

AN OPTIMISED MULTI-BASELINE APPROACH FOR ON-LINE MR-TEMPERATURE MONITORING ON COMMODITY GRAPHICS HARDWARE

B. Denis de Senneville^{1,2,3}, K. O. Noe⁴, M. Ries¹, M. Pedersen³, C. T. W. Moonen¹, T. S. Sorensen⁴

¹ IMF, UMR 5231 CNRS/Université Bordeaux 2, France, ² IUT Bordeaux 1, France,

³ MR Research Centre Institute of Clinical Medicine, University of Aarhus, Denmark

⁴ Department of Computer Science, University of Aarhus, Denmark

{baudouin,ries,moonen}@imf.u-bordeaux2.fr, {sangild,kn}@daimi.au.dk, michael@mr.au.dk

ABSTRACT

Magnetic Resonance Imaging (MRI) can be used for non invasive temperature mapping and is therefore a promising tool to monitor and control interventional therapies based on thermal ablation. The Proton Resonance Frequency shift MRI technique gives an estimate of the temperature by comparing phase changes between dynamically acquired images. These temperature measurements are prone to motion induced errors however, particularly in abdominal organs due to breathing.

Several computational approaches have been proposed previously to correct for these motion related errors on the measured temperature. They have required significant time to compute however, and have not been sufficiently fast for several real-time temperature mapping applications. This paper proposes to use modern graphics cards (GPUs) to assess on-line motion corrected thermal maps. The computation times obtained on the GPU are compared to an existing CPU reference implementation. An acceleration factor close to 7 was obtained for the processing of one slice (resolution 128×128 pixels), and higher than 21 for 12 slices, allowing a real-time implementation.

Index Terms— Magnetic Resonance Imaging, Real time systems, Temperature control, Image motion analysis, Motion compensation.

1. INTRODUCTION

Real-time Magnetic Resonance (MR) thermometry provides continuous temperature mapping inside the human body and is therefore a promising tool to monitor and control interventional therapies based on thermal ablation [1]. Thermotherapy procedures can be performed using High Intensity Focused Ultrasound (HIFU) devices which are non-invasive since the device itself is fully extra-corporal. The observable MR signal is a complex number $Me^{i\varphi}$. Grey levels on anatomical images are proportional to the magnitude value whereas phase value relates to the proton resonance frequency. The most widely used MR temperature mapping is based on temperature dependence of the water Proton Resonance Frequency (PRF) [2]. The temperature map at instant n (noted ΔT_n) can be obtained on-line by analyzing signal variation between the current phase image φ_n and a reference phase image φ_{ref} acquired before the hyperthermia (typically the first of the temporal series) as follows:

$$\Delta T_n = (\varphi_{ref} - \varphi_n) \cdot k \quad k = \frac{1}{\gamma \cdot \alpha \cdot B_0 \cdot TE} \quad (1)$$

where γ is the gyromagnetic ratio (≈ 42.58 MHz/Tesla), α the temperature coefficient (≈ 0.009 ppm/K), TE the echo time and B_0 the main magnetic field. This calculation is performed for each voxel to obtain temperature maps.

MR-temperature measurements allow on-line thermal dose evaluation during an intervention, which in turn permits immediate prediction of tissue necrosis, and hereby a prediction of the effectiveness of the therapeutic thermal treatment. Lethal effects of elevated temperatures have been studied by Sapareto et al. [3] who established an empirical relation between past temperature, duration of exposure, and cell death.

On-line temperature monitoring may also improve treatment efficiency by adapting local energy deposition to deliver a pre-defined thermal dose in a pre-defined volume; to obtain such a control, it has been shown that MR thermometry can be used to provide spatial and temporal temperature feedback for the power control of the heating device. Until now, such a control system has been demonstrated for immobilized tissues in vitro and in vivo.

Thermotherapy opens great prospects for treatment of vital organs such as the kidney, the liver, and the heart. These organs move however, and, since B_0 is generally spatially non-uniform, any phase measurements on a tissue sample taken at a different position will show a relative phase difference. Therefore, the attempt to detect temperature changes with equation (1) would be severely biased by motion induced phase changes [4]. A robust removal of these non-temperature related phase variations, i.e. motion correction, is a prerequisite for precise MR-thermometry on moving targets. Figure 1 depicts a multi-baseline approach which address this problem [5]. This technique is motivated by the fact that for most therapeutic applications within the human body, motion is caused by the respiratory cycle and is thus periodic. This can be exploited as follows:

- **Step 0:** a magnitude and phase lookup table, which covers the entire motion cycle, is established prior to MR-thermometry.
- **Step 1:** during the intervention, the phase image of the lookup table acquired with a similar organ position is selected, and used as a reference for temperature computation with equation (1). The correct temperature can now be estimated since the phase differences represent only temperature related phase changes..
- **Step 2:** temperature information is mapped to a reference position in order to allow computation based on past temperature measurements (such as on-line thermal dose evaluation and automatic power control of the heating device). In addition, when the heating is performed with a HIFU device, estimated organ displacement also allows dynamic adjustment

of the focal point position to track the targeted pathological tissue. Without such corrections, the treatment is inefficient or, worse, may induce unwanted destruction of healthy tissue.

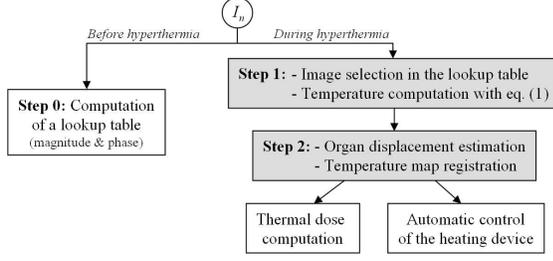


Fig. 1. Data processing sequence for temperature maps computation. In this study, we propose to accelerate processing steps reported in grey using a GPU hardware.

Two conditions must be met in order to make MR temperature mapping widely useful for thermal therapies in clinical practice: 1) adequate spatial and temperature resolution; 2) on-line availability of accurate temperature maps and thermal dose maps. Processing of an image must thus be done fast enough to ensure real-time monitoring of temperature evolution. In practical terms this implies that processing must be achieved within the delay between successive MR-acquisitions.

Recent techniques based on the use of additional information [6] [7] have been proposed to address step 1 and/or step 2. MR magnets now allow on-line acquisition of large data volumes. This allows assessment of a more robust and complex description of organ displacements. The feasibility of non-invasive thermo-ablation procedures using this information was demonstrated on mobile ex-vivo targets [5]. However, the numerical computation required for on-line treatment of acquired data sets limits the use of fast imaging techniques. In addition, with HIFU devices, a prediction of the target position is required for dynamic adjustment of the focal point location as the delay between the acquisition and the time when motion information has actually been computed is not negligible. This has previously constrained the use of those techniques to patients under artificial respiratory monitoring. Consequently, to generally assess thermal ablation in-vivo, an acceleration of those techniques has to be achieved. The idea proposed in this paper is to use graphics hardware (GPUs) as a fast parallel computational platform to significantly speedup the required computations to enable on-line estimation of target positions and online correction for motion related errors on thermal maps.

In practice, as on-line acquisition of 3D isotropic images is difficult because of technical limitations and Signal/Noise ratio considerations, individual 2D processing of each acquired slice is performed.

2. GPU IMPLEMENTATION

A GPU can be seen as a massively parallel coprocessor (the most recent GPUs have 16 SIMD multiprocessors containing 8 processors each) and is thus very well suited for computational problems that can be solved in parallel. Algorithms have to be formulated such that computation is distributed to a high number of independent threads. In the CUDA (Compute Unified Device Architecture) GPU programming framework [8] threads are organized in so-called thread blocks. Each thread block can currently contain up to 512 threads which are executed by a single multiprocessor using time slicing. This means that the computations of the individual threads

in each thread block are interleaved, primarily to hide memory latency. The GPU has a large amount of general DRAM global memory, which provides relatively slow data access compared to the on-chip shared memory that is also available on a per block level. This shared memory features very fast general read and write access that threads in the same block can use to share data. To optimize memory bandwidth, memory access from the individual threads can be aligned such that a memory accesses can be coalesced into a single contiguous memory access by the memory controller.

In this paper, a GPU-accelerated multi-baseline approach for correction of motion related errors on temperature maps is described and evaluated. The algorithm was implemented using CUDA v1.0 and C++. It was tested on an Intel Core 2 2.13 GHz with 2 Gb of RAM and a NVIDIA 8800 GTX graphic card with 768 Mb of RAM.

3. METHOD DESCRIPTION

3.1. Step 0 & Step 1 : On-line correction of motion related errors on temperature maps

Classic multi-baseline correction approaches use a complete collection of reference magnitude and phase images constructed before initiation of the thermal therapy. For this purpose, K images are acquired with the same MR protocol without hyperthermia (K is chosen to cover the entire motion cycle). Subsequently, during MR-thermometry, for a given organ position, the corresponding phase correction is selected and subtracted from the current phase. A robust selection criterion consists of evaluating the maximum inter-correlation coefficient between the actual magnitude image I_{cur} and each magnitude images stored the collection $I_{col}^k, \forall k \in [0, K[$:

$$\max \left(\frac{\sum_{x,y} \left((I_{cur}(x,y) - \bar{I}_{cur}) (I_{col}^k(x,y) - \bar{I}_{col}^k) \right)}{\sqrt{\sum_{x,y} (I_{cur}(x,y) - \bar{I}_{cur})^2 \cdot \sum_{x,y} (I_{col}^k(x,y) - \bar{I}_{col}^k)^2}} \right) \quad (2)$$

where \bar{I}_{cur} and \bar{I}_{col}^k are average pixel intensities of I_{cur} and I_{col}^k respectively.

3.1.1. Step 0: before hyperthermia

The following numerical expressions are computed each time a new image is acquired before hyperthermia:

1. $\gamma_k(x,y) = \left(I_{col}^k(x,y) - \bar{I}_{col}^k \right), \forall (x,y) \text{ and } \forall k \in [0, K[,$
2. $\sum_{x,y} \left(I_{col}^k(x,y) - \bar{I}_{col}^k \right)^2 = \sum_{x,y} \gamma_k(x,y)^2, \forall k \in [0, K[,$

This information is stored with the magnitude image information I_{col}^k in the global GPU device memory and is thus directly available during the hyperthermia procedure, avoiding costly data transfers from the CPU to the GPU.

3.1.2. Step 1: during hyperthermia

For each newly acquired image the following numerical expressions are computed:

1. $\sum_{x,y} \left((I_{cur}(x,y) - \bar{I}_{cur}) \cdot \gamma_k(x,y) \right), \forall k \in [0, K[,$
2. $\sum_{x,y} (I_{cur}(x,y) - \bar{I}_{cur})^2, \forall k \in [0, K[,$

using parallel kernels to assess the average pixel intensity \bar{I}_{cur} , as well as coefficients inside each sum, and sums computation. 1D thread blocks were used so that memory accesses were coalesced. An optimal device occupancy was found for a block size of 256 threads on the tested GPU hardware.

The phase image corresponding to the maximum inter-correlation (see equation (2)) is then selected and taken as reference for temperature computation with equation (1).

3.2. Step 2 : On-line organ displacement estimation and correction

The objective is to register each image voxel in the most recently acquired image (noted I_{cur}) with the corresponding reference image (noted I_{ref}). I_{ref} is chosen to be the first image in the temporal series in step 0.

3.2.1. Algorithm for organ displacement estimation

Motion can be computed with differential estimation methods of optical flow that estimate a velocity field assuming an intensity conservation during displacement (mathematically expressed by the left part of equation (3)). A regularity constraint is also required. The global regularity constraint proposed by Horn and Schunck [9] provides a good estimation of the displacement of the organs because it matches real organ motion assuming that motion field vectors have similar values for adjacent pixels (right part of equation (3)). We seek a transformation minimizing:

$$\int_y \int_x ([I_x u + I_y v + I_t]^2 + \alpha^2 [\|\nabla u\|_2^2 + \|\nabla v\|_2^2]) dx dy \quad (3)$$

where u and v are displacement vector components, I_x , I_y , I_t are the spatio-temporal partial derivatives of the intensity, and α is a user defined weighting factor (a typical value of 0.3 was used in this study).

3.2.2. Implementation on the GPU hardware

The registration method is outlined as a sequence diagram in figure 2. All the time consuming steps can be processed in parallel:

- Downsampling of an image with a factor 2.
- Upsampling of an image with a factor 2.
- Application of a spatial transformation on an image.
- Computation of spatio-temporal gradients of the intensity I_x , I_y and I_t .
- Resolution of the numerical scheme defined by equation (3).

A computation kernel is programmed for each processing step. Computations are performed on thread blocks of size $M \times N$ - each thread corresponding to one pixel. M was chosen higher than N to ensure coalesced memory accesses. Optimal device occupancy was found for $(M, N) = (32, 4)$ on the tested GPU hardware. Numerical minimization of equation (3) is solved with an iterative scheme based on the Jacobi method as follows:

$$\begin{cases} u^{k+1} = \bar{u}^k - \frac{I_x \cdot \bar{u}^k + I_y \cdot \bar{v}^k}{\alpha^2 + I_x^2 + I_y^2} \\ v^{k+1} = \bar{v}^k - \frac{I_x \cdot \bar{u}^k + I_y \cdot \bar{v}^k}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \quad (4)$$

where k denotes the iteration number (a typical number of 100 iterations was performed), \bar{u}^k and \bar{v}^k refers respectively to the average of u^k and v^k in a neighborhood 3×3 of the current pixel position.

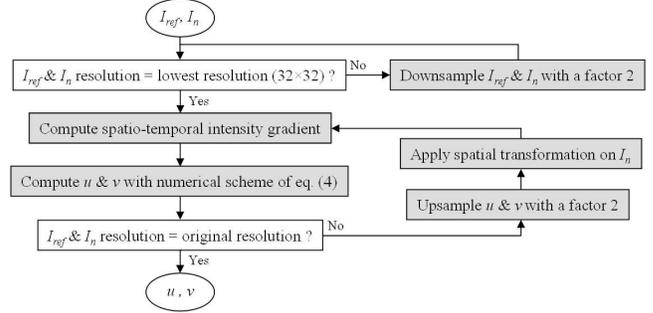


Fig. 2. Data processing sequence for motion estimation using a multi-resolution approach of the Horn&Schunck algorithm. Processing steps in grey are accelerated using the GPU hardware.

Blocks are transferred into shared memory for fast data access. The image is thus only read from and written to global memory at the beginning and at the end of each iteration respectively. For each pixel the computation of equation (4) requires intensity information in a neighborhood of size 3×3 pixels. Each block thus requires intensity information in a shared memory region of size $(M + 2) \times (N + 2)$ pixels. Each thread performs the numerical computations of equation (4) for one pixel of the acquired image.

Finally the estimated motion field is used to map temperature information to the reference position.

In case of multi-slice acquisitions, each kernel computes all slices within one invocation.

4. EXPERIMENTAL VALIDATION

4.1. Results obtained on a single slice

Computation times were measured with CPU and GPU-optimized implementations and reported in Table 1. An overall acceleration factor of 6.9 was obtained for one image of resolution 128×128 (composed of acceleration factors of 2.3 and 7.6 for step 1 and step 2 respectively), 14.3 for a resolution 256×256 (composed of acceleration factors of 3.8 and 16.4 for step 1 and step 2 respectively), 22 for a resolution 512×512 pixels (composed of acceleration factors of 4.5 and 26 for step 1 and step 2 respectively).

Processing step	128 × 128		256 × 256		512 × 512	
	CPU	GPU	CPU	GPU	CPU	GPU
Step 1	7.2	3.1	28.7	7.6	115.3	25.4
Step 2	153.8	20.3	636.5	38.9	2616.9	99
Total	161	23.4	665.2	46.5	2732.2	124.4

Table 1. Comparison between computation time (in milliseconds) required with CPU and GPU implementations, for each processing step, with different image resolutions (results for step 1 were obtained using a collection of 100 images).

4.2. Results obtained on multi-slice acquisitions

Figure 3 reports computation times measured with the GPU-optimized implementation on multi-slice acquisitions using an in-plane resolution of 128×128 pixels. A fixed CPU overhead of 1.4 ms is required for step 1, and of 16 ms for step 2 (see dashed lines). Then, for each slice, an average additional GPU contribution of 1.7 ms is required for step 1 and of 4.3 ms for step 2. Thus, although the individual processing of 12 slices in multi-baseline

collections of 100 images required $12 \times 7.2 = 86.4$ ms on the CPU, only 21.9 ms were required on the GPU (which reflects an acceleration factor close to 4). Similarly, although the 2D registration of 12 slices required $12 \times 153.8 = 1845.6$ ms on the CPU, only 67.4 ms were required on the GPU (which reflects an acceleration factor higher than 27). An overall acceleration factor higher than 21 was thus obtained in case of an acquisition of 12 slices.

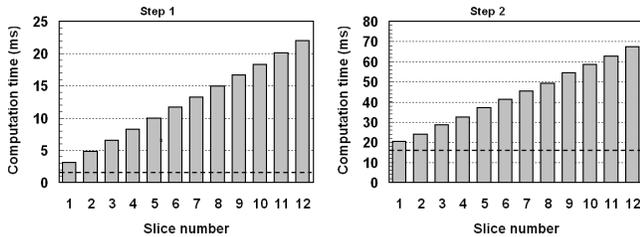


Fig. 3. Computation time (in milliseconds) obtained for different slice numbers of resolution 128×128 pixels. **Left:** step 1 (using a collection of 100 images), **Right:** step 2.

Figure 4 shows an example of the results obtained for abdominal organ displacement on a healthy volunteer under free breathing. All images were obtained on a 1.5 Tesla Philips Achieva system with a conventional gradient echo sequence ($TE=18$ ms). Each slice was acquired at a resolution of 128×128 with a voxel size of $2.5 \times 2.5 \times 8$ mm³ (see Fig. 4.a). 100 images were acquired during the pre-treatment step and stored in the multi-baseline collection to allow a precise sampling of the respiratory cycle. As expected, the displacement vectors direction is mainly vertical with a higher value for the liver as compared to the kidneys (see Fig. 4.b). Fig. 4.c and 4.d compare standard deviation on each pixel for a temporal series before and after correction respectively. With the proposed approach, more than 80% of pixels in the kidney depicts a temperature standard deviation lower than 3°C.

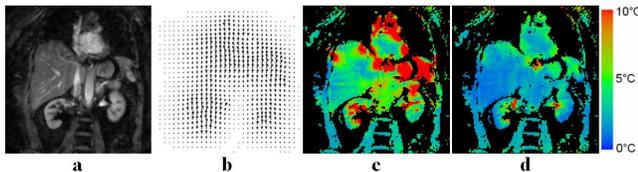


Fig. 4. MR thermometry on the abdomen of a free breathing volunteer. **(a)** anatomical image, **(b)** estimated displacement field vector, **(c, d)** temporal temperature standard deviation reported for each pixel without **(c)** and with **(d)** the proposed correction.

5. DISCUSSION AND CONCLUSION

This study shows that our GPU implementation achieves a significant speedup compared to our existing CPU implementation for correction of motion related errors on temperature maps. The total computation times on the GPU were clearly below the typical MR acquisition duration for all tests, demonstrating that on-line monitoring of temperature evolution is feasible under our experimental conditions. It is also interesting to note that computation is performed in an asynchronous way: the CPU is able to manage other processes (for example user interaction or piloting of the heating device) while the GPU makes computations. The obtained results open great opportunities to perform real-time temperature monitoring using recent fast MR thermometry sequences.

It is also interesting to note that, in step 1, simple computations are performed on a large amount of data, while complex numerical

calculations are performed on a small data set in step 2. Memory access latency explains that a more significant acceleration was obtained for step 2. Nevertheless, the proposed study demonstrates that the GPU accelerates the computation of both of these two “opposite” tasks.

The implemented registration algorithms rely on the assumption of conservation of magnitude pixel value along motion. However, this condition can be violated during thermotherapy as several MR relevant tissue properties such as T_1 and T_2 relaxation times can change during heating, leading to local signal intensity variations in the heated region. The global regularity constraint proposed by Horn and Schunck is more robust to this effect compared to algorithms estimating displacement on image blocks (which leads to significant errors on the estimated motion field). The implemented algorithm is well adapted for small temperature increases, as for example when heating is performed with a focused ultrasound device. In case of large temperature increases (as for example when heating is performed using laser or radio-frequency devices) adapted registration algorithms have to be used [10]. The results obtained in the present study show great prospects for implementation of such techniques on GPU hardware.

The proposed method combined with a fast MR acquisition and reconstruction protocol [11] offers good expectations for accurate real time characterization of complex 3D organ displacement.

In view of future clinical application, quantitative rapid MR temperature imaging may provide an effective real-time monitoring of the intervention and a clinical endpoint for the therapeutic treatment.

6. REFERENCES

- [1] Dodd G. D., et al., Minimally invasive treatment of malignant hepatic tumors: at the threshold of a major breakthrough, *Radiographics* 20(1):9-27, 2000.
- [2] Quesson B., et al., Magnetic resonance temperature imaging for guidance of thermotherapy, *JMRI*, 2000, 12:525-533.
- [3] Sapareto S. A., Dewey W. CL, Thermal dose determination in cancer therapy. *Int. J. Rad. Onc. Biol. Phys.* 10, 787-800. 1984.
- [4] De Poorter J., Noninvasive MRI thermometry with the proton resonance frequency method: study of susceptibility effects, *Magnetic Resonance in Medicine* 1995;34(3):359-67.
- [5] Denis de Senneville B., Mougenot C., Moonen C. T. W., Real time adaptive methods for treatment of mobile organs by MRI controlled High Intensity Focused Ultrasound, *Magnetic Resonance in Medicine*, 2007. 57(2):319-330.
- [6] Suprijanto S., et al., Displacement Correction Scheme for MR-Guided Interstitial Laser Therapy, *Lecture Notes in Computer Science* 2879 2003;2.
- [7] de Zwart J. A., et al., On-line correction and visualization of motion during MRI-controlled hyperthermia, *Magnetic Resonance in Medicine*, 2001;45(1):128-37.
- [8] <http://developer.nvidia.com/object/cuda.html>
- [9] Schunck B. G., Horn K. P., Determining optical flow, *Artificial intelligence*, 1981, 17:pp. 185-203.
- [10] Maclair G., et al., PCA-based image registration : application to on-line MR temperature monitoring of moving tissues, *IEEE, ICIP 2007*, vol.III, 141-144.
- [11] Hansen M. S., Atkinson D., Sorensen T. S., Cartesian SENSE and k-t SENSE Reconstruction using Commodity Graphics Hardware, *Magnetic Resonance in Medicine*, In press.