A Portable Multi-CPU/Multi-GPU Based Vertebra Localization in Sagittal MR Images

Mohamed Amine Larhmam(✉), Sidi Ahmed Mahmoudi,
Mohammed Benjelloun, Saïd Mahmoudi, and Pierre Manneback

Faculty of Engineering, University of Mons, 20, Place du Parc., Mons, Belgium
{MohamedAmine.Larhmam,Sidi.Mahmoudi,Mohammed.Benjelloun,
Said.Mahmoudi,Pierre.Manneback}@umons.ac.be

Abstract. Accurate Vertebra localization presents an essential step for automating the diagnosis of many spinal disorders. In case of MR images of lumbar spine, this task becomes more challenging due to vertebra complex shape and high variation of soft tissue. In this paper, we propose an efficient framework for spine curve extraction and vertebra localization in T1-weighted MR images. Our method is a fast parametrized algorithm based on three steps: 1. Image enhancing 2. Meanshift clustering [5] 3. Pattern recognition techniques. We propose also an adapted and effective exploitation of new parallel and hybrid platforms, that consist of both central (CPU) and graphic (GPU) processing units, in order to accelerate our vertebra localization method. The latter can exploit both NVIDIA and ATI graphic cards since we propose CUDA and OpenCL implementations of our vertebra localization steps. Our experiments are conducted using 16 MR images of lumbar spine. The related results achieved a vertebra detection rate of 95% with an acceleration ranging from 4 to 173 $\times$ thanks to the exploitation of Multi-CPU/Multi-GPU platforms.

Keywords: Vertebra localization · MR images · Mean-shift · Heterogeneous computing · GPU · CUDA · OpenCL

1 Introduction

Medical image analysis presents a necessary tool for clinical purposes, especially for diagnosing many orthopedic conditions, such as osteoporosis, spinal fractures and spine metastasis. In order to obtain an efficient spine analysis, accurate vertebra localization provides a relevant information for disease recognition and surgical planning. However, automating vertebra detection remains a challenging task due to their complex shape, their various appearances between patients, and density variability in image modalities (MRI, CT, X-ray, etc.). Therefore, we propose a fast parameterized and accurate method of vertebra localization in MR images, based on clustering and features extraction techniques. On the other hand, a key variable that has to be taken into account in medical applications

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part II, LNCS 8815, pp. 209–218, 2014.
DOI: 10.1007/978-3-319-11755-3_24

is the computation time, which can be so elevated in case of processing large volumes of high definition images.

To overcome this constraint, one can imagine to exploit parallel-based architectures such as cluster, grid computing, Graphics Processing Units, etc. The GPUs present an efficient solution, which is seriously hampered by the high costs of data transfer between CPU and GPU memories. Therefore, we developed a version that exploits all the available computing units within computers (CPUs or/and GPUs). This implementation can exploit both NVIDIA and ATI graphic cards, based on CUDA[1] and OpenCL[2] respectively. The remainder of the paper is organized as follows: Section 2 presents the related works, while the third Section is devoted to describe our method of vertebra localization in MR images. In Section 4, we present the portable Multi-CPU/Multi-GPU implementation that we propose in order to reduce the computation time. Experimental results are presented and evaluated in the fifth section. Finally, conclusions and future works are described in the last section.

2 Related Work

In the literature, several segmentation techniques have been investigated on vertebra shape extraction. In this context, Kelm *et al.* [8] proposed a learning based method for automatic detection and labeling of 3D spinal geometry in MRI and CT-scan. Glocker *et al.* [7] presented a new method of localization and identification of vertebrae in arbitrary field of view CT-scan. Ma *et al.* [11] used learning based edge detection and deformable surface model in order to perform a hierarchical segmentation of thoracic vertebrae in 3D CT-scan. For X-ray images, Larhmam *et al.* [9] combined a template matching method with clustering techniques to analyze vertebra alignment. Dealing with MDCT images, Baum *et al.* [3] proposed an algorithm for osteoporotic vertebral fracture detection.

Otherwise, many image processing algorithms contain phases that consist of similar calculations between image pixels. As result, these algorithms fit naturally with parallel implementation on GPU. In this category, Yang *et al.* [19] implemented several classic image processing algorithms on GPU with CUDA. Reichenbach *et al.* [16] combined the advantages of multicore CPUs, GPUs, and FPGAs to build up a heterogeneous image processing pipeline for adaptive optical systems. Authors in [1] developed an OpenCL library of image processing primitives (OpenCLIPP) in order to simplify the exploitation of GPUs. Some GPU works were dedicated to medical imaging in [17] which presents a survey of GPU based medical applications, related to segmentation, registration and visualization methods. There are also some works related to the exploitation of heterogeneous architectures that dispose of both CPUs and GPUs such as OpenCL and StarPU [2]. Recently, we developed CPU [9], parallel [12] and heterogeneous [10] implementations for vertebra detection in X-ray images. Our main contributions can be summarized in two points:

[1] CUDA. www.nvidia.com/cuda
[2] OpenCL. www.khronos.org/opencl

1. Fast vertebra localization algorithm using T1-weighted MR images based on pre-processing, mean-shift clusetring and feature extraction algorithms. This method can localize abnormal vertebrae with irregular shapes since it require no prior models.
2. A portable and hybrid solution for vertebra segmentation which can exploit the full computing power of both ATI and NVIDIA GPUs and also Multi-CPU/Multi-GPU platforms. This solution offers an efficient scheduling and management of memories within heterogeneous platforms in order to improve our applications performance.

3 CPU Implementation

The proposed approach is an automated framework for spine extraction and vertebra localization in sagittal T1-weighted MR images. Indeed, our method is a fast parameterized approach based on image enhancing, mean-shift clustering and pattern recognition algorithms. As output, lumbar spine curve is extracted followed by a localization and segmentation of vertebrae. Our approach is based on the following steps:

3.1 Contrast-Limited Adaptive Histogram Equalization

In order to enhance contrast in the input MR images (Fig.1(a)), we apply a histogram based technique called CLAHE [15]. This filter first computes different local histograms corresponding to each part of the image, and uses them to change the contrast of distinct regions of the image. This method is well known for limiting noise amplification. The result of this step is shown in Fig.1(b)

3.2 Morphological Opening

In order to smooth shapes and remove morphological noises on the enhanced image, we apply a morphology opening operation. This step consists of a dilatation of erosion using an ellipse structuring element. Indeed, the use of ellipse allows to highlight the shape of vertebrae. Fig.1(c) shows the result of this step.

3.3 Mean-Shift Clustering

Mean-shift is a versatile nonparametric clustering method based on density estimation. The mean-shift procedure became widely used in computer vision and image processing after reintroduction in [4] and [5]. The mean-shift algorithm is based on an iterative mode, it initializes a window and data points before computing their center of gravity. Then, it shifts the search window to the mean and repeat until convergence. The mean-shift uses a Kernel Density Estimation to estimate the probability density function of a feature space. The segmentation using The mean-shift algorithm is performed in two stages, the smoothing process followed by the clustering step. The procedure can be summed as follows:

- Choose the spacial and range bandwidths (h_s, h_r) of the search window
- For each pixel x_i of the image,
  1. Compute the mean-shift term $m(x_i)$ and increment $x_{i,j}$ value.
  2. Repeat until convergence $x_{i,c}$.
  3. Assign $z_i = (x_i^s, x_{i,c}^r)$.
- $(z_i)_{i=1,...,n}$ are pixels of the output smoothed image.
- The clusters are constructed by grouping all the (z_i) which are closer by (h_s, h_r) in both spacial and rage domain
- Each pixel x_i of the image takes the same cluster label of its point of convergence z_i. These labels constitute the output segmented image.

where (x^s, x^r) are the spacial and range parts of a given data point.

the *mean shift* term defines the translation vector toward the direction of the maximum increase in the density function [5]. The choice of the bandwidths (h_s, h_r) has a significant influence in mean-shift procedure. In case of spine MR images, the value of the bandwidths that maximizes detection accuracy is determined during experimentation process. The result of this step is shown in Fig.1(d)

3.4 Contour Extraction

In this step, we use the algorithm of Suzuki *et al.* [18] to collect information on contours from the result of mean-shift clustering. This method extracts the different contours of the image and stores them into vectors. This generates a collection of external contours ready to by analyzed.

3.5 Vertebra Localization

Ellipse Fitting. For the high level contours analysis, the binary shape of the vertebra is approximated with a parametric elliptic curve. Fitzgibon *et al.* [6] proposed an ellipse fitting technique which is robust to noise. This algorithm uses a least square fitting to find the best ellipse that describes the extracted contour.

Given $(x_i, y_i)(i = 1, ..., n)$ an n points contour. The objective of the method is to minimize the error between an ellipse $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$ and the contour. The method uses a least square optimization with the algebraic criteria.

$$\begin{cases} \min \sum_{i=0}^{n} (Ax^2 + Bxy + Cy^2 + Dx + Ey + F)^2 \\ B^2 - 4AC = 1 \end{cases} \tag{1}$$

where the constraint $B^2 - 4AC = 1$ ensures that the problem is elliptical.

Indeed, solving the system (1) enables to link each stored contour to an ellipse. Finally, we eliminate ellipses which are out of the range of the accepted size and rotation depending on the size and the position of the vertebrae. As a result, vertebra candidate center points are detected and clustered in an accumulator.

Spine Curve Detection. The objective of this step is to compute the spine curve and eliminate false detection. Thus, we use a least square polynomial fitting as follows:

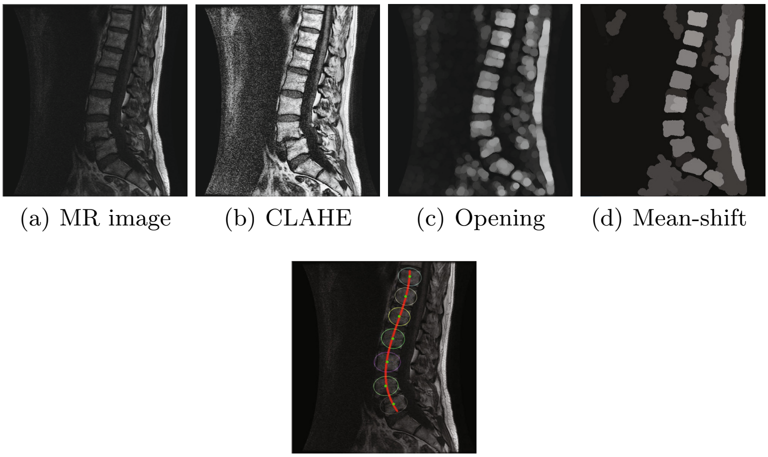
Let $(x_i, y_i)_{i=1...N}$ be the center points of the candidate vertebrae, detected by the ellipse fitting. We aim to fit a polynomial P of degree k with $k < N$ and coefficient $(a_j)_{j=0...k}$. A least square optimization method minimizes the error between the polynomial and the data point. The sum of squared residuals can be expressed (2).

$$S = \sum_{i=1}^{N} (y_i - P(x_i))^2 \tag{2}$$

The sum of squares S is a function of $(a_j)_{j=0...k}$. Then, its minimization is obtained by sitting the gradient to zero $\frac{\partial S}{\partial a_j} = 0, j = 0...k$

The problem can be expressed in matrix form $XA = Y$, where A is a column vector of the polynomial coefficient. Finally, the solution of the least squares problem is $A = X^{-1}Y$

After the extraction of a first curve, we calculate a mean error distance and we eliminate points which present a distance greater to this mean. We repeat this process until the convergence. Fig.1(e) shows the result of the final vertebra spine curve extraction and vertebra localization steps.



(a) MR image (b) CLAHE (c) Opening (d) Mean-shift

(e) Localization

Fig. 1. Framework of vertebra localization in case of a lumbar MR image

4 Hybrid Implementation

Despite the high accuracy of the method presented above, its computing time becomes so significant in case of treating bigger sets of high definition images.

Thus, we developed a portable Multi-CPU/Multi-GPU method that exploits all the available computing units for improving our method's performance. For this aim, we use the executive support StarPU [2] that offers a runtime for launching tasks within heterogeneous platforms. Our method is well adapted for exploiting both NVIDIA and ATI graphic cards since we propose CUDA and OpenCL implementations of each method's step. Indeed, the steps of histogram equalization (CLAHE), morphological opening and MeanShift segmentation are implemented heterogeneously for improving the performance of our vertebra detection method. The proposed implementation consists of applying parallel functions with GPUs [12] and sequential functions with CPUs. These processings are launched simultaneously. Our heterogeneous version can be summarized with three steps: images loading, heterogeneous processing and output images saving.

1. **Images loading :** the first step consists of loading the input images in StarPU buffers.
2. **Hybrid Processing :** after loading the input images, we can apply (within StarPU) simultaneously the CPU and GPU functions on the high intensive steps. The CPU functions are described in Section 3, while the GPU functions are presented with two versions : the first one presents CUDA implementations of histogram equalization (CLAHE), morphological opening and MeanShift segmentation steps in order to exploit NVIDIA graphic cards. These implementations are developed with the CUDA module of OpenCV [3]. The second version presents OpenCL implementations of the same steps (histogram equalization (CLAHE), morphological opening and MeanShift segmentation) in order to exploit ATI graphic cards. The latter are developed with OpenCL module of OpenCV [4]. StarPU consists of two principal modules. The first one is the codelet that allows to specify the type of computing units (CPUs or/and GPU) and the related implementations (C/C++, CUDA or OpenCL). The second module is represented by tasks that apply the codelet on images. In our case, each image is processed within one task on CPU or GPU. Notice that the selection of the type of GPU (NVIDIA or ATI) is not required since our method selects automatically the available GPU with the corresponding implementations (CUDA or OpenCL).
3. **Results presentation :** the last step consists of copying the output images on buffers using StarPU, which provides a function for automatic data transfer from GPU to CPU memory.
   For a better exploitation of heterogeneous platforms, we employed an effective scheduling of tasks, and also CUDA streaming technique in order to overlap data transfers by kernels executions on GPU. The latter is used within multiple GPUs. More detail about the heterogeneous treatment are described in [10]. Fig. 2 summarizes our heterogeneous implementation. The steps of contours extraction and vertebra localization (Section 3.4 and 3.5) exploit multiple CPUs only since they are applied on images with reduced

[3] OpenCV CUDA. www.opencv.org/platforms/cuda.html
[4] OpenCV OpenCL. http://docs.opencv.org/modules/ocl/doc/introduction.html

informations (after applying the CLAHE, Opening and Meanshift steps). Moreover,these steps present a high computational dependecy which is not adapted for parallel calculation. These steps are grouped in one phase called "Vertebra localization" as shown in Table 2. Notice that the use of StarPU allowed to reduce our development time since it allows for automatic data transfer between CPU and GPU memories. Moreover, StarPU offers an automatic selection of computing units (CPUs or/and GPUs) within one line of code. In our case, the use of StarPU required no more than 25 lines of code compared to the CUDA implementation.

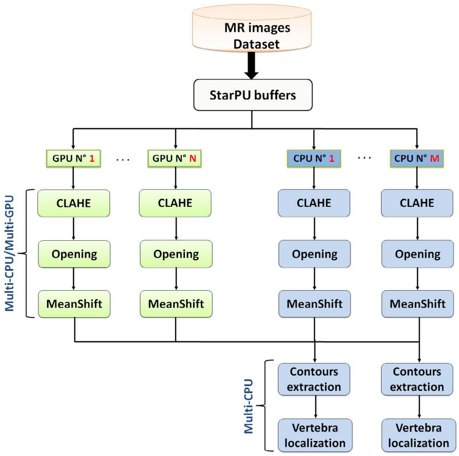


Fig. 2. Heterogeneous vertebra detection in MRI images

5 Experimental Results

Experimentations were conducted using 16 MRI data sets. The images were obtained from an online database proposed by SpineWeb[5]. All the images are T1-weighted MR data, and focus on the lumbar spine area. The data set contains some abnormal cases including degenerative diseases such as osteophyte and compression fractures. We extract the middle slice from each MR data set in order to get the maximum visible vertebrae. The image data set was inspected by an experienced radiologist, from our collaborative hospital, who annotated four landmarks on each visible vertebra. these landmarks are used to generate ground truth data for the detection validation step. Thus, a total of 106 vertebrae were annotated including the lumbar spine.

We set constant parameters for the automated process. Thereby, the band width (hs, hr) of the mean-shift algorithm, the size of the structuring element of

[5] http://spineweb.digitalimaginggroup.ca/

the morphological opening and the range of the accepted ellipses are experimental values determined according to the image data set and the size of vertebrae.

The validation of vertebra detection was performed by using our ground truth data. We focused on the detection of the five lumbar and the visible thoracic vertebrae. Therefore, candidate vertebra centers had to be located within the vertebra body. We compared the detection accuracy with the manual landmarks annotated by the expert radiologist. Indeed, we used the four landmarks that determined the corners to calculate a precise vertebra center. The method enabled a global vertebra detection rate of 95% on a total of 106 vertebrae and a RMS (root mean square) of 3.8 pixels. Intermediate results are shown in Fig 1.

Table 1 compares the detection rate and accuracy of the different lumbar (L1 to L5) and thoracic (Th12 to Th10) vertebrae. The final result showed a vertebra localization rate of 95%. Peng et *al.* [14] announced a mean detection rate of 95.5% of vertebra in MRI. Authors in [13] reported an average intervertebra disc localization rate of 95.4% based on MR images of the lumbar spine. Moreover, our automated method enabled a hight success rate and it is comparable to the state of the art. However, some vertebrae present a detection rate and accuracy lower than the mean. This is due to the presence of abnormal cases with degenerative diseases in the used data set. Therefore, some abnormal vertebrae showed irregular gray level which are difficult to segment. Also, the final accuracy of the detection can be decreased by the morphological opening that changes slightly the original shape of vertebrae.

Table 1. Detection rate and accuracy of vertebrae using 16 MR images

Vertebra type	**Detection rate (%)**	**RMS distance error (px)**
Th12-Th10	96	2.92
L1	100	2.74
L2	93	1.91
L3	100	4.43
L4	93	4.57
L5	87	4.68
Global mean	**95%**	**3.80**

The exploitation of heterogeneous architectures allowed to improve the performance of our method of vertebra localization in MR images. Table 2 compares the computing time between sequential (CPU), parallel (GPU) and hybrid version of the proposed method (Fig. 2). Notice that our performance scale up very well when exploiting multiple CPUs or/and GPUs. The experimentations have been applied on 200 MR images of 512 $\times$512 resolution. We note also that the NVIDIA GPUs offer better accelerations than the ATI ones. Thanks to CUDA which present actually the most performant GPU programming language. However, the use of OpenCL benefits from its compatibility with any type of GPU (NVIDIA or ATI). Notice that the tests were run on the following hardware:

Table 2. Vertebra localization using hybrid platforms (200 MR images of 512×512)

Steps	**1CPU**	**1GPU**		**4GPU/8CPU**			
				OpenCL (ATI)		**CUDA (NVIDIA)**	
	Time (T (s))	**T (s)**	**Acc (x)**	**T (s)**	**Acc (x)**	**T (s)**	**Acc (x)**
CLAHE	13.88 s	1.48 s	09.37 x	0.52 s	26.70 x	0.41 s	33.86 x
Opening	6.84 s	1.20 s	05.70 x	0.31 s	22.06 x	0.24 s	28.5 x
MeanShift	724 s	19.63 s	36.88 x	4.88 s	148.36 x	4.17 s	173.62 x
Vertebra localization	1.98 s	CPU only (1.98 s)		8CPU. T: 0.47 s Acc: 4.21 x			
Total Time	746.7 s	24.29 s	30.74 x	9.92 s	75.27 x	9.03 s	82.69 x

– CPU: 4 x Intel Core 2 Quad Q8200, 2.33GHz,
– GPU: 4 x NVIDIA GeForce GTX 580 with 1.5GB of RAM,

6 Conclusion and Future Works

In this paper, we presented an efficient method for spine extraction and vertebra localization in T1-weighted MR images. Our contribution is based on a preprocessing step, mean-shift clustering and pattern recognition techniques. This method is a fast parameterized processing with no prior learning. This work is a first step for a Computer Aided Diagnosis (CAD) system for spine abnormalities.

We proposed also a portable and heterogeneous implementation, which can exploit multiple CPUs and GPUs (from NVIDIA or ATI), of our method in order to improve its performance. This enabled to apply, in a fast way, the method on larger sets of MR images. Several techniques of optimizations were employed to achieve a full exploitation of the computing power of machines. As future work we plan to include a machine learning step in order to enhance vertebra detection and build a module for vertebra metastasis detection. We plan also to apply a complexity estimation of each step of our vertebra segmentation method in order to have a better distribution of tasks between the available computing units (CPUs or/and GPUs).

Acknowledgments. The authors would like to thank Dr. S. Drisis, radiologist from Jules Bordet Hospital, for the annotation process. The authors thank also the anonymous reviewers for their insightful comments.

References

1. Akhloufi, M., Campagna, A.: OpenCLIPP: OpenCL Integrated Performance Primitives library for computer vision applications. In: Proc. SPIE Electronic Imaging 2014, Intelligent Robots and Computer Vision XXXI: Algorithms and Techniques, pp. 25–31 (2014)

2. Augonnet, C., et al.: StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. Concurrency and Computation: Practice and Experience **23**(2), 187–198 (2011)
3. Baum, T., et al.: Automatic detection of osteoporotic vertebral fractures in routine thoracic and abdominal MDCT. European Radiology, 1–9 (2014)
4. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 790–799 (1995)
5. Comaniciu, D., et al.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 603–619 (2002)
6. Fitzgibbon, A., Fisher, R.B.: A buyer's guide to conic fitting. In: British Machine Vision Conference, pp. 513–522 (1995)
7. Glocker, B., Feulner, J., Criminisi, A., Haynor, D.R., Konukoglu, E.: Automatic localization and identification of vertebrae in arbitrary field-of-view CT scans. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part III. LNCS, vol. 7512, pp. 590–598. Springer, Heidelberg (2012)
8. Kelm, B.M., et al.: Spine detection in CT and MR using iterated marginal space learning. Medical Image Analysis **17**, 1283–1292 (2012)
9. Larhmam, M.A., et al.: Vertebra identification using template matching modelmp and k-means clustering. International Journal of Computer Assisted Radiology and Surgery, 1–11 (2013)
10. Lecron, F., et al.: Heterogeneous Computing for Vertebra Detection and Segmentation in X-Ray Images. Journal of Biomedical Imaging, 1–12 (2011)
11. Ma, J., Lu, L., Zhan, Y., Zhou, X., Salganicoff, M., Krishnan, A.: Hierarchical segmentation and identification of thoracic vertebra using learning-based edge detection and coarse-to-fine deformable model. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part I. LNCS, vol. 6361, pp. 19–27. Springer, Heidelberg (2010)
12. Mahmoudi, S.A., et al.: GPU-Based Segmentation of Cervical Vertebra in X-Ray Images. In: International Conference on Cluster Computing, pp. 1–8 (2010)
13. Oktay, A.B., Akgul, Y.S.: Localization of the lumbar discs using machine learning and exact probabilistic inference. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011, Part III. LNCS, vol. 6893, pp. 158–165. Springer, Heidelberg (2011)
14. Peng, et al.: Automated vertebra detection and segmentation from the whole spine MR images. In: Proceedings of Medical Imaging Computing and Computer Assisted Intervention (2007)
15. Pizer, S.M., et al.: Adaptive histogram equalization and its variations. Comput. Vision Graph. Image Process. **39**(3), 355–368 (1987)
16. Reichenbach, M., Seidler, R., Fey, D.: Heterogeneous computer architectures: An image processing pipeline for optical metrology. In: International Conference on Reconfigurable Computing, pp. 1–8 (2012)
17. Shi, L., et al.: A survey of GPU-based medical image computing techniques. Quant. Imaging Med. Surg. **2**(3), 188–206 (2012)
18. Suzuki, et al.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing **30** (1985)
19. Yang, Z., et al.: Parallel Image Processing Based on CUDA. In: International Conference on Computer Science and Software Engineering, China, pp. 198–201 (2008)