, no. 5, pp. 712–716, Jun. 2011.

[30] H. Mousazadeh, B. Marami, S. Sirouspour, and A. Patriciu, "GPU implementation of a deformable 3D image registration algorithm," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2011, pp. 4897–4900.

[31] D. Ruijters, B. M. ter Haar Romeny, and P. Suetens, "GPU-accelerated elastic 3D image registration for intra-surgical applications," *Comput. Methods Programs Biomed.*, vol. 103, no. 2, pp. 104–112, Aug. 2011.

[32] C. M. S. Nambakhsh, T. M. Peters, A. Islam, and I. Ben Ayed, "Right ventricle segmentation with probability product kernel constraints," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 8149, K. Mori *et al.*, Eds. Berlin, Germany: Springer, 2013, pp. 509–517.

[33] P. V. Tran. (2016). "A fully convolutional neural network for cardiac segmentation in short-axis MRI." [Online]. Available: https://arxiv.org/abs/1604.00494

KUMARADEVAN PUNITHAKUMAR (S'06–M'08–SM'17) received the B.Sc.Eng. degree (Hons.) in electronic and telecommunication engineering from the University of Moratuwa, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from McMaster University.

From 2001 to 2002, he was an Instructor with the Department of Electronic and Telecommunication Engineering, University of Moratuwa. From 2002 to 2007, he was a Teaching Assistant and a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, McMaster University, in 2008. From 2008 to 2012, he was an Imaging Research Scientist with GE Healthcare, Canada. He is currently an Assistant Professor with the Department of Radiology and Diagnostic Imaging, University of Alberta, and the Operational and Computational Director with the Servier Virtual Cardiac Center, Mazankowski Alberta Heart Institute.

Dr. Punithakumar was a recipient of the Industrial Research and Development Fellowship by the National Sciences and Engineering Research Council of Canada in 2008, and the GE Innovation Award in 2009. His interest include medical image analysis and visualization, information fusion, object tracking, and nonlinear filtering.

PIERRE BOULANGER has a double appointment as a Professor with the Department of Computing Science and the Department of Radiology and Diagnostic Imaging, University of Alberta. He cumulates more than 32 years of experience in 3-D computer vision, medical imaging, and the applications of virtual reality systems to medicine. He is currently the Director of the Advanced Man Machine Interface Laboratory and also the Scientific Director of the Servier Virtual Cardiac Centre. In 2013, he received the CISCO Chair in healthcare solutions, a ten years investment by CISCO systems in the development of new IT technologies for healthcare in Canada. His work has contributed to gain him an international recognition in this field, publishing over 320 scientific papers and collaborating with more than ten universities, research labs, and industrial companies across the world. His main research topics are on the development of new techniques for telemedicine, patient specific modeling using sensor fusion, and the application of virtual/augmented reality technologies to medical training, simulation, and collaborative diagnostics.

MICHELLE NOGA is a Radiologist trained at University of Alberta and British Columbia's Children's and Women's hospital. She is currently an Associate Professor with the Department of Radiology and Diagnostic Imaging, and a Radiologist with Medical Imaging Consultants, University of Alberta. She has authored over 80 publications and conference proceedings and holds three patents. He established the first Cardiac MRI Program in Alberta, and has successfully established the first centre for virtual 3-D display of cardiac imaging in a clinical setting, a product of a $1,000,000 grant from Servier Canada. Her areas of interest include pediatric cardiac MRI and CT, post-processing of cross-sectional cardiac imaging, 3-D visualization, rapid prototyping, pediatric airway, and finite-element analysis.