J Real-Time Image Proc (2019) 16:1891–1908
DOI 10.1007/s11554-017-0734-z

SURVEY PAPER

Techniques of medical image processing and analysis accelerated by high-performance computing: a systematic literature review

Carlos A. S. J. Gulo[1,2] · Antonio C. Sementille[3] · João Manuel R. S. Tavares[4]

Received: 18 January 2017 / Accepted: 2 November 2017 / Published online: 16 November 2017

Abstract Techniques of medical image processing and analysis play a crucial role in many clinical scenarios, including in diagnosis and treatment planning. However, immense quantities of data and high complexity of the algorithms often used are computationally demanding. As a result, there now exists a wide range of techniques of medical image processing and analysis that require the application of high-performance computing solutions in order to reduce the required runtime. The main purpose of this review is to provide a comprehensive reference source of techniques of medical image processing and analysis that have been accelerated by high-performance computing solutions. With this in mind, the articles available in the Scopus and Web of Science electronic repositories were searched. Subsequently, the most relevant articles found were individually analyzed in order to identify: (a) the metrics used to evaluate computing performance, (b) the high-performance computing solution used, (c) the parallel design adopted, and (d) the task of medical image processing and analysis involved. Hence, the techniques of medical image processing and analysis found were identified, reviewed, and discussed, particularly in terms of computational performance. Consequently, the techniques reviewed herein present the progress made so far in reducing the computational runtime involved, and the difficulties and challenges that remain to be overcome.

Keywords Medical imaging · Image segmentation · Image registration · Image reconstruction

✉ João Manuel R. S. Tavares
tavares@fe.up.pt

Carlos A. S. J. Gulo
sander@unemat.br

Antonio C. Sementille
sementille@fc.unesp.br

[1] CNPq National Scientific and Technological Development Council, Research Group PIXEL - UNEMAT, Alto Araguaia-MT, Brazil

[2] Programa Doutoral em Engenharia Informática, Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

[3] Departamento de Ciências da Computação, Faculdade de Ciências, Universidade Estadual Paulista-UNESP, Bauru-SP, Brazil

[4] Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

1 Introduction

Throughout the history of computer systems, the evolution of processors and increases in computing speed have been closely related. Traditionally, the integrated circuit industry has fitted ever more transistors into a single chip thereby achieving high performance [45]. However, this approach is limited by physical restrictions of silicon, mainly excessive energy consumption and overheating of processors [90].

In recent years, advances in this area have taken a different direction, leading to modern processor architecture used for (a) multi-core CPUs (which contain two or more processing cores) and (b) the general purpose computing on graphics processing units (GPGPU), which is defined in this review as "many-core architecture". Both many- and multi-core architectures exploit parallelism features that offer performance gains and faster computing [90].

The demand for high-performance computing has generally been addressed with costly computational systems. However, in view of the popularity of graphics processing units (GPUs) and the adoption of parallel programming methods, a number of research areas can advance significantly without the need for major investment in computational systems. Examples of these areas include: scientific simulation [19], life sciences [91], statistical modeling [91], emerging data-intensive applications [91], electronic design automation [91], ray tracing and rendering [96], computer vision [23], signal processing [23, 91], and medical image processing and analysis [15, 29, 88].

The area of medical image processing and analysis has contributed to significant medical advances [7, 23, 50, 81, 83, 88, 101] by integrating systems and techniques that support more efficient clinical diagnosis. These systems and techniques are based on images acquired by different imaging modalities such as, endoscopy [52], X-ray [88], microscopy [47, 68], computed tomography (CT) [26, 57], optical coherence tomography (OCT) [67], magnetic resonance (MR) [2, 15], functional magnetic resonance (fMR) [3, 97], magnetic resonance elastography (MRE) [20], positron emission tomography (PET) [17, 42, 43], single photon emission computed tomography (SPECT) [28], and 3D ultrasound computer tomography (USCT) [7].

Medical imaging assists physicians in extracting information for the purposes of diagnosing diseases, surgical intervention, treatment and follow-up of diseases, as well as in designing better rehabilitation plans [29, 37, 95, 97]. Such extraction of relevant clinical information is a complex task requiring advanced computational systems able to process and obtain image-based features accurately and consistently within the shortest possible runtime. As a result, a new research area has emerged that combines computational techniques used for medical image processing and analysis [23, 81, 88] and high-performance computing solutions [7, 50, 83, 101]. These two components can be briefly described as follows:

- Medical image processing and analysis—Typically, the researchers of this area attempt to find solutions that start by improving the quality of the input images, and then apply operations on the enhanced images in order to identify and extract meaningful clinical information [23, 81, 88]. In this context, the term "medical image processing and analysis" is used throughout the present review.
- High-performance computing—The main goal of this area is to optimize computational methods to achieve greater robustness, effectiveness, efficiency, and faster execution. To accomplish these objectives, parallel computing techniques are usually exploited to use the maximum available performance in the computational architecture adopted [7, 50, 83, 101].

The number of researchers combining techniques of medical image processing and analysis and of high-performance computing has increased considerably in recent years; consequently, this article aims to present an updated systematic literature review of this area. The scientific articles selected for this review provide valuable information for researchers in the two fields identified; specifically, the articles address methods, techniques, imaging modalities, metrics of computational performance, and the most frequently used computing architectures. The contributions made by each selected article are therefore set out and the remaining research gaps are identified; this will be of significant value to those who intend to develop, evaluate, and compare algorithms used in medical image processing and analysis accelerated by high-performance computing architectures.

The term "performance" is sometimes ambiguous; hence, in this article, "performance" refers to the efficiency of computing systems when executing algorithms, including the factors of throughput, latency, and availability. The methodology employed to select, identify, and validate the articles considered is presented in Sect. 2; the main findings extracted from the articles analyzed are summarized in Sect. 2.1; the contributions found in the selected articles and the gaps identified are presented and discussed in Sect. 3; finally, concluding remarks are presented in Sect. 4.

2 Systematic literature review

This section describes the protocol used to locate, gather, and appraise the state of the art under study. The first issue that was examined was the range of high-performance computing platforms and methods that have been used to speed up techniques of medical image processing and analysis. In addition, the following complementary questions were considered:

1. Which imaging modality was involved?
2. Which task of medical image processing and analysis was addressed?
3. Which human organ or tissue was analyzed?
4. What computational architecture was adopted and/or developed?
5. Which high-performance computing technique was adopted and/or developed?
6. Is the approach adopted and/or developed able to achieve real time?

The criteria defined for the selection of articles were as follows:

1. Domain
   (a) Medical image processing and analysis; and
   (b) High-performance computing.
2. Methods
   (a) Techniques of medical image processing and analysis accelerated by high-performance computing solutions.
3. Measures
   (a) Techniques of medical image processing and analysis; and
   (b) Performance in runtime.

After defining the selection criteria, the next step involved defining the exclusion criteria, which were as follows:

1. Duplicated references; for example, the same article retrieved from the different electronic repositories searched;
2. Less than four pages;
3. No description available on the technique of medical image processing and analysis;
4. No information available on the metric used to assess computing performance;
5. None of the research questions under consideration (numbered 1–5) are addressed.

Before initiating the article-gathering process, the language of the articles, the research domains, and the electronic repositories to be considered were defined. We decided to only review articles written in English, the dominant language used in the scientific domains of computer science and engineering. The repositories selected for searching were: *Scopus*[1] and *Web of Science*.[2]

The systematic review was carried out from March 2016 to August 2016 and updated in March 2017. Table 1 presents the search terms used when querying each repository and the total number of articles retrieved.

The search of the Web of Science repository was defined in order to locate the articles related to each of the following queries: (a) "*medical image*" OR "*medical imaging*", (b) "*high performance computing*" OR "*parallel computing*" OR "*parallel programming*" OR "*real-time processing*". These queries were combined using the AND logical operator in order to mimic the equivalent searches in the other repository. "*image processing*" was not used in the search because it could generalize the results too much; instead, the purpose of using "*medical image*" and "*medical imaging*" was to gather all scientific articles related to techniques of medical image processing and analysis.

After removing the 467 duplicate references, each of the remaining 2, 112 articles were then filtered according to the selection criteria, as shown in Table 2. The selection criteria were applied systematically to the title, keywords, and abstract of the articles in the electronic repositories searched, and this resulted in 594 articles. The content of each abstract was initially analyzed with the aim of identifying evidence of the use of high-performance computing architectures in order to support the acceleration of techniques of medical image processing and analysis.

Additionally, each article was classified according to three priority levels:

- *Prio-1*: Articles that are very relevant and suitable for the review such that there was evidence of the (previously defined) article-extraction criteria in the title, abstract, and even keyword fields ;
- *Prio-2*: Articles that are less important but still suitable;
- *Prio-3*: Articles that may be relevant to other related research, but are not main sources of knowledge for this review.

The classification priorities of the articles selected from each repository are indicated in Table 3. The values shown in this table indicate the suitability of each repository relative to each classification priority previously enumerated.

2.1 Review of selected articles

In the evaluation stage, the sections of each article presenting the applicable methodology, results, and conclusions were analyzed, in order to identify important information that answers the research questions (1–5) defined in Sect. 2.

In this review, a total of 594 articles were initially selected; however, 507 articles were then removed in accordance with the exclusion criteria, and the 87 remaining articles were analyzed in depth. The exclusion criteria were defined in such a way as to answer the aforementioned, main research questions. Hence, it was critical to identify in each article: the metric(s) used to evaluate computational performance; the high-performance computing architecture and parallel design involved; and the object(s), i.e., tissue(s) or organ(s), addressed by the technique(s) of medical image processing and analysis. Therefore, during the in-depth analysis of each article, critical information was collected to answer each specific research question.

Table 4 presents in descending chronological order the most relevant information extracted from the 87 articles

[1] http://www.scopus.com—Science Direct.

[2] http://apps.webofknowledge.com—Web of Science Core Collection.

Table 1 Total number of articles retrieved from each electronic repository

Repositories	Queries performed	No. of articles
Scopus	TITLE-ABS-KEY (("medical image" OR "medical imaging") AND ("high performance computing" OR "parallel programming" OR "parallel computing" OR "real-time processing")) AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))	421
Web of science	Filtering using the same queries searched above	2158
Total		2579

Table 2 Total articles retrieved, duplicated, and remaining after applying each criteria

Repositories	Retrieved	Duplicated	Selection criteria	Exclusion criteria
Scopus	421	17	288	32
Web of science	2158	450	306	55
Total	2579	467	594	87

Table 3 Relevance of each repository used to retrieve articles related to techniques of medical image processing and analysis combined with high-performance computing solutions

Repository	Prio-1 (%)	Prio-2 (%)	Prio-3 (%)
Scopus	71.64	17.41	10.95
Web of science	17.82	5.63	76.55

analyzed, including the description of the main high-performance computing methods applied to the acceleration of the techniques of medical image processing and analysis. The speedup column presents the computational performance results achieved by the authors in respect of the methods studied. Here, speedup is defined as the ratio of the execution time of serial and parallel implementations when both are applied on the same dataset and running on the same computer.

One conclusion drawn from the articles found is that, in recent years, and especially in the last decade, there has been considerable research into the use of techniques of image processing and analysis accelerated by high-performance computing solutions.

The first step in medical imaging consists of acquiring the data using a suitable imaging device and then reconstructing the related images. After that, a number of techniques of image processing and analysis can be applied, such as image reconstruction, image filtering, image segmentation, and image registration.

2.1.1 Image reconstruction

Image reconstruction is the process used to generate 2D/3D images of an object from the data, i.e., signals, acquired by an imaging device. In the data acquisition stage, the imaging device is responsible for converting the anatomical/physiological information into digital signals. However, digital signals are easily corrupted by noise introduced by the electronic/mechanical components of the imaging device [87]. Dominant physical effects such as resolution, attenuation, and scatter, are spatially variant, and in the cases of attenuation and scatter, may also differ according to the type of object, i.e., tissues, under study. In addition, a number of noise source displacements occur when acquiring MRE images. Lengthy extended movements produce common ambiguity errors, which, for example, result in weak estimates in regions with low signal noise rate. Susceptible effects generate inconsistencies during the estimation stage and result in erroneous estimate displacements. In general, all the image reconstruction approaches demand high computational costs and require large memory capacity, for example, in MRI, SPECT, and CT cases, where large datasets are used to reconstruct complex 3D images.

The article of Miller and Butler [57] considers the implementation of the maximum a posteriori (MAP) and maximum likelihood (ML) methods in a system that creates a complete 3D reconstruction from CT images and is accelerated by massively parallel processors. The iterative expectation-maximization (EM) algorithm, which is applied in order to generate ML and MAP estimates for SPECT image acquisitions, is considered highly complex in terms of computation [57]. Their parallel system was implemented on a massively parallel computer (DECmpp-SX 128 $\times$ 128 processor) and designed according to the single instruction, multiple data stream (SIMD) parallel programming model. Although the implementation did not indicate a linear scalability, the speedup achieved was 64$\times$, relative to an optimal programmed implementation to be executed in a reduced instruction set computing (RISC) architecture (64 $\times$ 64 processor). Formiconi et al. [28] also

Table 4 Summary of the studies found related to techniques of medical image processing and analysis supported by high-performance computing solutions

Research	Impact factor	Imaging modality(ies)	Image task(s)	Object(s) analyzed	Parallel architecture(s)	Parallel programming model	Speedup
Miller and Butler [57]	2.12	CT, SPECT	reconstruction	Brain	Massively parallel processor (MPP)	SIMD	64×
Kerr and Bartlett [44]	0.90	CT, SPECT	Reconstruction	Cardiac	MPP	SIMD	139×
Higgins and Swift [37]	0.30	CT	Reconstruction	Cardiac	MPP	SIMD	5×
Formiconi et al. [28]	0.45	CT, SPECT	Reconstruction	Brain	MPP	MIMD	135×
Christensen [12]	1.57	CT	Registration	Craniofacial	MPP and Cluster	SIMD and MIMD	20×
Daggett and Greenshields [15]	7.68	MRI	classification	Bladder and urethra	Cluster	SPMD	6×
Warfield et al. [95]	5.10	CT, MRI	Registration	Brain	Cluster	MIMD	15×
Saiviroonporn et al. [71]	2.10	CT, MRI	Segmentation	Bones, aorta, kidneys, skin, brain	MPP	SIMD	10×*
Yip et al. [98]	0.61	MRI	Reconstruction	Skull	Cluster	MIMD	500×
Rohlfing and Maurer [68]	17.28	MRI, microscopy	Registration	Brain and breast	MPP	MIMD	50×
Wachowiak and Peters [92, 93]	1.15, 3.72	MRI	Registration	Brain and heart	Cluster	MIMD	5×
Tirado-Ramos et al. [86]	1.84	MRI and CT	Reconstruction	Beast	Cluster	MIMD	3×
Doyley et al. [20]	1.84	MRE	Reconstruction	Beast	Cluster	MIMD	3×
Salomon et al. [72]	1.66	MRI	Registration	Brain	Cluster	MIMD	10×
Eidheim et al. [22]	0.91	Ultrasound	Segmentation	Liver	GPU	SIMT	34×*
Crane et al. [14]	0.90	MRI	Reconstruction	Brain	Cluster	MIMD	3×
Deng et al. [18]	2.90	CT	Reconstruction	Shepp-Logan phantom	Cluster	MIMD	32×
Dandekar and Shekhar [17]	4.6	CT, PET	Registration	Abdominal	FPGA	SIMD	30×
Yeh and Fu [97]	1.5	fMRI	Classification	Brain	Cluster	MIMD and SPMD	2×
Kalmoun et al. [41]	2.7	CT	Reconstruction	Heart	Cluster	MIMD	28×
Kumar et al. [47]	2.22	Microscopy	Reconstruction	Breast	Cluster	MIMD	2×
Samant et al. [73]	12.77	4DCT	Registration	Lung	GPU	SIMT	56×
Sehellmann et al. [77]	2.77	PET	Reconstruction	Lung	GPU	SIMT	7.5×
Melvin et al. [53]	0.66	CT	Reconstruction	Shepp-Logan phantom	Multi-core	SIMD	30×
Kegel et al. [42, 43]	3.62, 1.14	PET	Reconstruction	Rats	Multi-core	SPMD	3×
Rehman et al. [65]	5.62	MRI	Registration	Brain	GPU	SIMT	965×
Rohrer and Gong [69]	0.37	CT, MRI	Registration	Abdominal	CBEA	SIMD and MIMD	13×*
Zhuge et al. [100, 101]	2, 3.33	CT, MRI	Segmentation	Head	GPU	SIMT	18×*
Moyano-Avila et al. [58]	0	X-Ray	Reconstruction	Vessels	MPP	MIMD	15×
Chung et al. [13]	1.57	Microscopy	Reconstruction	Viruses	GPU	SIMT	16×
Shackleford et al. [79]	14	3D CT	Registration	Lung	GPU	SIMT	15×*

Table 4 continued

Research	Impact factor	Imaging modality(ies)	Image task(s)	Object(s) analyzed	Parallel architecture(s)	Parallel programming model	Speedup
Shams et al. [80, 81]	11	CT, MRI, PET	Registration	Brain	GPU	SIMT	50×*
Gabriel et al. [29]	2.57	FNAC	Segmentation	Thyroid	Cluster and Multi-core	MIMD and SIMD	11×
Lapeer et al. [48]	2.57	CT, MRI	Registration	Head	GPU	SIMT	10×
Zhu and Cochoff [99]	1.42	CT, PET	Registration	Lung	Multi-core	SPMD	2–10×
D'Amore et al. [16]	1	MRI	Segmentation	Skin	Multi-core	SIMD	6×
Meng et al. [54]	4.5	CT	Reconstruction	Lung	Cloud computing	MIMD	10×
Schmid et al. [76]	2.66	MRI	Segmentation	Bones	GPU	SIMT	70×
Schellmann et al. [75]	2.33	PET	Reconstruction	Mouse	GPU	SIMT	2×
Gao et al. [31]	1.16	MRI	Segmentation	Brain	GPU	SIMT	1440×*
Lee et al. [49]	7.8	MRI	Registration	Brain	GPU	SIMT	129×
Adeshina et al. [1]	1.66	MRA	Reconstruction	Brain	GPU	SIMT	3×
Murphy et al. [59]	22.4	MRI	Reconstruction	Torso	GPU and multi-core	SIMT and SIMD	40×
Zinterhof [103]	0	CT	Classification	Kidney	GPU	SIMT	120×
Shi et al. [83]	0.40	CT, MRI	Segmentation and reconstruction	Head, breast, vessels	GPU and multi-core	SIMT and SIMD	40×*
Rodrigues and Bernardes [67]	2	OCT	Filtering	Retinal	GPU	SIMT	18×*
Domanski et al. [19]	1.25	CT	Reconstruction	Brain	GPU and multi-core	SIMT and SIMD	9×
Treibig et al. [88]	5.75	CT, X-ray	Reconstruction	Rabbit	Multi-core	SIMD	6×
Gallea et al. [30]	1.28	CT, MRI	Registration	Brain	GPU	SIMT	100×
Saran et al. [74]	1.33	MRI	Segmentation	Breast	GPU and multi-core	SIMT and SIMD	35×
El-Moursy et al. [25]	0.66	3D MRI	Segmentation	Brain	Cluster	MIMD	2.6×
Balla-Arabé and Gao [5]	1.33	MRI	Segmentation	Breast	GPU	SIMT	6×*
Eklund et al. [24]	9	fMRI	Registration, segmentation, filtering	Brain	GPU	SIMT	195–525×
Barros et al. [6]	0	CT	Segmentation	Brain	GPU	SIMT	36×
Alvarado et al. [4]	2.33	CT, PET, MRI	Segmentation	Brain	GPU and multi-core	SIMT and SIMD	8
Birk et al. [7]	7	USCT	Reconstruction	Breast	GPU and multi-core	MIMD	25×*
Blas et al. [9]	2	CT	Reconstruction	Rats	GPU and multi-core	SIMT and SIMD	2×
Mafi and Sirouspour [50]	3.33	MRI	Reconstruction	Stomach	GPU	SIMT	28×*
Meng [55]	1.33	CT	Registration	Thorax	GPU	SIMT	255×
Wei et al. [96]	0.33	MRI	Reconstruction	Eye optics	GPU	SIMT	100×
Fan and Xie [27]	0	CT	Reconstruction	Shepp-Logan phantom	GPU	SIMT	20×
Serrano et al. [78]	0.5	CT	Reconstruction	Human body	GPU and Cluster	SIMT and MIMD	22×
Gates et al. [32]	8	CT	Segmentation	Brain	GPU	SIMT	43.5×
Akgun et al. [3]	1.50	fMRI	Segmentation	Brain	GPU and multi-core	SIMT and SIMD	157×

Table 4 continued

Research	Impact factor	Imaging modality(ies)	Image task(s)	Object(s) analyzed	Parallel architecture(s)	Parallel programming model	Speedup
Tan et al. [85]	0	Microscopy	Reconstruction	Virus	FPGA, GPU and multi-core	SIMD, SIMT and MIMD	14×
Mahmoudi and Manneback [51]	1.50	X-ray and MRI	Segmentation	Vertebra	Multi-core and multi-GPU	SIMD and MIMD	98×*
Johnsen et al. [40]	7.50	MRI	Registration	Breast	GPU	SIMT	5×
Hamdaoui et al. [35]	0	MRI	Reconstruction	Brain	FPGA	SIMD	37×
Cai et al. [10]	2.50	MRI	Registration	Lung	GPU and multi-core	SIMT and SIMD	4×
Smistad et al. [84]	0	CT, 3D ultrasound	Filtering and segmentation	Bone structure, retina blood vessels	GPU and multi-core	SIMD and SIMT	20×
Gulo et al. [34]	3	Ultrasound	Filtering	Stomach	GPU	SIMT	10×
Nguyena et al. [61]	4.50	MRI	Filtering	Brain	GPU, Cluster, and multi-core	SIMT, MIMD and SIMD	510×
Koestler et al. [46]	3.50	X-ray	Reconstruction	Head	GPU	SIMT	1.6×
Hu et al. [38]	1	CT	Reconstruction	Thorax	GPU	SIMT	202×
Du et al. [21]	0	CT, MRI	Registration	Brain, lung	GPU	SIMT	17×
Ellingwood et al. [26]	1	CT	Registration	Lung	GPU	SIMT	112×
Heras et al. [36]	2	MRI, CT	Segmentation	Brain	GPU	SIMT	6×
Chen et al. [11]	7	Ultrasound	Reconstruction	Forearm	GPU	SIMT	60×
Aitali et al. [2]	2	MRI	Segmentation	Skin	GPU	SIMT	52×
Riegler et al. [66]	4	Endoscopy	Classification	Gastrointestinal	Multi-core and GPU	SIMD and SIMT	10×
Pang et al. [64]	7	Ultrasound	Segmentation	Breast	GPU	SIMT	16×
Wang et al. [70, 94]	4, 2	CT	Reconstruction	Lungs	GPU	SIMT	4×
Jaros et al. [39]	2	CT	Segmentation	Heart and liver	GPU	SIMT	44×

The Impact Factor column was calculated using the ratio of the number of Google citations of the article and the number of years since its publication

* The performance achieved real-time

presented a parallel implementation of the EM algorithm; however, their approach was combined with ML estimates and applied in order to reconstruct images from SPECT data. The authors designed their implementation on the basis of a multiple instruction, multiple data stream (MIMD) parallel programming model and used a World Wide Web (WWW) interface. A massively parallel computer, Cray T3D, was used to calculate their computational solution remotely.

Massively parallel computers were adopted by Kerr and Bartlett [44] as described in another article. The authors examined the simulation and rapid training of a very large artificial neural network that reconstructs and compresses SPECT images. In this study, when comparing the performances obtained by CPU- and Parallel-based implementations, a speedup of 139× was achieved. The authors designed the suggested algorithm on the basis of the SIMD model.

Another research study that developed a parallel computer architecture was presented in the Higgins and Swift [37]'s article. These authors defined a "meta-computer" as a combination of communication devices and a heterogeneous processing architecture. Their goal was to implement a new parallel architecture using the parallel computer MasPar in order to manage multiple workstation interactions and process 3D medical images as fast as possible. The parallel architecture used in the experiments included typical tasks of medical image processing and analysis: image preprocessing, morphological and topological image operations, image segmentation, image manipulation, image measurement and the input and output of images. The approach of the authors resulted in a performance 5× faster than the equivalent algorithm implemented using a sequential fashion programming model.

Doyley et al. [20] proposed in their article a parallel approach to obtain partial volume reconstructions from 3D high-resolution data. The authors combined the finite element method (FEM) and the Newton–Raphson iterative scheme in this approach, which was implemented using Message Passing Interface (MPI) and executed on a PC-cluster. In the experiments, the authors adopted an optimized sequential approach in contrast to a parallel-based one. The parallel version improved the in/out storage disk operations and achieved a linear speedup.

Kumar et al. [47] developed a middleware system based on a PC-cluster architecture, the purpose of which was to support the execution of a set of techniques of image processing and analysis. These techniques were divided into two main stages: preprocessing and analysis. These tasks resulted in preprocessed data that could be queried and analyzed using the techniques of image analysis. The authors combined data and task parallelism models in order to achieve better scalability; moreover, they implemented the tasks of image processing and analysis by changing the number of processors in the PC-cluster; in the experiments performed, a 2× speedup was obtained with the best cluster configuration found.

In the approach of Kegel et al. [42, 43], the Threading Building Blocks (TBB) library and the OpenMP application programming interface were adopted and compared in order to evaluate programming effort, programming style and abstraction, and runtime performance. The authors presented several implementations for systems that support shared- and distributed memory of the list mode ordered subset expectation maximization (LM OSEM) algorithm, resulting in reducing of the processing time spent on reconstruction of PET images. LS OSEM is a computationally intensive block-iterative algorithm for 3D image reconstruction. The authors concluded that the TBB library is much easier to implement than OpenMP, especially when starting a new implementation to exploit parallelism; however, they did not analyze the exact influence of the grain, the block size, or the scheduling strategy for different amounts of input data on the program performance.

The approach presented by Murphy et al. [59] consists of an optimized iterative method, self-consistent parallel imaging (SPIRiT), combined with compressed sensing for image reconstruction. This approach allows auto-calibrating parallel imaging[3] reconstructions with clinically feasible runtimes. The purpose was to achieve real-time performance via a hybrid implementation using both multi-GPU and multi-core CPUs as parallel execution platforms. Two data parallelism models, SIMD and SIMT, were exploited and optimized through Streaming SIMD Extensions (SSE) and compute unified device architecture (CUDA) instructions, respectively. Parallel GPU and CPU implementation achieved the speedup of 40× when comparing with the runtime of a sequential C++ implementation using high-performance libraries and compiled with full compiler optimization.

Domanski et al. [19] developed a cluster Web services (CWS) framework capable of taking advantage of massively parallel technologies composed of a PC-cluster[4] and GPUs.[5] This framework facilitated communication between the client and server through the Internet in order to balance and distribute the computational load. Although the framework was able to solve a wide range of scientific problems, its main application was the full reconstruction of CT images. The parallel programming languages adopted were Open Computing Language (OpenCL) and

[3] Parallel imaging is a well-established acceleration technique based on the spatial sensitivity of array receivers [59].

[4] 32 Intel Xeon CPU cores.

[5] 6 NVIDIA cards with Tesla GPU.

MPI, for the GPU architecture and the PC-cluster, respectively.

Treibig et al. [88] presented an approach to the achievement of optimal performance according to the processor specifications and different optimization levels. The authors presented a number of low-level optimizations and algorithms for a back-projection reconstruction strategy from CT data, running on multi-core processors. The implementation was based on SSE and Advanced Vector Extensions (AVX) instructions. The result of this approach was a speedup of up to 6×; however, the authors considered that further studied were needed (a) to improve the implementation performance using distributed memory, (b) to optimize and analyze the AVX kernel update, and also (c) to include the new AVX2 operations collector.

Blas et al. [9] described the performance optimization process of a modular application based on a GPU architecture using the Feldkamp, Davis, and Kress (FDK) reconstruction algorithm. However, even though the authors performed most parallelization procedures using the SIMT model, the projection decomposition step was performed using the SIMD model and the Open Multi-Processing (OpenMP) language. The experiments were conducted with different multi-GPU configurations and code optimization levels, and a speedup of up to 2× was achieved relative to the implementations discussed in their own literature review. Meng et al. [54] accelerated the FDK algorithm using MapReduce in a cloud computing environment. Map functions were used to filter and back-project subsets of projections, and Reduce function to aggregate those partial back-projections into the whole volume. The findings of this approach were the reconstruction time achieved, whose correlation with the number of nodes employed was roughly linear. Experiments showed a speedup of 10× using 200 nodes for all cases, when compared to the same code executed on a single machine.

Birk et al. [7, 8] adopted multi-GPU and multi-core as a parallel architecture in order to accelerate 3D reconstructions based on ray casting from ultrasound data. Their approach was extended to identify the ideal number of GPUs required to reconstruct high-resolution image volumes, especially when the processing load had substantially greater DRAM capacity than the CPU system. However, the approach was not able to display in real time the high-resolution images at the pre-visualization stage. The experiments took into consideration the implementation of the optimized method for both architectures: multi-core and multi-GPU. The authors emphasized that they combined SIMT and SIMD parallel programming models.

Wei et al. [96] presented a research that used a ray tracing technique to simulate retinal image formations. This approach simulated realistic light refraction through ocular structures in 3D using polygonal meshes and GPU parallel computing.

Chen et al. [11] described a novel imaging system for real clinical applications. The system could provide incremental volume reconstructions and volume rendering; it could also generate high-quality 3D ultrasound strain images in near real-time due to a GPU-based implementation. The approach achieved a 60× speedup compared to a CPU-based implementation. However, it could not provide real-time imaging because the time spent on complex data processing and data transfer was excessive.

2.1.2 *Image filtering*

Rodrigues and Bernardes [67] improved the process of speckle noise reduction for visual analysis of medical images like optical coherence tomography. The authors proposed preserving edges and other relevant features through filter expansion from 3D OCT images of the posterior segment of the human eye for the adaptive complex-diffusion filter. Their implementation was divided into an environment setup stage and four other stages that were called iteratively. CUDA kernels were considered in parallel convolutions, parallel reductions, and element-wise arithmetic operations over the inputs.

Nguyena et al. [61] presented a hybrid parallelization scheme with the aim of accelerating the NL-Means filter algorithm. In their approach, the authors divided the input 3D MRI volume into sub-volumes in order to reduce the search region at the boundary zone. Then the image was divided into superimposed images and the superposition of the search region radius. In the implementation stage, the following parallel technologies were used: MPI, multi-threading on multi-core machines and GPUs. Communication between each cluster node was enabled by using MPI. The main contributions of the authors are an approach that requires different modes of implementation and the possibility of using the MPI technology alone or in conjunction with POSIX Threads (Pthreads) and GPUs. This latter approach reduced the computational time by a factor of approximately 510 when applied to 3D medical data. On the other hand, high memory usage emerged as a drawback of this approach, with up to three times more memory required than with the original method.

Gulo et al. [34] described in their study how to use the high-performance computing CUDA-based architecture as a computational infrastructure to accelerate an algorithm for noise image removal. The parallel GPU-based implementation developed was compared against the corresponding sequential CPU-based implementation in several experiments. The parallelization of the image smoothing method based on a variational model using CUDA

architecture reduced the runtime by up to 10 times in comparison with the CPU-based implementation.

2.1.3 Image segmentation

Image segmentation is one of the most important operations of the image processing and analysis area, being responsible for identifying and delineating objects of interest in input images. In general, tasks of 3D visualization, interpolation, filtering, classification, and even registration depend heavily on the image segmentation results in order to achieve optimum performances [82, 101, 102]. There are several approaches of image segmentation based on, for example, thresholding [5, 71], clustering [29], and deformable models [72].

Daggett and Greenshields [15] designed a parallel algorithm using a PC-cluster to segment MRI images by means of automatic image classification in order to reduce the inter-process communication overhead. This parallel algorithm was based on the virtual shared memory technique, which enables processes to communicate by directly sharing data as though it existed in a global shared memory space. The main idea was to segment anatomical images in order to obtain quantitative anatomical features and geometrically shaped models of the objects under study.

In the article of Yeh and Fu [97], an approach called parallel adaptive simulated annealing was developed to assist computer-aided measurements for identifying the associated activation regions of the brain through response waveform of functional MR images. This approach was based on a coarse-grained model performed on a cluster of four PCs; it was designed using the MPI parallel programming language and the single program, multiple data stream (SPMD) data decomposition model. The purpose of this parallelism was to reduce the computational time required by the minimization of the weighted sum of the squared Euclidean distances between each input vector and the prototypes. Additionally, it was able to automatically make clinical diagnoses of schizophrenia and multiple sclerosis.

Gabriel et al. [29] suggested Gabor filtering for texture-based image segmentation of thyroid cells. This approach was based on distributed memory and exploited a PC-cluster and the current multi-core CPU architecture. The authors combined several metrics to evaluate the performance of their approach; they then used OpenMP and MPI to compare the speedup, communication overhead, the different memory systems, and the different number of threads used. The multi-core architecture achieved the highest speedups, which were up to 11× faster compared to the PC-cluster. Although the authors presumed that their computational system would be able to make medical diagnoses, their implementation did not have a module for image analysis, or even a tool for the addition of an image set combined with the related diagnosis result.

Zhuge et al. [101, 102] developed a semi-automatic segmentation method based on the fuzzy connected technique, which was implemented using a GPU architecture. Moreover, they designed a robust and efficient parallel version of Dijkstra's algorithm in a SIMD model. This new approach took advantage of the CUDA architecture, especially by supporting atomic read/write operations in the GPU global memory.

Shi et al. [83] proposed an automatic image segmentation method for medical images based on a pulse coupling neural network combined with the 2D Tsallis entropy. Stronger adaptability, high image segmentation precision, and adequate image reconstruction from CT and MR data were the main advantages of this approach. The achievement with this GPU-based approach was the rendering of 3D volume images in real time using ray tracing implemented using a SIMT model.

In the approach by Saran et al. [74], the rigid registration of magnetic resonance venography (MRV) images and magnetic resonance angiography (MRA) images based in mutual information is performed to increase the accuracy of vessels segmentation in MRI images. The unfavorable effects of Rician noise and RF inhomogeneity in the MRI, MRA, and MRV images during the vessels segmentation are removed by applying a subtraction schema where the cost function and the choice of the minimization method are executed simultaneously using multi-core and GPU.

Balla-Arabé and Gao [5] presented a new level set method (LSM) for image segmentation. The authors designed a selective entropy-based energy functional method, robust against noise, and new selective entropy external forces for the Lattice Boltzmann method (LBM). The LSM and LBM were combined and implemented on GPUs. However, LBM requires significant memory and the approach did not achieve volume image segmentation in real time. Hence, the authors identified a need for future studies to extend their approach to a GPU cluster environment.

Aitali et al. [2] exploited the performance of GPU to accelerate a Bias Field Correction Fuzzy C-Means algorithm used for segmenting MR images. This approach was applied to correct the inhomogeneity intensity and segment the input images simultaneously. However, the expensive computation required by the algorithm demanded optimization strategies in order to reduce the runtime; hence, the authors adopted the SIMD architecture to model their approach. The GPU implementation achieved about 52× speedup relative to the CPU implementation and consisted of a novel SIMD architecture for bias field estimation and image segmentation.

Heras et al. [36] used GPU features to accelerate the Fast Two-Cycle method, which is a level set-based segmentation method. In their approach, they aimed to divide the active domain into fixed-size tiles and therefore intensively use shared memory space, resulting in a low latency close to that of the register space. Although the authors did not use real images, they measured the performance of their approach using a set of realistic MRI data volumes produced by an MRI simulator. The volumes produced by this simulator are available to be downloaded at the BrainWeb Simulated Brain Database[6], and they have been broadly used in other published articles. In the experiments, the GPU approach achieved about 6× speedup relative to the CPU implementation.

2.1.4 Image registration

Image registration is a computational task that establishes a common geometric reference frame across two or more image datasets; it is required, for example, in the comparison or fusion of image data obtained at different times or using different imaging modalities or devices [65, 68]. Intensity-based registration techniques are accurate, efficient, and robust; in addition, they depend on the interpolation scheme, search space, a similarity metric, and an optimization approach [92]. Consequently, these techniques are based on geometric transformations [12], optimization algorithms [92], and measures of similarity [17, 26].

The mutual information-based (MI-based) deformable registration algorithm was considered promising by Dandekar and Shekhar [17], mainly because it was able to correct the misalignment of tissue in CT slice images. The authors demonstrated a registration accuracy comparable to one achieved by a group of clinical experts [17, 95]. Computationally, MI-based registration is extremely intensive and so requires several thousand of iterations, with the precise number depending on the degree of the initial misalignment, the transformation complexity, the image content, and the optimization algorithm used to maximize the MI function. In order to reduce the runtime on the order of minutes or seconds, and thereby become suitable for clinical routine use, MI-based algorithms have been accelerated in parallel architectures such as clusters [12, 30], GPU [30, 55, 81], multi-core cell broadband engine architecture (CBEA) [69], and field programmable gate array (FPGA) [17].

Christensen [12] developed a 3D linear elastic transformation model using an SGI Challenge parallel computer in order to generate global non-rigid deformations of template image volumes. This approach was optimized to maximize the ratio of computation to the parallelization overhead. In this research, parallel overhead consisted of the runtimes for creating processes, starting and ending parallel regions, and running extra code required for parallelization. The authors performed experiments using implementations optimized for MasPar (SIMD) and Challenge (multiple instruction, multiple data (MIMD)) parallel architectures. The MIMD parallel programming model achieved speeds of up to 20× greater than the SIMD model.

Warfield et al. [95] presented a new registration algorithm that identifies features in image scans which need to be aligned and find the transform that minimizes the mismatch of corresponding tissue labels. This approach was implemented on a parallel platform in order to conform to a clinically acceptable timeframe. The authors adopted a multi-core PC-cluster and the MPI language as the high-performance computational infrastructure to perform the experiments; their approach was designed based on the MIMD-based parallel programming model.

Rohlfing and Maurer [68] solved problems related to the high computational efforts that are commonly incurred when non-rigid image registration techniques are used. The authors took advantage of shared-memory multiprocessor computer architectures as well as data and task partition parallel programming models. Non-rigid image registration techniques demand lengthy execution times because of the input images are usually large and because the adopted transformation model adopted requires substantially more time to compute and evaluate the similarity measure used. The experiments were performed on an SGI Origin 3800 massively parallel computer, and all the results were compared using different degrees of parallelism (2, 16, 32, and 48 threads); the performance achieved showed a reduced linear execution time.

Salomon et al. [72] presented a parallel implementation of a deformable image registration approach based on the multi-resolution technique. In this study, the authors designed their implementation by applying the MIMD parallel programming model and the OpenMP parallel programming language. However, the SIMD parallel programming model can be considered most suitable when a large number of processors are used. This parallel approach achieved a speedup of up to 10× when applied to the registration of 3D MR images.

Wachowiak and Peters [92] developed two methods—DIviding RECTangles (DIRECT) and Multi-Directional Search (MDS)—that were used to optimize a similarity metric, which is an essential component of intensity-based medical image registration algorithms. The DIRECT method was employed as a global technique for linearly bounded problems and was followed by local refinements

[6] BrainWeb Simulated Brain Database—http://www.bic.mni.mcgill.ca/brainweb.

attained with the MDS method. This approach was implemented and optimized for execution in shared memory systems. With the use of 8 or 12 CPUs on a PC-cluster, the results demonstrated efficiency gains, yielding a speedup of up to 5×.

Rehman et al. [65] employed GPU architecture to achieve high performance using the multi-resolution approach that is typically applied in non-rigid 3D image registration. In this article, the authors developed a parallel approach of non-rigid registration by regarding it as an optimal mass transport problem. The experiments showed a speedup improvement in the parallel architecture of up to 965× relative to the CPU-based implementation.

Rohrer and Gong [69] and Shams et al. [81] enabled different high-performance computing architectures to achieve real-time image registration. Rohrer and Gong [69] combined mutual information and multi-resolution techniques, and implemented them on a heterogeneous multi-core architecture called CBEA. The implementation of this approach on a GPU architecture Shams et al. [80, 81] made an innovative contribution to the computing of MI by computing joint histograms. On the basis of this approach, the registration of 3D CT, PET and MR images was achieved in real time.

Assuming relatively small nonlinear displacements and deformations in the registration of CT and MRI data related to the head, Lapeer et al. [48] presented a point-based registration method. This new method was developed in order to speed up a nonlinear multimodal registration algorithm on a GPU architecture. The approach integrated the radial basis function (RBF) as a smooth function and sought to mimic the interacting deformation of biological tissues. The performance tests demonstrated that the GPU-based implementation yielded a runtime 10× faster than that of the CPU-based implementation.

Zhu and Cochoff [99] demonstrated how to use parallel programming patterns aiming to obtain better performance in applications relating to image visualization, registration, and fusion. The parallel programming pattern used depends on the architecture adopted. Thus, it can involve data parallelism, task parallelism, coordination based on events, data sharing, asynchronous calls, and fork/join. Using multi-core and symmetric multiprocessor (SMP) architectures, the speed was up to 10× faster relative to a CPU architecture. In addition, the parallel implementation confirmed the presence of the important features of portability and flexibility.

Mafi and Sirouspour [50] developed a GPU-based computational platform for real-time analysis of soft object deformation. This GPU-based computing scheme solved a large system of linear equations and updates the nonlinear FEM matrices in real time. However, this approach can be extended to even further optimize all computations related to single- and double-precision operations. In addition, it can enable multiple GPU-based computing, deformation analysis with multiple contact points, and auto-adaptive mesh refinement in order to improve analysis accuracy.

Ellingwood et al. [26] presented a novel computation- and memory-efficient Diffeomorphic Multi-Level B-Spline Transform Composite method on GPU for the performance of non-rigid mass-preserving registration of CT volumetric images. The authors adopted the sum of squared tissue volume difference (SSTVD) as the similarity criterion to preserve the lung tissue mass; hence, SSTVD was used for computing the tissue volume. A cubic B-Spline-based free-form deformation (FFD) transformation model was employed for capturing the non-rigid deformation of objects such as human lungs. The experiments used lung CT images, which indicated a speedup of 112 times relative to the single-threaded CPU version, and of 11 times compared to the 12-threaded version when considering the average time per iteration using the GPU implementation. The authors compared the following types of algorithms: single-threaded CPU-based, multi-threaded GPU-based, and GPU-based.

3 Discussion

The deployment of high-performance computing techniques has greatly contributed to reducing the processing time of techniques used for medical image processing and analysis, making them suitable for routine clinical use. Briefly, these techniques were used in order to exploit all the computational power commonly available in modern high-computing architectures such as multi-core, GPU, and PC-cluster.

Following the recent advances in GPU [48, 50, 55, 65, 67, 80, 81, 96, 101, 102], multi-core [3, 7–9, 19, 26, 29, 59, 83, 88, 99], and FPGA [17, 56, 62, 85, 89] architectures, researchers have confirmed a trend toward lower computational costs without any consequential reduction in terms of the accuracy of the techniques of image processing and analysis. Hence, Murphy et al. [59], Shi et al. [83], Domanski et al. [19], Saran et al. [74], Alvarado et al. [4], Birk et al. [7, 8], Serrano et al. [78] designed their models using parallel programming in GPU and multi-core; on the other hand, Blas et al. [9], Tan et al. [85], Mahmoudi and Manneback [51], Cai et al. [10], Nguyen et al. [60], Riegler et al. [66] have demonstrated an approach which is more focused on load balancing techniques, multi-GPU, GPU, and multi-core architectures. Therefore, there is an increasing number of methodologies that achieve high performance levels and that combine parallel programming methods and high-performance computing architectures;

furthermore, the runtime and energy consumption required by these methodologies are decreasing considerably.

The articles evaluated in this review provide an overview on techniques of medical image processing and analysis accelerated by high-performance computing solutions. Figure 1 shows that the majority of the selected articles were published in the last decade and that the last five years have seen remarkable progress thanks to multi-core processors and GPU architecture [23]. It is important to highlight that this review covers papers published up to March 2017.

Although the articles listed in Table 4 report on highly positive speedup findings, it is important to analyze these results carefully. The majority of the selected articles indicated speedup as the main metric used to evaluate the performance gain. Almost half of the articles compared sequential and parallel implementations, as can be seen in Rohlfing and Maurer [68], Dandekar and Shekhar [17], Yeh and Fu [97], Rehman et al. [65], Rohrer and Gong [69], Zhuge et al. [100], Shams et al. [80, 81], Gabriel et al. [29], Lapeer et al. [48], Zhu and Cochoff [99], Murphy et al. [59], Shi et al. [83], Birk et al. [7, 8], Blas et al. [9], Mafi and Sirouspour [50], Meng [55]. One of the greatest challenges in this sort of comparison is to describe how well sequential implementation was optimized, and more particularly: (1) whether the SSE instruction set was used; (2) whether the code was compiled in 32 or 64 bits; and (3) whether 32- or 64-bit floating point operations were used. This sort of optimization is critical when comparing implementations that use multi-core, GPU, or cluster architectures. Usually, it is necessary to rewrite code in order to improve application performance and so exploit the benefit of parallelization. As a result, it is good practice to divide an application into smaller tasks that can be executed in parallel [33]. However, during task deconstruction, the communication process and the general coordination of processing jobs among the processors used need to be taken into account.

When adopting a parallel programming design, two main features must be taken into account: (1) the parallel architecture and (2) the type of processor communication [63]. The high computational costs of data access and task performance are dependent on the computational resources available to the computing system. Hence, parallel design should make use of data decomposition and allocate available memory efficiently.

Most of the analyzed articles focused on the parallelizing of techniques of medical image reconstruction and registration. PC-clusters are the parallel infrastructure most often adopted by researchers [15, 20, 47, 71, 72, 95, 97], FPGA [17, 56, 62, 85, 89], in addition to the most recent GPU-based technologies [5, 48, 50, 55, 65, 80, 81, 96, 101, 102] and multi-core [3, 7–9, 19, 26, 29, 59, 83, 88, 99] architectures. Moreover, it is clear that the research topic discussed in this review is recent and promising, as confirmed by the remarkable increase in the number of related scientific articles published in the last decade. In summary, the reviewed articles demonstrated a reduction in the runtime, including in real time, which is ideal for routine medical applications. However, just a few of the selected articles focused on speeding up techniques of medical image segmentation, which suggests a potential topic for further research.

This article presents a concise and up-to-date review of techniques of medical image processing and analysis that have been implemented based on high-performance computing solutions. As a result, related researchers can identify: (a) the GPUs as computing systems, (b) the SIMD as the main parallel programming model, that have been most widely used to deal with the typical demands of techniques of medical image processing and analysis. The most used computing systems are presented in Fig. 2. In particular, this review also reveals that data-parallel computations with high arithmetic intensity are well suited to SIMD parallelization; then, it is well suited for the computation on GPUs. This is because the execution model of GPUs is based on SIMD parallel programming model, which allows

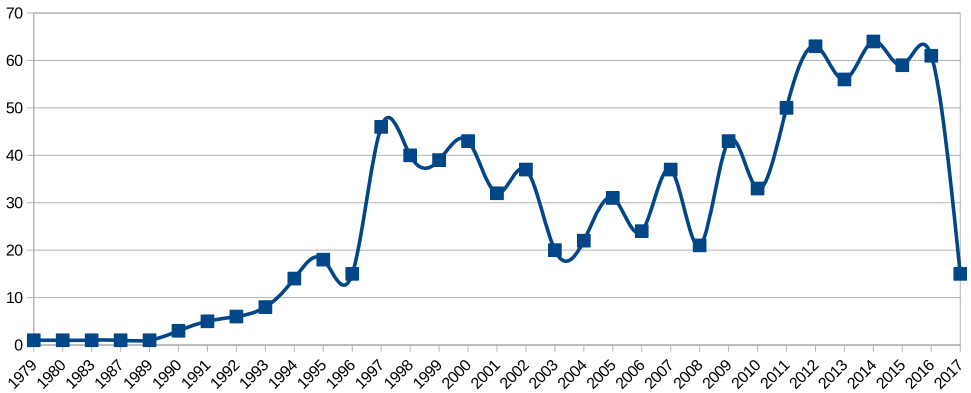

Fig. 1 Distribution of selected articles related to techniques of medical image processing and analysis accelerated by high-performance computing solutions published in recent years

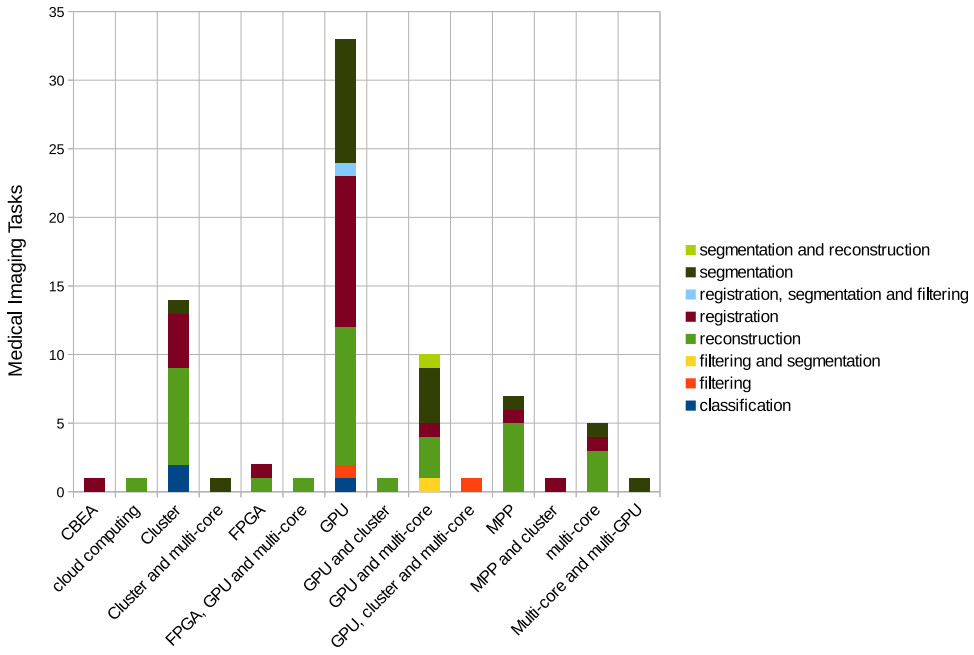

Fig. 2 Main parallel programming models applied to accelerate tasks of medical image processing and analysis

multiple processing elements to perform the same operation on multiple data, concurrently.

The greatest programming efforts found in the selected articles are: (a) the learning curve required for programming parallel implementations, (b) obtaining a complete understanding of the advanced concepts related to memory hierarchy, (c) and the design of the shortest-possible, optimal data paths.

Usually, modifying the design of a sequential algorithm in order to make it parallel requires changing the programming model, the programming language, and the memory access strategy. Successful implementation of these changes will also achieve maximum performance and a higher optimization level due to lower throughput across different memory types.

4 Conclusion

In this article, the main research articles relating to the combination of techniques of medical image processing and analysis with different high-performance computing solutions have been reviewed. The selected articles describe the use of high-performance computing systems, including multi-core, GPU, FPGA, and PC-cluster, and their capacity to support tasks of medical image processing and analysis.

This article reviewed a set of articles related to complex techniques of medical image processing and analysis, and experiments performed using high-performance computing systems. By combining parallel computer solutions with algorithms of medical image processing and analysis, the scientific community is able to make significant advances in the field of medicine, especially by reducing the required runtime; this in turn enables solutions to be implemented in routine clinical scenarios. Moreover, this article will be useful in developing new research that evaluates and compares different algorithms of medical image processing and analysis supported by high-performance computing solutions.

GPUs are considered to be extremely fast processors, especially when used in computational systems like multi-GPU. On the other hand, the use of multiple GPUs has presented additional challenges; for instance, regarding the efficient management of reading and/or writing data on the data store system, time-consuming data transfers between the CPU and GPU, and load balancing. The main issue in shared memory systems is that data must be protected against simultaneous access so that errors and data inconsistency can be avoided; additionally, the number of parallel tasks must be at least the same number of processing units (cores), and each task must have enough memory for its computing requirements.

Acknowledgements The first author would like to thank the Universidade do Estado de Mato Grosso (UNEMAT), in Brazil, and the National Scientific and Technological Development Council (“Conselho Nacional de Desenvolvimento Cientí?fico e Tecnológico”—CNPq), process 234306/2014-9, grant with reference #2010/15691-0, for the support given. The authors gratefully acknowledge the funding received from Project NORTE-01-0145-FEDER-000022—SciTech—Science and Technology for Competitive and Sustainable Industries, co-financed by “Programa Operacional Regional do Norte” (NORTE2020), through “Fundo Europeu de Desenvolvimento Regional” (FEDER).

References

1. Adeshina, A.M., Hashim, R., Khalid, N.E.A., Abidin, S.Z.Z.: Locating abnormalities in brain blood vessels using parallel computing architecture. Interdiscip. Sci.-Comput. Life Sci. **4**(3), 161–172 (2012). https://doi.org/10.1007/s12539-012-0132-y
2. Aitali, N., Cherradi, B., Abbassi, A.E., Bouattane, O., Youssfi, M.: Parallel implementation of bias field correction fuzzy c-means algorithm for image segmentation. Int. J. Adv. Comput. Sci. Appl. **7**(3), 375–383 (2016)
3. Akgun, D., Sakoglu, U., Esquivel, J., Adinoff, B., Mete, M.: GPU accelerated dynamic functional connectivity analysis for functional MRI data. Comput. Med. Imaging Graph. **43**, 53–63 (2015). https://doi.org/10.1016/j.compmedimag.2015.02.009
4. Alvarado, R., Tapia, J.J., Rolon, J.C.: Medical image segmentation with deformable models on graphics processing units. J. Supercomput. **68**(1), 339–364 (2014). https://doi.org/10.1007/s11227-013-1042-4
5. Balla-Arabé, S., Gao, X.: Geometric active curve for selective entropy optimization. Neurocomputing **139**, 65–76 (2014). https://doi.org/10.1016/j.neucom.2013.09.058
6. Barros, R., Van Geldermalsen, S., Boers, A., Belloum, A., Marquering, H., Olabarriaga, S.: Heterogeneous platform programming for high performance medical imaging processing. Lecture Notes in Computer Science 8374 LNCS:301–310, (2014) https://doi.org/10.1007/978-3-642-54420-0_30
7. Birk, M., Dapp, R., Ruiter, N., Becker, J.: GPU-based iterative transmission reconstruction in 3D ultrasound computer tomography. J. Parallel Distrib. Comput. **74**(1), 1730–1743 (2014). https://doi.org/10.1016/j.jpdc.2013.09.007
8. Birk, M., Zapf, M., Balzer, M., Ruiter, N., Becker, J.: A comprehensive comparison of GPU- and FPGA-based acceleration of reflection image reconstruction for 3D ultrasound computer tomography. J. Real-Time Image Proc. **9**(1, SI), 159–170 (2014). https://doi.org/10.1007/s11554-012-0267-4
9. Blas, J.G., Abella, M., Isaila, F., Carretero, J., Desco, M.: Surfing the optimization space of a multiple-GPU parallel implementation of a X-ray tomography reconstruction algorithm. J. Syst. Softw. **95**, 166–175 (2014). https://doi.org/10.1016/j.jss.2014.03.083
10. Cai, Y., Guo, X., Zhong, Z., Mao, W.: Dynamic meshing for deformable image registration. Comput. Aided Des. **58**(SI), 141–150 (2015). https://doi.org/10.1016/j.cad.2014.08.009
11. Chen, Z., Chen, Y., Huang, Q.: Development of a wireless and near real-time 3D ultrasound strain imaging system. IEEE Trans. Biomed. Circuits Syst. **10**(2), 394–403 (2016). https://doi.org/10.1109/TBCAS.2015.2420117
12. Christensen, G.E.: MIMD vs. SIMD parallel processing: a case study in 3D medical image registration. Parallel Comput. **24**, 1369–1383 (1998). https://doi.org/10.1016/S0167-8191(98)00062-3
13. Chung, J., Sternberg, P., Yang, C.: High-performance three-dimensional image reconstruction for molecular structure determination. Int. J. High Perform. Comput. Appl. **24**(2), 117–135 (2010). https://doi.org/10.1177/1094342009106293
14. Crane, J., Crawford, F., Nelson, S.: Grid enabled magnetic resonance scanners for near real-time medical image processing. J. Parallel Distrib. Comput. **66**(12), 1524–1533 (2006). https://doi.org/10.1016/j.jpdc.2006.03.009
15. Daggett, T., Greenshields, I.: A cluster computer system for the analysis and classification of massively large biomedical image data. Comput. Biol. Med. **28**(1), 47–60 (1998). https://doi.org/10.1016/S0010-4825(97)00032-2
16. D'Amore, L., Casaburi, D., Marcellino, L., Murli, A.: Numerical solution of diffusion models in biomedical imaging on multicore processors. Int. J. BioMed. Imaging **2011**(1), 1–16 (2011). http://doi.org/10.1155/2011/680765
17. Dandekar, O., Shekhar, R.: FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions. IEEE Trans. Biomed. Circuits Syst. **1**(2), 116–127 (2007). https://doi.org/10.1109/TBCAS.2007.909023
18. Deng, J., Yu, H., Ni, J., He, T., Zhao, S., Wang, L., Wang, G.: A parallel implementation of the Katsevich algorithm for 3-D CT image reconstruction. J. Supercomput. **38**(1), 35–47 (2006). https://doi.org/10.1007/s11227-006-6675-0
19. Domanski, L., Bednarz, T., Gureyev, T., Murray, L., Huang, B.E., Nesterets, Y., Thompson, D., Jones, E., Cavanagh, C., Wang, D., Vallotton, P., Sun, C., Khassapov, A., Stevenson, A., Mayo, S., Morell, M., George, A.W., Taylor, J.A.: Applications of heterogeneous computing in computational and simulation science. Int. J. Comput. Sci. Eng. **8**(3), 240–252 (2013)
20. Doyley, M., Van Houten, E., Weaver, J., Poplack, S., Duncan, L., Kennedy, F., Paulsen, K.: Shear modulus estimation using parallelized partial volumetric reconstruction. IEEE Trans. Med. Imaging **23**(11), 1404–1416 (2004). https://doi.org/10.1109/TMI.2004.834624
21. Du, X., Dang, J., Wang, Y., Wang, S., Lei, T.: A parallel non-rigid registration algorithm based on b-spline for medical images. Comput. Math. Methods Med. **2016**(1), 1–14 (2016). http://doi.org/10.1155/2016/7419307
22. Eidheim, O., Skjermo, J., Aurdal, L.: Real-time analysis of ultrasound images using GPU. In: Lemke, H., Inamura, K., Doi, K., Vannier, M., Farman, A. (eds) CARS 2005: Computer Assisted Radiology and Surgery, International Congress Series, vol. 1281, pp. 284–289, https://doi.org/10.1016/j.ics.2005.03.187, 19th International Congress and Exhibition on Computer Assisted Radiology and Surgery (2005)
23. Eklund, A., Dufort, P., Forsberg, D., LaConte, S.M.: Medical image processing on the GPU-past, present and future. Med. Image Anal. **17**(8), 1073–1094 (2013). https://doi.org/10.1016/j.media.2013.05.008
24. Eklund, A., Dufort, P., Villani, M., LaConte, S.: BROCCOLI: software for fast fMRI analysis on many-core CPUs and GPUs. Front. Neuroinform. **8**(24), 1–19 (2014). https://doi.org/10.3389/fninf.2014.00024
25. El-Moursy, A.A., ElAzhary, H., Younis, A.: High-accuracy hierarchical parallel technique for hidden markov model-based 3D magnetic resonance image brain segmentation. Concurr. Comput.-Pract. Exp. **26**(1), 194–216 (2014). https://doi.org/10.1002/cpe.2959
26. Ellingwood, N.D., Yin, Y., Smith, M., Lin, C.L.: Efficient methods for implementation of multi-level nonrigid mass-preserving image registration on GPUs and multi-threaded CPUs. Comput. Methods Programs Biomed. **127**, 290–300 (2016). https://doi.org/10.1016/j.cmpb.2015.12.018
27. Fan, Z., Xie, Y.: A block-wise approximate parallel implementation for ART algorithm on CUDA-enabled GPU. Biomed. Mater. Eng. **26**(1), S1027–S1035 (2015). https://doi.org/10.3233/BME-151398
28. Formiconi, A., Passeri, A., Guelfi, M., Masoni, M., Pupi, A., Meldolesi, U., Malfetti, P., Calori, L., Guidazzoli, A.: World wide web interface for advanced SPECT reconstruction algorithms implemented on a remote massively parallel computer. Int. J. Med. Inform. **47**, 125–138 (1997). https://doi.org/10.1016/S1386-5056(97)00089-0
29. Gabriel, E., Venkatesan, V., Shah, S.: Towards high performance cell segmentation in multispectral fine needle aspiration cytology of thyroid lesions. Comput. Methods Programs Biomed. **98**(3), 231–240 (2010). https://doi.org/10.1016/j.cmpb.2009.07.008

30. Gallea, R., Ardizzone, E., Pirrone, R., Gambino, O.: Three-dimensional fuzzy kernel regression framework for registration of medical volume data. Pattern Recognit. **46**(11), 3000–3016 (2013). https://doi.org/10.1016/j.patcog.2013.03.025
31. Gao, Y., Yang, J., Xu, X., Shi, F.: Efficient cellular automaton segmentation supervised by pyramid on medical volumetric data and real time implementation with graphics processing unit. Expert Syst. Appl. **38**(6), 6866–6871 (2011). https://doi.org/10.1016/j.eswa.2010.12.049
32. Gates, M., Heath, M.T., Lambros, J.: High-performance hybrid CPU and GPU parallel algorithm for digital volume correlation. Int. J. High Perform. Comput. Appl. **29**(1, SI), 92–106 (2015). https://doi.org/10.1177/1094342013518807
33. Gebali, F.: Algorithms and Parallel Computing. Wiley, London (2011)
34. Gulo, C.A.S.J., de Arruda, H.F., de Araujo, A.F., Sementille, A.C., Tavares, J.M.R.S.: Efficient parallelization on gpu of an image smoothing method based on a variational model. J. Real-Time Image Proc. (2016). https://doi.org/10.1007/s11554-016-0623-x
35. Hamdaoui, F., Sakly, A., Mtibaa, A.: FPGA implementation of particle swarm optimization based on new fitness function for MRI images segmentation. Int. J. Imaging Syst. Technol. **25**(2), 139–147 (2015). https://doi.org/10.1002/ima.22130
36. Heras, J.L.R.D.B., Arguello, F., Kainmueller, D., Zachow, S., Boo, M.: GPU-accelerated level-set segmentation. J. Real-Time Image Proc. **12**(1), 15–29 (2016). https://doi.org/10.1007/s11554-013-0378-6
37. Higgins, W.E., Swift, R.D.: Distributed system for processing 3D medical images. Comput. Biol. Med. **27**(2), 97–115 (1997). https://doi.org/10.1016/S0010-4825(96)00042-X
38. Hu, J., Zhao, X., Zhang, H.: A GPU-based multi-resolution approach to iterative reconstruction algorithms in X-ray 3D dual spectral computed tomography. Neurocomputing **215**(SI), 71–81 (2016). https://doi.org/10.1016/j.neucom.2016.01.115
39. Jaros, M., Strakos, P., Karasek, T., Riha, L., Vasatova, A., Jarogova, M., Kozubek, T.: Implementation of K-means segmentation algorithm on Intel Xeon Phi and GPU: application in medical imaging. Adv. Eng. Softw. **103**, 21–28 (2017). https://doi.org/10.1016/j.advengsoft.2016.05.008
40. Johnsen, S.F., Taylor, Z.A., Clarkson, M.J., Hipwell, J., Modat, M., Eiben, B., Han, L., Hu, Y., Mertzanidou, T., Hawkes, D.J., Ourselin, S.: NiftySim: a GPU-based nonlinear finite element package for simulation of soft tissue biomechanics. Int. J. Comput. Assist. Radiol. Surg. **10**(7), 1077–1095 (2015). https://doi.org/10.1007/s11548-014-1118-5
41. Kalmoun, E.M., Kostler, H., Rude, U.: 3D optical flow computation using a parallel variational multigrid scheme with application to cardiac C-arm CT motion. Image Vis. Comput. **25**(9), 1482–1494 (2007). https://doi.org/10.1016/j.imavis.2006.12.017
42. Kegel, P., Schellmann, M., Gorlatch, S.: Using OpenMP vs. threading building blocks for medical imaging on multi-cores. Lecture Notes in Computer Science 5704 LNCS:654–665, (2009) https://doi.org/10.1007/978-3-642-03869-3_62
43. Kegel, P., Schellmann, M., Gorlatch, S.: Comparing programming models for medical imaging on multi-core systems. Concurr. Comput.-Pract. Exp. **23**(10), 1051–1065 (2011). https://doi.org/10.1002/cpe.1671
44. Kerr, J.P., Bartlett, E.B.: Medical image-processing utilizing neural networks trained on a massively-parallel computer. Comput. Biol. Med. **25**(4), 393–403 (1995). https://doi.org/10.1016/0010-4825(95)00017-X
45. Kirk, D., Hwu, W.M.: Programming Massively Parallel Processors: A Hands-on Approach. Elsevier, Amsterdam (2010)
46. Koestler, H., Stuermer, M., Pohl, T.: Performance engineering to achieve real-time high dynamic range imaging. J. Real-Time Image Proc. **11**(1), 127–139 (2016). https://doi.org/10.1007/s11554-012-0312-3
47. Kumar, V., Rutt, B., Kurc, T., Catalyurek, U., Pan, T., Chow, S., Lamont, S., Martone, M., Saltz, J.: Large-scale biomedical image analysis in grid environments. IEEE Trans. Inf Technol. Biomed. **12**(2), 154–161 (2008). https://doi.org/10.1109/TITB.2007.908466
48. Lapeer, R.J., Shah, S.K., Rowland, R.S.: An optimised radial basis function algorithm for fast non-rigid registration of medical images. Comput. Biol. Med. **40**(1), 1–7 (2010). https://doi.org/10.1016/j.compbiomed.2009.10.002
49. Lee, D., Dinov, I., Dong, B., Gutman, B., Yanovsky, I., Toga, A.W.: CUDA optimization strategies for compute- and memory-bound neuroimaging algorithms. Comput. Methods Programs Biomed. **106**(3), 175–187 (2012). https://doi.org/10.1016/j.cmpb.2010.10.013
50. Mafi, R., Sirouspour, S.: GPU-based acceleration of computations in nonlinear finite element deformation analysis. Int. J. Numer. Methods Biomed. Eng. **30**(3), 365–381 (2014). https://doi.org/10.1002/cnm.2607
51. Mahmoudi, S., Manneback, P.: Multi-CPU/multi-GPU based framework for multimedia processing. In: IFIP Advances in Information and Communication Technology, vol. 456, 54–65 (2015). https://doi.org/10.1007/978-3-319-19578-0_5
52. Melo, R., Falcao, G., Barreto, J.: Real-time HD image distortion correction in heterogeneous parallel computing systems using efficient memory access patterns. J. Real-Time Image Proc. **11**(1), 83–91 (2016). https://doi.org/10.1007/s11554-012-0304-3
53. Melvin, C., Xu, M., Thulasiraman, P.: HPC for iterative image reconstruction in CT, vol. 273, pp. 61–68 (2008). https://doi.org/10.1145/1370256.1370265
54. Meng, B., Pratx, G., Xing, L.: Ultrafast and scalable cone-beam CT reconstruction using MapReduce in a cloud computing environment. Med. Phys. **38**(12), 6603–6609 (2011). https://doi.org/10.1118/1.3660200
55. Meng, L.: Acceleration method of 3D medical images registration based on compute unified device architecture. Bio-Med. Mater. Eng. **24**(1), 1109–1116 (2014). https://doi.org/10.3233/BME-130910
56. Mertes, J.G., Marranghello, N., Pereira, A.S.: Real-time module for digital image processing developed on a FPGA. Int. Fed. Autom. Control Proc. Volumes **46**(28), 405–410 (2013). https://doi.org/10.3182/20130925-3-CZ-3023.00072
57. Miller, M., Butler, C.: 3D maximum a posteriori estimation for single photon emission computed tomography on massively-parallel computers. IEEE Trans. Med. Imaging **12**(3), 560–565 (1993). https://doi.org/10.1109/42.241884
58. Moyano-Avila, E., Orozco-Barbosa, L., Quiles, F.J.: Parallel algorithms based on the temporal-window method for non-alternating 3D-WT over angiographies using a multicomputer. J. Signal Process. Syst. Signal Image Video Technol. **55**(1–3), 267–279 (2009). https://doi.org/10.1007/s11265-008-0188-4
59. Murphy, M., Alley, M., Demmel, J., Keutzer, K., Vasanawala, S., Lustig, M.: Fast l1 -SPIRiT compressed sensing parallel imaging MRI: scalable parallel implementation and clinically feasible runtime. IEEE Trans. Med. Imaging **31**(6), 1250–1262 (2012). https://doi.org/10.1109/TMI.2012.2188039
60. Nguyen, T.A., Nakib, A., Nguyen, H.N.: Medical image denoising via optimal implementation of non-local means on hybrid parallel architecture. Comput. Methods Programs Biomed. **129**, 29–39 (2016). https://doi.org/10.1016/j.cmpb.2016.02.002
61. Nguyena, T.A., Nakib, A., Nguyen, H.N.: Medical image denoising via optimal implementation of non-local means on

hybrid parallel architecture. Comput. Methods Programs Biomed. **129**, 29–39 (2016). https://doi.org/10.1016/j.cmpb.2016.02.002
62. Nieto, A., Brea, V., Vilariño, D.L., Osorio, R.R.: Performance analysis of massively parallel embedded hardware architectures for retinal image processing. EURASIP J. Image Video Process. **10**(1), 1–17 (2011). https://doi.org/10.1186/1687-5281-2011-10
63. Page, D.: A Practical Introduction to Computer Architecture. Springer, Berlin (2009). https://doi.org/10.1007/978-1-84882-256-6
64. Pang, W.M., Choi, K.S., Qin, J.: Fast gabor texture feature extraction with separable filters using GPU. J. Real-Time Image Proc. **12**(1), 5–13 (2016). https://doi.org/10.1007/s11554-013-0373-y
65. Rehman, T., Haber, E., Pryor, G., Melonakos, J., Tannenbaum, A.: 3D nonrigid registration via optimal mass transport on the GPU. Med. Image Anal. **13**(6), 931–940 (2009). https://doi.org/10.1016/j.media.2008.10.008
66. Riegler, M., Lux, M., Griwodz, C., Spampinato, C., De Lange, T., Eskeland, S., Pogorelov, K., Tavanapong, W., Schmidt, P., Gurrin, C., Johansen, D., Johansen, H., Halvorsen, P.: Multimedia and medicine: teammates for better disease detection and survival. Association for Computing Machinery, Inc, pp. 968–977 (2016) https://doi.org/10.1145/2964284.2976760
67. Rodrigues, P., Bernardes, R.: 3-D adaptive nonlinear complex-diffusion despeckling filter. IEEE Trans. Med. Imaging **31**(12), 2205–2212 (2012). https://doi.org/10.1109/TMI.2012.2211609
68. Rohlfing, T., Maurer, J.C.R.: Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. IEEE Trans. Inf Technol. Biomed. **7**(1), 16–25 (2003). https://doi.org/10.1109/TITB.2003.808506
69. Rohrer, J., Gong, L.: Accelerating 3D nonrigid registration using the cell broadband engine processor. IBM J. Res. Dev. **53**(5), 1–10 (2009). https://doi.org/10.1147/JRD.2009.5429078
70. Sabne, A., Wang, X., Kisner, S., Bouman, C., Raghunathan, A., Midkiff, S.: Model-based iterative CT image reconstruction on GPUs. In: Association for Computing Machinery, pp. 207–220 (2017) https://doi.org/10.1145/3018743.3018765
71. Saiviroonporn, P., Robatino, A., Zahajszky, J., Kikinis, R., Jolesz, F.: Real-time interactive three-dimensional segmentation. Acad. Radiol. **5**(1), 49–56 (1998). https://doi.org/10.1016/S1076-6332(98)80011-1
72. Salomon, M., Heitz, F., Perrin, G.R., Armspach, J.P.: A massively parallel approach to deformable matching of 3D medical images via stochastic differential equations. Parallel Comput. **31**(1), 45–71 (2005). https://doi.org/10.1016/j.parco.2004.12.003
73. Samant, S., Xia, J., Muyan-Oelik, P., Owens, J.: High performance computing for deformable image registration: towards a new paradigm in adaptive radiotherapy. Med. Phys. **35**(8), 3546–3553 (2008). https://doi.org/10.1118/1.2948318
74. Saran, A.N., Nar, F., Saran, M.: Vessel segmentation in MRI using a variational image subtraction approach. J. Electr. Eng. Comput. Sci. **22**(2), 499–516 (2014). https://doi.org/10.3906/elk-1206-18
75. Schellmann, M., Gorlatch, S., Meilaender, D., Koesters, T., Schaefers, K., Wuebbeling, F., Burger, M.: Parallel medical image reconstruction: from graphics processing units (GPU) to grids. J. Supercomput. **57**(2, SI), 151–160 (2011). https://doi.org/10.1007/s11227-010-0397-z
76. Schmid, J., Guitian, J.A.I., Gobbetti, E., Magnenat-Thalmann, N.: A GPU framework for parallel segmentation of volumetric images using discrete deformable models. Vis. Comput. **27**(2, SI), 85–95 (2011). https://doi.org/10.1007/s00371-010-0532-0
77. Sehellmann, M., Vörding, J., Gorlatch, S., Meiländer, D.: Cost-effective medical image reconstruction: from clusters to graphics processing units. In: Proceedings of the 5th Conference on Computing Frontiers, pp. 283–291 (2008). https://doi.org/10.1145/1366230.1366278
78. Serrano, E., Blas, J., Carretero, J.: A comparative study of an X-ray tomography reconstruction algorithm in accelerated and cloud computing systems. Concurr Comput **27**(18), 5538–5556 (2015). https://doi.org/10.1002/cpe.3599
79. Shackleford, J.A., Kandasamy, N., Sharp, G.C.: On developing b-spline registration algorithms for multi-core processors. Phys. Med. Biol. **55**(21), 6329–6351 (2010). https://doi.org/10.1088/0031-9155/55/21/001
80. Shams, R., Sadeghi, P., Kennedy, R., Hartley, R.: Parallel computation of mutual information on the GPU with application to real-time registration of 3D medical images. Comput. Methods Programs Biomed. **99**(2), 133–146 (2010). https://doi.org/10.1016/j.cmpb.2009.11.004
81. Shams, R., Sadeghi, P., Kennedy, R., Hartley, R.: A survey of medical image registration on multicore and the GPU. IEEE Signal Process. Mag. **27**(2), 50–60 (2010). https://doi.org/10.1109/MSP.2009.935387
82. Sharma, R., Sharma, A.: Segmentation methods in atherosclerosis vascular imaging. J. Inform. Med. Slov. **11**, 52–69 (2006)
83. Shi, W., Li, Y., Miao, Y., Hu, Y.: Research on the key technology of image guided surgery. Prz. Elektrotech. **88**(3B), 29–33 (2012)
84. Smistad, E., Bozorgi, M., Lindseth, F.: Fast: framework for heterogeneous medical image computing and visualization. Int. J. Comput. Assist. Radiol. Surg. **10**(11), 1811–1822 (2015). https://doi.org/10.1007/s11548-015-1158-5
85. Tan, G., Zhang, C., Wang, W., Zhang, P.: SuperDragon: a heterogeneous parallel system for accelerating 3D reconstruction of cryo-electron microscopy images. ACM Trans. Reconfig. Technol. Syst. **8**(4), 1–22 (2015). https://doi.org/10.1145/2851141.2851163
86. Tirado-Ramos, A., Sloot, P., Hoekstra, A., Bubak, M.: An integrative approach to high-performance biomedical problem solving environments on the grid. Parallel Comput. **30**(9–10), 1037–1055 (2004). https://doi.org/10.1016/j.parco.2004.07.010
87. Toennies, K.D.: Digital Image Acquisition, pp. 21–82. Springer, London (2012). https://doi.org/10.1007/978-1-4471-2751-2_2
88. Treibig, J., Hager, G., Hofmann, H.G., Hornegger, J., Wellein, G.: Pushing the limits for medical image reconstruction on recent standard multicore processors. Int. J. High Perform. Comput. Appl. **27**(2), 162–177 (2013). https://doi.org/10.1177/1094342012442424
89. Ustun, T., Iftimia, N., Ferguson, R., Hammer, D.: Real-time processing for fourier domain optical coherence tomography using a field programmable gate array. Rev. Sci. Instrum. **79**(11) (2008). https://doi.org/10.1063/1.3005996
90. Vadja, A.: Programming Many-Core Chips. Springer, Berlin (2011). https://doi.org/10.1007/978-1-4419-9739-5
91. Mei, W., Hwu, W. (eds.): GPU Computing GEMS - Emerald Edition. Morgan Kaufmann, Los Altos (2012)
92. Wachowiak, M., Peters, T.: High-performance medical image registration using new optimization techniques. IEEE Trans. Inf Technol. Biomed. **10**(2), 344–353 (2006). https://doi.org/10.1109/TITB.2006.864476
93. Wachowiak, M.P., Peters, T.M.: Parallel optimization approaches for medical image registration. Lect. Notes Comput. Sci. **3216**, 781–788 (2004)
94. Wang, X., Sabne, A., Kisner, S., Raghunathan, A., Bouman, C., Midkiff, S.: High performance model based image reconstruction. ACM **12**(2), 1–12 (2016). https://doi.org/10.1145/2851141.2851163
95. Warfield, S.K., Jolesz, F.A., Kikinis, R.: A high performance computing approach to the registration of medical imaging data.

Parallel Comput. **24**, 1345–1368 (1998). https://doi.org/10.1016/S0167-8191(98)00061-1
96. Wei, Q., Patkar, S., Pai, D.K.: Fast ray-tracing of human eye optics on graphics processing units. Comput. Methods Programs Biomed. **114**(3), 302–314 (2014). https://doi.org/10.1016/j.cmpb.2014.02.003
97. Yeh, J.Y., Fu, J.: Parallel adaptive simulated annealing for computer-aided measurement in functional MRI analysis. Expert Syst. Appl. **33**(3), 706–715 (2007). https://doi.org/10.1016/j.eswa.2006.06.018
98. Yip, H., Ahmad, I., Pong, T.: An efficient parallel algorithm for computing the gaussian convolution of multi-dimensional image data. J. Supercomput. **14**(3), 233–255 (1999). https://doi.org/10.1023/A:1008137531862
99. Zhu, Y.M., Cochoff, S.M.: Medical image viewing on multicore platforms using parallel computing patterns. IT Prof. **12**(2), 33–41 (2010). https://doi.org/10.1109/MITP.2010.62
100. Zhuge, Y., Cao, Y., Miller, R.W.: GPU accelerated fuzzy connected image segmentation by using CUDA. In: IEEE Engineering in Medicine and Biology Society, pp. 6341–6344 (2009). https://doi.org/10.1109/IEMBS.2009.5333158
101. Zhuge, Y., Cao, Y., Udupa, J.K., Miller, R.W.: Parallel fuzzy connected image segmentation on GPU. Med. Phys. **38**(7), 4365–4371 (2011). https://doi.org/10.1118/1.3599725
102. Zhuge, Y., Ciesielski, K.C., Udupa, J.K., Miller, R.W.: GPU-based relative fuzzy connectedness image segmentation. Med. Phys. **40**(1), 1–10 (2013). https://doi.org/10.1118/1.4769418
103. Zinterhof, P.: High-throughput-screening of medical image data on heterogeneous clusters. Lecture Notes in Computer Science 7116 LNCS:368–377, (2012) https://doi.org/10.1007/978-3-642-29843-1_42, cited By 0

Carlos A. S. J. Gulo is pursuing his Ph.D. degree in Informatics Engineering at the Faculty of Engineering of the University of Porto, in Portugal. He is graduated in Data Processing Technology from the Educational Foundation of Andradina (1997), and he has a specialization course in Computer Science from the Londrina State University (2000), in Brazil. He received his M.Sc. degree in Computer Science from the State University Paulista "Júlio de Mesquita Filho" (UNESP), in Brazil (2012). Carlos is a professor at Mato Grosso State University-UNEMAT since 2004, and he is also the leader of the Research Group PIXEL (image processing, computational vision and interactive applications). His main research areas are high-performance computing, computer vision, educational technology and virtual learning environments.

Antonio Carlos Sementille received his BSc degree in Computer Science from the State University Paulista "Júlio de Mesquita Filho" (UNESP), in Brazil (1988), his M.Sc. degree in Computer Science from the Federal University of São Carlos (UFSCar) (1994), and his Ph.D. from the São Paulo University (USP) (1999). Since 2010, he has been senior researcher in Advanced Interfaces and Adjunct Professor at UNESP in Bauru, São Paulo, Brazil.

João Manuel R. S. Tavares is graduated in Mechanical Engineering from the University of Porto, Portugal (1992). He also earned his M.Sc. degree and Ph.D. degree in Electrical and Computer Engineering from the University of Porto in 1995 and 2001, respectively. In 2015, he achieved the full Habitation in Mechanical Engineering from the University of Porto. He is a senior researcher and project coordinator at the Institute of Science and Innovation in Mechanical and Industrial Engineering (INEGI) and an Associate Professor at the Department of Mechanical Engineering of the Faculty of Engineering of the University of Porto (FEUP). Prof. João Tavares is co-editor of more than 40 books, co-author of more than 35 book chapters, 550 articles in international and national journals and conferences, and 3 international and 2 national patents. He has been a committee member of several international and national journals and conferences, is co-founder and co-editor of the book series "Lecture Notes in Computational Vision and Biomechanics" published by Springer, founder and Editor-in-Chief of the journal "Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization" published by Taylor & Francis, and co-founder and co-chair of the international conference series: CompIMAGE, ECCOMAS VipIMAGE, ICCEBS and BioDental. Also, he has been (co-)supervisor of several M.Sc. and Ph.D. thesis and supervisor of several post-doc projects, and has participated in many scientific projects both as researcher and as coordinator. His main research areas include computational vision, medical imaging, computational mechanics, scientific visualization, human–computer interaction and new product development. (More information can be found at: www.fe.up.pt/∼tavares)