

Carlos A. S. J. Gulo · Antonio C. Sementille ·

João Manuel R. S. Tavares

Techniques of Medical Image Processing and Analysis accelerated by High-Performance Computing: A Systematic Literature Review

Received: date / Accepted: date

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

Carlos A. S. J. Gulo
CNPq National Scientific and Technological Development Council
Research Group PIXEL - UNEMAT, Brazil
Programa Doutoral em Engenharia Informática, Instituto de Ciência e Inovação em Engenharia
Mecânica e Engenharia Industrial, Faculdade de Engenharia, Universidade do Porto, Portugal
E-mail: carlos.gulo@fe.up.pt

Antonio C. Sementille
Departamento de Ciências da Computação, Faculdade de Ciências, Universidade Estadual
Paulista-UNESP, Brazil
E-mail: sementille@fc.unesp.br

João Manuel R.S. Tavares
Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento
de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Portugal
E-mail: tavares@fe.up.pt (**corresponding author**)

application of high-performance computing solutions in order to reduce the required run-time. 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 run-time involved, and the difficulties and challenges that remain to be overcome.

Keywords Medical Imaging · Image Segmentation · Image Registration · Image Reconstruction

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 [91, 23], and medical image processing and analysis [15, 29, 88].

The area of medical image processing and analysis has contributed to significant medical advances [81, 88, 23, 101, 83, 7, 50] 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 [68, 47], Computed Tomography (CT) [57, 26], Optical Coherence Tomography (OCT) [67], Magnetic Resonance (MR) [15, 2], Functional Magnetic Resonance (fMR) [97, 3], 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 [37, 95, 97, 29]. 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 run-time. As a result, a new research area has emerged that combines computational techniques used for medical image processing and analysis [81, 88, 23] and high-performance computing solutions [101, 83, 7, 50]. 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 from them [81, 88, 23]. 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 [101, 83, 7, 50].

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 Section 2; the main findings extracted from the articles analyzed are summarized in Section 2.1; the contributions found in the selected articles and the gaps identified are presented and discussed in Section 3; finally, concluding remarks are presented in Section 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 run-time.

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

1. Duplicated references; i.e., 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 to 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.

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	2,158
Total		2,579

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.

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	2, 158	450	306	55
Total	2,579	467	594	87

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.

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%

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 to 5) defined in Section 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 2.1 presents in descending chronological order the most relevant information extracted from the 87 articles 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, speedups 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.

Research	Impact Factor	Imaging Modality(ies)	Image Task(s)	Object(s) Analyzed	Parallel Architecture(s)	Parallel Programming Model	Speedup
Miller and Butler, 1993	2.12	CT, SPECT	reconstruction	brain	Massively Parallel Processor (MPP)	SIMD	64x
Kerr and Bartlett, 1995	0.90	CT, SPECT	reconstruction	cardiac	MPP	SIMD	139x
Higgins and Swift, 1997	0.30	CT	reconstruction	cardiac	MPP	SIMD	5x
Formiconi et al., 1997	0.45	CT, SPECT	reconstruction	brain	MPP	MIMD	135x
Christensen, 1998	1.57	CT	registration	craniofacial	MPP and Cluster	SIMD and MIMD	20x
Daggett and Greenshields, 1998	7.68	MRI	classification	bladder and urethra	Cluster	SPMD	6x
Warfield et al., 1998	5.10	CT, MRI	registration	brain	Cluster	MIMD	15x
Saiviroonporn et al., 1998	2.10	CT, MRI	segmentation	bones, aorta, kidneys, skin, brain	MPP	SIMD	10x* 3
Yip et al., 1999	0.61	MRI	reconstruction	skull	Cluster	MIMD	500x
Rohlfing and Maurer, 2003	17.28	MRI, Microscopy	registration	brain and breast	MPP	MIMD	50x
Wachowiak and Peters, 2004, 2006	1.15, 3.72	MRI	registration	brain and heart	Cluster	MIMD	5x
Tirado-Ramos et al., 2004	1.84	MRI and CT	reconstruction	beast	Cluster	MIMD	3x
Doyley et al., 2004	1.84	MRE	reconstruction	beast	Cluster	MIMD	3x
Salomon et al., 2005	1.66	MRI	registration	brain	Cluster	MIMD	10x
Eidheim et al., 2005	0.91	ultrasound	segmentation	liver	GPU	SIMT	34x*
Crane et al., 2006	0.90	MRI	reconstruction	brain	Cluster	MIMD	3x
Deng et al., 2006	2.90	CT	reconstruction	shepp-Logan phantom	Cluster	MIMD	32x
Dandekar and Shekhar, 2007	4.6	CT, PET	registration	abdominal	FPGA	SIMD	30x
Yeh and Fu, 2007	1.5	fMRI	classification	brain	Cluster	MIMD and SPMD	2x
Kalmoun et al., 2007	2.7	CT	reconstruction	heart	Cluster	MIMD	28x
Kumar et al., 2008	2.22	Microscopy	reconstruction	breast	Cluster	MIMD	2x
Samant et al., 2008	12.77	4DCT	registration	lung	GPU	SIMT	56x
Schellmann et al., 2008	2.77	PET	reconstruction	lung	GPU	SIMT	7.5x

continued on next page

– continued from previous page

Research	Impact Factor	Imaging Modality(ies)	Image Task(s)	Object(s) Analyzed	Parallel Architecture(s)	Parallel Programming Model	Speedup
Melvin et al., 2008	0.66	CT	reconstruction	shepp-Logan phantom	multi-core	SIMD	30x
Kegel et al., Kegel et al., 2009, 2011	3.62, 1.14	PET	reconstruction	rats	multi-core	SPMD	3x
Rehman et al., 2009	5.62	MRI	registration	brain	GPU	SIMT	965x
Rohrer and Gong, 2009	0.37	CT, MRI	registration	abdominal	CBEA	SIMD and MIMD	13x*
Zhuge et al., Zhuge et al., 2009, 2011	2, 3.33	CT, MRI	segmentation	head	GPU	SIMT	18x*
Moyano-Avila et al., 2009	0	X-Ray	reconstruction	vessels	MPP	MIMD	15x
Chung et al., 2010	1.57	microscopy	reconstruction	viruses	GPU	SIMT	16x
Shackelford et al., 2010	14	3D CT	registration	lung	GPU	SIMT	15x*
Shams et al., Shams et al., 2010, 2010	11	CT, MRI, PET	registration	brain	GPU	SIMT	50x*
Gabriel et al., 2010	2.57	FNAC	segmentation	thyroid	Cluster and multi-core	MIMD and SIMD	11x
Lapeer et al., 2010	2.57	CT, MRI	registration	head	GPU	SIMT	10x
Zhu and Cochoff, 2010	1.42	CT, PET	registration	lung	multi-core	SPMD	2-10x
D'Amore et al., 2011	1	MRI	segmentation	skin	multi-core	SIMD	6x
Meng et al., 2011	4.5	CT	reconstruction	lung	cloud computing	MIMD	10x
Schmid et al., 2011	2.66	MRI	segmentation	bones	GPU	SIMT	70x
Schellmann et al., 2011	2.33	PET	reconstruction	mouse	GPU	SIMT	2x
Gao et al., 2011	1.16	MRI	segmentation	brain	GPU	SIMT	1440x*
Lee et al., 2012	7.8	MRI	registration	brain	GPU	SIMT	129x
Adeshina et al., 2012	1.66	MRA	reconstruction	brain	GPU	SIMT	3x
Murphy et al., 2012	22.4	MRI	reconstruction	torso	GPU and multi-core	SIMT and SIMD	40x
Zinterhof, 2012	0	CT	classification	kidney	GPU	SIMT	120x
Shi et al., 2012	0.40	CT, MRI	segmentation and reconstruction	head, breast, vessels	GPU and multi-core	SIMT and SIMD	40x*
Rodrigues and Bernardes, 2012	2	OCT	filtering	retinal	GPU	SIMT	18x*
Domanski et al., 2013	1.25	CT	reconstruction	brain	GPU and multi-core	SIMT and SIMD	9x
Treibig et al., 2013	5.75	CT, X-ray	reconstruction	rabbit	multi-core	SIMD	6x
Gallea et al., 2013	1.28	CT, MRI	registration	brain	GPU	SIMT	100x

continued on next page

– continued from previous page

Research	Impact Factor	Imaging Modality(ies)	Image Task(s)	Object(s) Analyzed	Parallel Architecture(s)	Parallel Programming Model	Speedup
Saran et al., 2014	1.33	MRI	segmentation	breast	GPU and multi-core	SIMT and SIMD	35x
El-Moursy et al., 2014	0.66	3D MRI	segmentation	brain	Cluster	MIMD	2.6x
Balla-Arabé and Gao, 2014	1.33	MRI	segmentation registration,	breast	GPU	SIMT	6x*
Eklund et al., 2014	9	fMRI	segmentation, filtering	brain	GPU	SIMT	195- 525x
Barros et al., 2014	0	CT	segmentation	brain	GPU	SIMT	36x
Alvarado et al., 2014	2.33	CT, PET, MRI	segmentation	brain	GPU and multi-core	SIMT and SIMD	8
Birk et al., 2014	7	USCT	reconstruction	breast	GPU and multi-core	MIMD	25x*
Blas et al., 2014	2	CT	reconstruction	rats	GPU and multi-core	SIMT and SIMD	2x
Mafi and Sirouspour, 2014	3.33	MRI	reconstruction	stomach	GPU	SIMT	28x*
Meng, 2014	1.33	CT	registration	thorax	GPU	SIMT	255x
Wei et al., 2014	0.33	MRI	reconstruction	eye optics shepp-Logan	GPU	SIMT	100x
Fan and Xie, 2015	0	CT	reconstruction	phantom	GPU	SIMT	20x
Serrano et al., 2015	0.5	CT	reconstruction	human body	GPU and Cluster	SIMT and MIMD	22x
Gates et al., 2015	8	CT	segmentation	brain	GPU	SIMT	43.5x
Akgun et al., 2015	1.50	fMRI	segmentation	brain	GPU and multi-core	SIMT and SIMD	157x
Tan et al., 2015	0	microscopy	reconstruction	virus	FPGA, GPU and multi-core	SIMD, SIMT and MIMD	14x
Mahmoudi and Manneback, 2015	1.50	X-ray and MRI	segmentation	vertebra	multi-core and multi-GPU	SIMD and MIMD	98x*
Johnsen et al., 2015	7.50	MRI	registration	breast	GPU	SIMT	5x
Hamdaoui et al., 2015	0	MRI	reconstruction	brain	FPGA	SIMD	37x
Cai et al., 2015	2.50	MRI	registration	lung bone	GPU and multi-core	SIMT and SIMD	4x
Smistad et al., 2015	0	CT, 3D ultrasound	filtering and segmentation	structure, retina blood vessels	GPU and multi-core	SIMD and SIMT	20x

continued on next page

– continued from previous page

Research	Impact Factor	Imaging Modality(ies)	Image Task(s)	Object(s) Analyzed	Parallel Architecture(s)	Parallel Programming Model	Speedup
Gulo et al., 2016	3	Ultrasound	filtering	stomach	GPU	SIMT	10x
Nguyena et al., 2016	4.50	MRI	filtering	brain	GPU, Cluster, and multi-core	SIMT, MIMD and SIMD	510x
Koestler et al., 2016	3.50	X-ray	reconstruction	head	GPU	SIMT	1.6x
Hu et al., 2016	1	CT	reconstruction	thorax	GPU	SIMT	202x
Du et al., 2016	0	CT, MRI	registration	brain, lung	GPU	SIMT	17x
Ellingwood et al., 2016	1	CT	registration	lung	GPU	SIMT	112x
Heras et al., 2016	2	MRI,CT	segmentation	brain	GPU	SIMT	6x
Chen et al., 2016	7	Ultrasound	reconstruction	forearm	GPU	SIMT	60x
Aitali et al., 2016	2	MRI	segmentation	skin	GPU	SIMT	52x
Riegler et al., 2016	4	endoscopy	classification	gastrointestinal	multi-core and GPU	SIMD and SIMT	10x
Pang et al., 2016	7	ultrasound	segmentation	breast	GPU	SIMT	16x
Wang et al., Sabne et al., 2016, 2017	4, 2	CT	reconstruction	lungs	GPU	SIMT	4x
Jaros et al., 2017	2	CT	segmentation	heart and liver	GPU	SIMT	44x

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

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 128x128 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 64x, relative to an optimal programmed implementation to be executed in a Reduced Instruction Set Computing (RISC) architecture (64x64 processor). Formiconi et al. [28] also 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 139x 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 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 pre-processing, 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 5x 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 2x 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 imaging⁴ reconstructions with clinically-feasible run-times. 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 40x 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-cluster⁵ and GPUs⁶. This framework facilitated communication between the client and server through the Internet in order to balance and distribute the

⁴ Parallel imaging is a well-established acceleration technique based on the spatial sensitivity of array receivers [59].

⁵ 32 Intel Xeon CPU cores

⁶ 6 NVIDIA cards with Tesla GPU

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 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 6x; however, the authors considered that further studies 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 2x 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 was the reconstruction time achieved, whose correlation with the number of nodes employed was roughly linear. Experiments showed a speedup of 10x 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 60x 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 11x 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 schemam 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 in-homogeneity intensity and segment the input images simultaneously. However, the expensive computation required by the algorithm demanded optimization strategies in order to reduce the run-time; hence, the authors adopted the SIMD architecture to model their approach. The GPU implementation achieved about 52x 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 Database⁷ and they have been broadly used in other published articles. In the experiments, the GPU approach achieved about 6x 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 [68, 65]. 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

⁷ BrainWeb Simulated Brain Database - <http://www.bic.mni.mcgill.ca/brainweb>

the MI function. In order to reduce the run-time 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 [81, 55, 30], 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 run-times 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 20x 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 10x 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 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 5x.

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 965x 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. [81, 80] 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 non-linear 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 non-linear multi-modal 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 run-time 10x 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 10x 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 non-linear 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 nonrigid 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 nonrigid 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 [65, 101, 102, 81, 80, 48, 67, 50, 55, 96], multi-core [29, 99, 59, 83, 19, 88, 7, 8, 9, 3, 26] and FPGA [17, 85, 56, 89, 62] architectures, researchers have confirmed a trend towards 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 run-time 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.

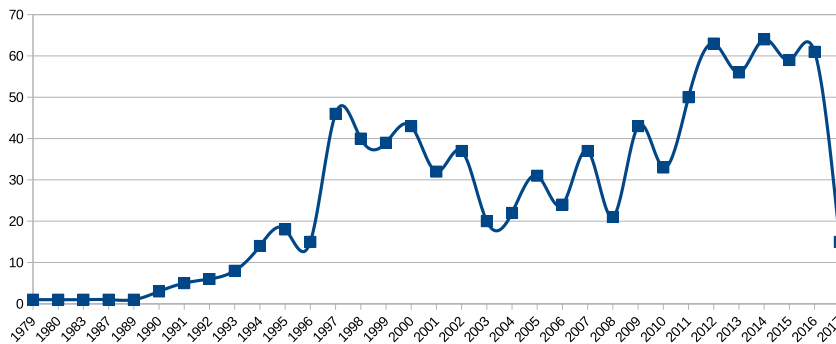


Fig. 1 Distribution of selected articles related to techniques of medical image processing and analysis accelerated by high-performance computing solutions published in recent years.

Although the articles listed in Table 2.1 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 Rohlring 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. [81, 80], 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, 95, 71, 20, 72, 97, 47], FPGA [17, 85, 56, 89, 62], in addition to the most recent GPU-based technologies [65, 101, 102, 81, 80, 48, 5, 50, 55, 96] and multi-core [29, 99, 59, 83, 19, 88, 7, 8, 9, 3, 26] 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 run-time, 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 is presented in Figure 2. In particular, this review also reveals that data-parallel computations with high arithmetic intensity is well-suited to SIMD parallelization then it is well-suited to computation on GPUs. This is

because the execution model of GPUs is based on SIMD parallel programming model, which allows 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.

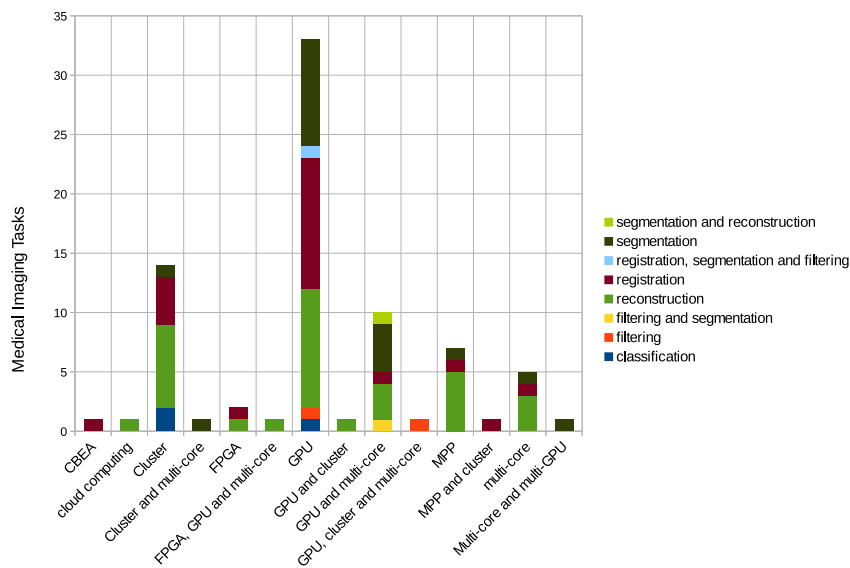


Fig. 2 Main parallel programming models applied to accelerate tasks of medical image processing and analysis.

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 run-time; 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.

5 Acknowledgments

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 Científico 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 AM, Hashim R, Khalid NEA, Abidin SZZ (2012) Locating abnormalities in brain blood vessels using parallel computing architecture. *Interdisciplinary Sciences-Computational Life Sciences* 4(3):161–172, DOI 10.1007/s12539-012-0132-y
2. Aitali N, Cherradi B, Abbassi AE, Bouattane O, Youssfi M (2016) Parallel implementation of bias field correction fuzzy c-means algorithm for image

-
- segmentation. *International Journal of Advanced Computer Science and Applications* 7(3):375–383
3. Akgun D, Sakoglu U, Esquivel J, Adinoff B, Mete M (2015) GPU accelerated dynamic functional connectivity analysis for functional MRI data. *Computerized Medical Imaging and Graphics* 43:53 – 63, DOI <http://dx.doi.org/10.1016/j.compmedimag.2015.02.009>
 4. Alvarado R, Tapia JJ, Rolon JC (2014) Medical image segmentation with deformable models on graphics processing units. *Journal of Supercomputing* 68(1):339–364, DOI 10.1007/s11227-013-1042-4
 5. Balla-Arabé S, Gao X (2014) Geometric active curve for selective entropy optimization. *Neurocomputing* 139:65–76, DOI <http://dx.doi.org/10.1016/j.neucom.2013.09.058>
 6. Barros R, Van Geldermalsen S, Boers A, Belloum A, Marquering H, Olabarriaga S (2014) Heterogeneous platform programming for high performance medical imaging processing. *Lecture Notes in Computer Science* 8374 LNCS:301–310, DOI 10.1007/978-3-642-54420-0_30
 7. Birk M, Dapp R, Ruitter N, Becker J (2014) GPU-based iterative transmission reconstruction in 3D ultrasound computer tomography. *Journal of Parallel and Distributed Computing* 74(1):1730–1743, DOI <http://dx.doi.org/10.1016/j.jpdc.2013.09.007>
 8. Birk M, Zapf M, Balzer M, Ruitter N, Becker J (2014) A comprehensive comparison of GPU- and FPGA-based acceleration of reflection image reconstruction for 3D ultrasound computer tomography. *Journal of Real-Time Image Processing* 9(1, SI):159–170, DOI 10.1007/s11554-012-0267-4

-
9. Blas JG, Abella M, Isaila F, Carretero J, Desco M (2014) Surfing the optimization space of a multiple-GPU parallel implementation of a X-ray tomography reconstruction algorithm. *Journal of Systems and Software* 95:166–175, DOI <http://dx.doi.org/10.1016/j.jss.2014.03.083>
 10. Cai Y, Guo X, Zhong Z, Mao W (2015) Dynamic meshing for deformable image registration. *Computer-Aided Design* 58(SI):141–150, DOI 10.1016/j.cad.2014.08.009
 11. Chen Z, Chen Y, Huang Q (2016) Development of a wireless and near real-time 3D ultrasound strain imaging system. *IEEE Transactions on Biomedical Circuits and Systems* 10(2):394–403, DOI 10.1109/TBCAS.2015.2420117
 12. Christensen GE (1998) MIMD vs. SIMD parallel processing: A case study in 3D medical image registration. *Parallel Computing* 24:1369–1383, DOI [http://dx.doi.org/10.1016/S0167-8191\(98\)00062-3](http://dx.doi.org/10.1016/S0167-8191(98)00062-3)
 13. Chung J, Sternberg P, Yang C (2010) High-performance three-dimensional image reconstruction for molecular structure determination. *International Journal of High Performance Computing Applications* 24(2):117–135, DOI 10.1177/1094342009106293
 14. Crane J, Crawford F, Nelson S (2006) Grid enabled magnetic resonance scanners for near real-time medical image processing. *Journal of Parallel and Distributed Computing* 66(12):1524–1533, DOI 10.1016/j.jpdc.2006.03.009
 15. Daggett T, Greenshields I (1998) A cluster computer system for the analysis and classification of massively large biomedical image data. *Computers in Biology and Medicine* 28(1):47–60, DOI [http://dx.doi.org/10.1016/S0010-0285\(98\)00062-3](http://dx.doi.org/10.1016/S0010-0285(98)00062-3)

1016/S0010-4825(97)00032-2

16. D'Amore L, Casaburi D, Marcellino L, Murli A (2011) Numerical solution of diffusion models in biomedical imaging on multicore processors. *International Journal of BioMedical Imaging* 2011, DOI 10.1155/2011/680765
17. Dandekar O, Shekhar R (2007) FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions. *IEEE Transactions on Biomedical Circuits and Systems* 1(2):116–127, DOI 10.1109/TBCAS.2007.909023
18. Deng J, Yu H, Ni J, He T, Zhao S, Wang L, Wang G (2006) A parallel implementation of the Katsevich algorithm for 3-D CT image reconstruction. *Journal of Supercomputing* 38(1):35–47, DOI 10.1007/s11227-006-6675-0
19. Domanski L, Bednarz T, Gureyev T, Murray L, Huang BE, Nesterets Y, Thompson D, Jones E, Cavanagh C, Wang D, Vallotton P, Sun C, Khassapov A, Stevenson A, Mayo S, Morell M, George AW, Taylor JA (2013) Applications of heterogeneous computing in computational and simulation science. *International Journal of Computational Science and Engineering* 8(3):240–252
20. Doyley M, Van Houten E, Weaver J, Poplack S, Duncan L, Kennedy F, Paulsen K (2004) Shear modulus estimation using parallelized partial volumetric reconstruction. *IEEE Transactions on Medical Imaging* 23(11):1404–1416, DOI 10.1109/TMI.2004.834624

-
21. Du X, Dang J, Wang Y, Wang S, Lei T (2016) A parallel nonrigid registration algorithm based on b-spline for medical images. *Computational and Mathematical Methods in Medicine* 2016, DOI 10.1155/2016/7419307
 22. Eidheim O, Skjermo J, Aurdal L (2005) 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, DOI 10.1016/j.ics.2005.03.187, 19th International Congress and Exhibition on Computer Assisted Radiology and Surgery
 23. Eklund A, Dufort P, Forsberg D, LaConte SM (2013) Medical image processing on the GPU - past, present and future. *Medical Image Analysis* 17(8):1073–1094, DOI 10.1016/j.media.2013.05.008
 24. Eklund A, Dufort P, Villani M, LaConte S (2014) BROCCOLI: Software for fast fMRI analysis on many-core CPUs and GPUs. *Frontiers in Neuroinformatics* 8, DOI 10.3389/fninf.2014.00024
 25. El-Moursy AA, ElAzhary H, Younis A (2014) High-accuracy hierarchical parallel technique for hidden markov model-based 3D magnetic resonance image brain segmentation. *Concurrency and Computation-Practice & Experience* 26(1):194–216, DOI 10.1002/cpe.2959
 26. Ellingwood ND, Yin Y, Smith M, Lin CL (2016) Efficient methods for implementation of multi-level nonrigid mass-preserving image registration on GPUs and multi-threaded CPUs. *Computer Methods and Programs in Biomedicine* 127:290 – 300, DOI <http://dx.doi.org/10.1016/j.cmpb.2015.12.018>

-
27. Fan Z, Xie Y (2015) A block-wise approximate parallel implementation for ART algorithm on CUDA-enabled GPU. *Biomedical Materials and Engineering* 26(1):S1027–S1035, DOI 10.3233/BME-151398, international Workshop on Biological Mechanics in Conjunction with the 4th International Conference on Biomedical Engineering and Biotechnology (ICBEB)
 28. Formiconi A, Passeri A, Guelfi M, Masoni M, Pupi A, Meldolesi U, Malfetti P, Calori L, Guidazzoli A (1997) World wide web interface for advanced SPECT reconstruction algorithms implemented on a remote massively parallel computer. *International Journal of Medical Informatics* 47:125–138, DOI [http://dx.doi.org/10.1016/S1386-5056\(97\)00089-0](http://dx.doi.org/10.1016/S1386-5056(97)00089-0)
 29. Gabriel E, Venkatesan V, Shah S (2010) Towards high performance cell segmentation in multispectral fine needle aspiration cytology of thyroid lesions. *Computer Methods and Programs in Biomedicine* 98(3):231–240, DOI <http://dx.doi.org/10.1016/j.cmpb.2009.07.008>
 30. Gallea R, Ardizzone E, Pirrone R, Gambino O (2013) Three-dimensional fuzzy kernel regression framework for registration of medical volume data. *Pattern Recognition* 46(11):3000–3016, DOI 10.1016/j.patcog.2013.03.025
 31. Gao Y, Yang J, Xu X, Shi F (2011) Efficient cellular automaton segmentation supervised by pyramid on medical volumetric data and real time implementation with graphics processing unit. *Expert Systems with Applications* 38(6):6866–6871, DOI 10.1016/j.eswa.2010.12.049
 32. Gates M, Heath MT, Lambros J (2015) High-performance hybrid CPU and GPU parallel algorithm for digital volume correlation. *International Journal*

-
- of High Performance Computing Applications 29(1, SI):92–106, DOI 10.1177/1094342013518807
33. Gebali F (2011) Algorithms and Parallel Computing. John Wiley & Sons
34. Gulo CASJ, de Arruda HF, de Araujo AF, Sementille AC, Tavares JMRS (2016) Efficient parallelization on gpu of an image smoothing method based on a variational model. Journal of Real-Time Image Processing DOI 10.1007/s11554-016-0623-x
35. Hamdaoui F, Sakly A, Mtibaa A (2015) FPGA implementation of particle swarm optimization based on new fitness function for MRI images segmentation. International Journal of Imaging Systems and Technology 25(2):139–147, DOI 10.1002/ima.22130
36. Heras JLRDB, Arguello F, Kainmueller D, Zachow S, Boo M (2016) GPU-accelerated level-set segmentation. Journal of Real-Time Image Processing 12(1):15–29, DOI 10.1007/s11554-013-0378-6
37. Higgins WE, Swift RD (1997) Distributed system for processing 3D medical images. Computers in Biology and Medicine 27(2):97–115, DOI [http://dx.doi.org/10.1016/S0010-4825\(96\)00042-X](http://dx.doi.org/10.1016/S0010-4825(96)00042-X)
38. Hu J, Zhao X, Zhang H (2016) A GPU-based multi-resolution approach to iterative reconstruction algorithms in X-ray 3D dual spectral computed tomography. Neurocomputing 215(SI):71–81, DOI 10.1016/j.neucom.2016.01.115
39. Jaros M, Strakos P, Karasek T, Riha L, Vasatova A, Jarogova M, Kozubek T (2017) Implementation of K-means segmentation algorithm on Intel Xeon Phi and GPU: Application in medical imaging. Advances in Engineering

-
- Software 103:21–28, DOI 10.1016/j.advengsoft.2016.05.008
40. Johnsen SF, Taylor ZA, Clarkson MJ, Hipwell J, Modat M, Eiben B, Han L, Hu Y, Mertzaniidou T, Hawkes DJ, Ourselin S (2015) NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics. *International Journal of Computer Assisted Radiology and Surgery* 10(7):1077–1095, DOI 10.1007/s11548-014-1118-5
 41. Kalmoun EM, Kostler H, Rude U (2007) 3D optical flow computation using a parallel variational multigrid scheme with application to cardiac C-arm CT motion. *Image and Vision Computing* 25(9):1482–1494, DOI 10.1016/j.imavis.2006.12.017
 42. Kegel P, Schellmann M, Gorlatch S (2009) Using OpenMP vs. threading building blocks for medical imaging on multi-cores. *Lecture Notes in Computer Science* 5704 LNCS:654–665, DOI 10.1007/978-3-642-03869-3_62
 43. Kegel P, Schellmann M, Gorlatch S (2011) Comparing programming models for medical imaging on multi-core systems. *Concurrency and Computation-Practice & Experience* 23(10):1051–1065, DOI 10.1002/cpe.1671
 44. Kerr JP, Bartlett EB (1995) Medical image-processing utilizing neural networks trained on a massively-parallel computer. *Computers in Biology and Medicine* 25(4):393–403, DOI 10.1016/0010-4825(95)00017-X
 45. Kirk D, Hwu WM (2010) *Programming Massively Parallel Processors: A Hands-on Approach*. Elsevier
 46. Koestler H, Stuermer M, Pohl T (2016) Performance engineering to achieve real-time high dynamic range imaging. *Journal of Real-Time Image*

-
- Processing 11(1):127–139, DOI 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 (2008) Large-scale biomedical image analysis in Grid environments. *IEEE Transactions on Information Technology in Biomedicine* 12(2):154–161, DOI 10.1109/TITB.2007.908466
48. Lapeer RJ, Shah SK, Rowland RS (2010) An optimised radial basis function algorithm for fast non-rigid registration of medical images. *Computers in Biology and Medicine* 40(1):1–7, DOI 10.1016/j.compbimed.2009.10.002
49. Lee D, Dinov I, Dong B, Gutman B, Yanovsky I, Toga AW (2012) CUDA optimization strategies for compute- and memory-bound neuroimaging algorithms. *Computer Methods and Programs in Biomedicine* 106(3):175–187, DOI 10.1016/j.cmpb.2010.10.013
50. Mafi R, Sirouspour S (2014) GPU-based acceleration of computations in nonlinear finite element deformation analysis. *International Journal for Numerical Methods in Biomedical Engineering* 30(3):365–381, DOI 10.1002/cnm.2607
51. Mahmoudi S, Manneback P (2015) Multi-CPU/multi-GPU based framework for multimedia processing. *IFIP Advances in Information and Communication Technology* 456:54–65, DOI 10.1007/978-3-319-19578-0_5
52. Melo R, Falcao G, Barreto J (2016) Real-time HD image distortion correction in heterogeneous parallel computing systems using efficient memory access patterns. *Journal of Real-Time Image Processing* 11(1):83–91, DOI 10.1007/s11554-012-0304-3

-
53. Melvin C, Xu M, Thulasiraman P (2008) HPC for iterative image reconstruction in CT. vol 273, pp 61–68, DOI 10.1145/1370256.1370265
 54. Meng B, Pratz G, Xing L (2011) Ultrafast and scalable cone-beam CT reconstruction using MapReduce in a cloud computing environment. *Medical Physics* 38(12):6603–6609, DOI 10.1118/1.3660200
 55. Meng L (2014) Acceleration method of 3D medical images registration based on compute unified device architecture. *Bio-medical materials and engineering* 24(1):1109–1116, DOI 10.3233/BME-130910
 56. Mertes J, Marranghello N, Pereira A (2013) Real-time module for digital image processing developed on a FPGA. vol 12, pp 405–410, DOI 10.3182/20130925-3-CZ-3023.00072
 57. Miller M, Butler C (1993) 3D maximum a posteriori estimation for single photon emission computed tomography on massively-parallel computers. *IEEE Transactions on Medical Imaging* 12(3):560–565, DOI 10.1109/42.241884
 58. Moyano-Avila E, Orozco-Barbosa L, Quiles FJ (2009) Parallel algorithms based on the temporal-window method for non-alternating 3D-WT over angiographies using a multicomputer. *Journal of Signal Processing Systems for Signal Image and Video Technology* 55(1-3):267–279, DOI 10.1007/s11265-008-0188-4
 59. Murphy M, Alley M, Demmel J, Keutzer K, Vasanawala S, Lustig M (2012) Fast l_1 -SPIRiT compressed sensing parallel imaging MRI: Scalable parallel implementation and clinically feasible runtime. *IEEE Transactions on Medical Imaging* 31(6):1250–1262, DOI 10.1109/TMI.2012.2188039

60. Nguyen TA, Nakib A, Nguyen HN (2016) Medical image denoising via optimal implementation of non-local means on hybrid parallel architecture. *Computer Methods and Programs in Biomedicine* 129:29–39, DOI 10.1016/j.cmpb.2016.02.002
61. Nguyena TA, Nakib A, Nguyen HN (2016) Medical image denoising via optimal implementation of non-local means on hybrid parallel architecture. *Computer Methods and Programs in Biomedicine* 129:29 – 39, DOI <http://dx.doi.org/10.1016/j.cmpb.2016.02.002>
62. Nieto A, Brea V, Vilarino DL, Osorio RR (2011) Performance analysis of massively parallel embedded hardware architectures for retinal image processing. *EURASIP Journal on Image and Video Processing* DOI 10.1186/1687-5281-2011-10
63. Page D (2009) *A Practical Introduction to Computer Architecture*. Springer, DOI 10.1007/978-1-84882-256-6
64. Pang WM, Choi KS, Qin J (2016) Fast gabor texture feature extraction with separable filters using GPU. *Journal of Real-Time Image Processing* 12(1):5–13, DOI 10.1007/s11554-013-0373-y
65. Rehman T, Haber E, Pryor G, Melonakos J, Tannenbaum A (2009) 3D nonrigid registration via optimal mass transport on the GPU. *Medical Image Analysis* 13(6):931–940, DOI <http://dx.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 (2016) *Multimedia and medicine: Teammates for better disease*

-
- detection and survival. Association for Computing Machinery, Inc, pp 968–977, DOI 10.1145/2964284.2976760
67. Rodrigues P, Bernardes R (2012) 3-D adaptive nonlinear complex-diffusion despeckling filter. *IEEE Transactions on Medical Imaging* 31(12):2205–2212, DOI 10.1109/TMI.2012.2211609
68. Rohlfing T, Maurer J CR (2003) Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. *IEEE Transactions on Information Technology in Biomedicine* 7(1):16–25, DOI 10.1109/TITB.2003.808506
69. Rohrer J, Gong L (2009) Accelerating 3D nonrigid registration using the cell broadband engine processor. *IBM Journal of Research and Development* 53(5), DOI 10.1147/JRD.2009.5429078
70. Sabne A, Wang X, Kisner S, Bouman C, Raghunathan A, Midkiff S (2017) Model-based iterative CT image reconstruction on GPUs. Association for Computing Machinery, pp 207–220, DOI 10.1145/3018743.3018765
71. Saiviroonporn P, Robotino A, Zahajszky J, Kikinis R, Jolesz F (1998) Real-time interactive three-dimensional segmentation. *Academic Radiology* 5(1):49–56, DOI 10.1016/S1076-6332(98)80011-1
72. Salomon M, Heitz F, Perrin GR, Armspach JP (2005) A massively parallel approach to deformable matching of 3D medical images via stochastic differential equations. *Parallel Computing* 31(1):45–71, DOI <http://dx.doi.org/10.1016/j.parco.2004.12.003>
73. Samant S, Xia J, Muyan-Oelik P, Owens J (2008) High performance computing for deformable image registration: Towards a new paradigm in

- adaptive radiotherapy. *Medical Physics* 35(8):3546–3553, DOI 10.1118/1.2948318
74. Saran AN, Nar F, Saran M (2014) Vessel segmentation in MRI using a variational image subtraction approach. *Journal of Electrical Engineering and Computer Sciences* 22(2):499–516, DOI 10.3906/elk-1206-18
75. Schellmann M, Gorchach S, Meilaender D, Koesters T, Schaefer K, Wuebbeling F, Burger M (2011) Parallel medical image reconstruction: from graphics processing units (GPU) to grids. *Journal of Supercomputing* 57(2, SI):151–160, DOI 10.1007/s11227-010-0397-z
76. Schmid J, Guitian JAI, Gobbetti E, Magnenat-Thalmann N (2011) A GPU framework for parallel segmentation of volumetric images using discrete deformable models. *Visual Computer* 27(2, SI):85–95, DOI 10.1007/s00371-010-0532-0
77. Schellmann M, Vörding J, Gorchach S, Meiländer D (2008) Cost-effective medical image reconstruction: From clusters to graphics processing units. pp 283–291, DOI 10.1145/1366230.1366278
78. Serrano E, Blas J, Carretero J (2015) A comparative study of an X-ray tomography reconstruction algorithm in accelerated and cloud computing systems. *Concurrency Computation* 27(18):5538–5556, DOI 10.1002/cpe.3599
79. Shackleford JA, Kandasamy N, Sharp GC (2010) On developing b-spline registration algorithms for multi-core processors. *Physics in Medicine and Biology* 55(21):6329–6351, DOI 10.1088/0031-9155/55/21/001

-
80. Shams R, Sadeghi P, Kennedy R, Hartley R (2010) Parallel computation of mutual information on the GPU with application to real-time registration of 3D medical images. *Computer Methods and Programs in Biomedicine* 99(2):133–146, DOI 10.1016/j.cmpb.2009.11.004
 81. Shams R, Sadeghi P, Kennedy R, Hartley R (2010) A survey of medical image registration on multicore and the GPU. *IEEE Signal Processing Magazine* 27(2):50–60, DOI 10.1109/MSP.2009.935387
 82. Sharma R, Sharma A (2006) Segmentation methods in atherosclerosis vascular imaging. *Journal Informatica Medica Slovenica* 11:52–69
 83. Shi W, Li Y, Miao Y, Hu Y (2012) Research on the key technology of image guided surgery. *Przegląd Elektrotechniczny* 88(3B):29–33
 84. Smistad E, Bozorgi M, Lindseth F (2015) Fast: framework for heterogeneous medical image computing and visualization. *International Journal of Computer Assisted Radiology and Surgery* 10(11):1811–1822, DOI 10.1007/s11548-015-1158-5
 85. Tan G, Zhang C, Wang W, Zhang P (2015) SuperDragon: A heterogeneous parallel system for accelerating 3D reconstruction of cryo-electron microscopy images. *ACM Transactions on Reconfigurable Technology and Systems* 8(4), DOI 10.1145/2740966
 86. Tirado-Ramos A, Sloot P, Hoekstra A, Bubak M (2004) An integrative approach to high-performance biomedical problem solving environments on the grid. *Parallel Computing* 30(9-10):1037–1055, DOI 10.1016/j.parco.2004.07.010

87. Toennies KD (2012) *Digital Image Acquisition*, Springer London, London, pp 21–82. DOI 10.1007/978-1-4471-2751-2_2
88. Treibig J, Hager G, Hofmann HG, Hornegger J, Wellein G (2013) Pushing the limits for medical image reconstruction on recent standard multicore processors. *International Journal of High Performance Computing Applications* 27(2):162–177, DOI 10.1177/1094342012442424
89. Ustun T, Iftimia N, Ferguson R, Hammer D (2008) Real-time processing for fourier domain optical coherence tomography using a field programmable gate array. *Review of Scientific Instruments* 79(11), DOI 10.1063/1.3005996
90. Vadja A (2011) *Programming Many-Core Chips*. Springer, DOI 10.1007/978-1-4419-9739-5
91. mei W Hwu W (ed) (2012) *GPU Computing GEMS - Emerald Edition*. Morgan Kaufmann
92. Wachowiak M, Peters T (2006) High-performance medical image registration using new optimization techniques. *IEEE Transactions on Information Technology in Biomedicine* 10(2):344–353, DOI 10.1109/TITB.2006.864476
93. Wachowiak MP, Peters TM (2004) Parallel optimization approaches for medical image registration, *Lecture Notes in Computer Science*, vol 3216, pp 781–788
94. Wang X, Sabne A, Kisner S, Raghunathan A, Bouman C, Midkiff S (2016) High performance model based image reconstruction. *Association for Computing Machinery*, vol 12, DOI 10.1145/2851141.2851163

-
95. Warfield SK, Jolesz FA, Kikinis R (1998) A high performance computing approach to the registration of medical imaging data. *Parallel Computing* 24:1345–1368, DOI [http://dx.doi.org/10.1016/S0167-8191\(98\)00061-1](http://dx.doi.org/10.1016/S0167-8191(98)00061-1)
 96. Wei Q, Patkar S, Pai DK (2014) Fast ray-tracing of human eye optics on graphics processing units. *Computer Methods and Programs in Biomedicine* 114(3):302–314, DOI <http://dx.doi.org/10.1016/j.cmpb.2014.02.003>
 97. Yeh JY, Fu J (2007) Parallel adaptive simulated annealing for computer-aided measurement in functional MRI analysis. *Expert Systems with Applications* 33(3):706–715, DOI <http://dx.doi.org/10.1016/j.eswa.2006.06.018>
 98. Yip H, Ahmad I, Pong T (1999) An efficient parallel algorithm for computing the gaussian convolution of multi-dimensional image data. *Journal of Supercomputing* 14(3):233–255, DOI 10.1023/A:1008137531862
 99. Zhu YM, Cochoff SM (2010) Medical image viewing on multicore platforms using parallel computing patterns. *IT Professional* 12(2):33–41, DOI 10.1109/MITP.2010.62
 100. Zhuge Y, Cao Y, Miller RW (2009) GPU accelerated fuzzy connected image segmentation by using CUDA. *IEEE Engineering in Medicine and Biology Society* pp 6341–4, DOI 10.1109/IEMBS.2009.5333158
 101. Zhuge Y, Cao Y, Udupa JK, Miller RW (2011) Parallel fuzzy connected image segmentation on GPU. *Medical Physics* 38(7):4365–4371, DOI 10.1118/1.3599725

102. Zhuge Y, Ciesielski KC, Udupa JK, Miller RW (2013) GPU-based relative fuzzy connectedness image segmentation. *Medical Physics* 40(1), DOI 10.1118/1.4769418
103. Zinterhof P (2012) High-throughput-screening of medical image data on heterogeneous clusters. *Lecture Notes in Computer Science* 7116 LNCS:368–377, DOI 10.1007/978-3-642-29843-1_42, cited By 0