Microprocessors and Microsystems 39 (2015) 998–1011

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

# Correlation ratio based volume image registration on GPUs

Ang Li [a,b,*], Akash Kumar [b], Yajun Ha [c], Henk Corporaal [a]

[a] Eindhoven University of Technology, Eindhoven, The Netherlands
[b] National University of Singapore, Singapore
[c] Institute for Infocomm Research, Singapore

ARTICLE INFO

*Article history:*
Available online 12 May 2015

*Keywords:*
Image registration
Histogram
GPU
Correlation ratio
Conflict-free

ABSTRACT

Volume image registration remains one of the best candidates for Graphics Processing Unit (GPU) acceleration because of its enormous computation time and plentiful data-level parallelism. However, an efficient GPU implementation for image registration is still challenging due to the heavy utilization of expensive atomic operations for similarity calculations. In this paper, we first propose five GPU-friendly Correlation Ratio (CR) based methods to accelerate the process of image registration. Compared to widely used Mutual Information (MI) based methods, the CR-based approaches require less resource for shadow histograms, a faster storage, such as the on-chip scratchpad memory, therefore can be fully exploited to achieve better performance. Second, we make design space exploration of the CR-based methods, and study the trade-off of introducing shadow histograms on different storage (shared memory, global memory) by computation units of different granularity (thread, warp, thread block). Third, we exhaustively test the proposed designs on GPUs of different generations (Fermi, Kepler and Maxwell) so that performance variations due to hardware migration are addressed. Finally, we evaluate the performance impact corresponding to the tuning of concurrency, algorithm settings as well as overheads incurred by preprocessing, smoothing and workload unbalancing. We highlight our last CR approach which completely avoids updating conflicts of histogram calculation, leading to substantial performance improvements (up to 55× speedup over naive CPU implementation). It reduces the registration time from 145 s to 2.6 s for two typical $256 \times 256 \times 160$ volume images on a Kepler GPU.


## 1. Introduction

Volume image registration (VIR), the process of generating a transformation that maximizes the similarity between two volume images [1] (see Fig. 1), is one of the fundamental components frequently encountered in many medical image processing applications [2]. Among various medical registration frameworks, FMRIB's Linear Image Registration Tool (FLIRT) [3,4] is reported to be effective and robust [5]. Several similarity functions are exploited in FLIRT, the default one, however, is Correlation Ratio (CR) [6]. Based on information theory, CR exhibits comparative robustness and stability as the Mutual Information (MI) methods [4,7]. It is also reported that CR is more accurate and easier to compute than MI [4], which is confirmed by this paper as well.

VIR traditionally requires enormous computation time (e.g. registering two $256 \times 256 \times 160$ images spends 145 s). The calculation of the similarity function, however, is the most dominant component which takes over 98% of the registration time. Meanwhile, the similarity function is inherently data parallel [8] as voxels of the volume images can be processed independently. Therefore, ever since Nvidia published Compute Unified Device Architecture (CUDA) [9], people are seeking to accelerate VIR as well as the similarity function calculations via GPU. However, an efficient GPU implementation for VIR is still challenging due to heavy utilization of expensive atomic operations for similarity calculations, which frequently turn into a performance bottleneck [10]. Although several approaches are proposed [10–14], most of them are specifically targeted for MI and still fail to resolve the bottleneck very effectively.

In this paper, we show that, compared to MI, the CR-based similarity functions are more suitable for the GPU platform. We thus explore the design space of CR and propose five CR-based similarity function implementations. The FLIRT registration framework is implemented to embed these similarity functions to construct a complete registration procedure. We show the trade-off between benefits and overheads of mapping local sub-histograms (or shadow histograms) to different storage (shared memory, global

* Corresponding author is currently at: Eindhoven University of Technology, Eindhoven, The Netherlands.

*E-mail addresses:* ang.li@tue.nl (A. Li), akash@nus.edu.sg (A. Kumar), ha-y@i2r.a-star.edu.sg (Y. Ha), h.corporaal@tue.nl (H. Corporaal).

http://dx.doi.org/10.1016/j.micpro.2015.04.002
0141-9331/

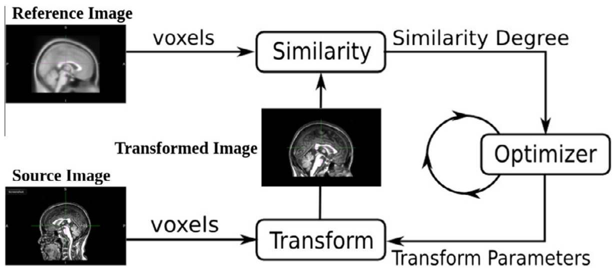


**Fig. 1.** Image registration. In the example, the source image is a raw MRI image while the reference image is a template. The registration framework measures the similarity between the transformed image and the reference image and tunes the transform matrices accordingly based on the searching strategies. After registration, the raw image is supposed to be aligned with the template when applying the obtained transform.

memory) by execution units of different granularity (thread, warp, thread block). The proposed designs are exhaustively tested on GPUs of different generations (Fermi, Kepler and Maxwell) so that performance variations due to hardware migrations are addressed. Further, the performance impact corresponding to the tuning of concurrency, algorithm settings (such as the number of bins) as well as overheads induced by preprocessing, smoothing and workload unbalancing are also evaluated. It is highlighted that, in the last proposed scheme, the updating conflicts of histogram calculation are completely avoided, leading to substantial performance improvements. Our best scheme achieves over 55$\times$ speedup compared to the original FLIRT version on CPU, which reduces the registration time from 145 s to 2.6 s for typical $256 \times 256 \times 160$ 3D images on a Kepler platform. Hence, the contributions of this paper are:

- Five CR based registration implement schemes for GPU. To the best of our knowledge, this is the first time the CR method is reported to be employed for image registration on GPUs. Experimental results show that CR outperforms MI, both on speed and accuracy.
- A novel design that completely eliminates the updating conflicts. This highlights the significant advantage of CR over MI on the GPU platforms.
- The trade-off between benefits of exploiting shadow histograms and its concomitant overhead based on comparisons among different schemes.
- An exhaustive and detailed evaluation of the schemes for different generations of GPUs. In this way, we address the stability and portability of the proposed designs while acquiring more details about the hardware capabilities.

The rest of the paper is organized as follows. Section 2 introduces the background of image registration, FLIRT framework and histogram calculation. Section 3 presents the proposed schemes to implement the CR similarity function. Section 4 validates these schemes on hardware. Section 5 discusses the related performance considerations. Section 6 reviews related works. Finally, Section 7 draws the conclusion.

## 2. Background

In this section, we first briefly describe the meaning of image registration, the process of FLIRT framework and the definition of Correlation Ratio. We then present histogram calculation and explain why conflicts exist.

### 2.1. Image registration

Image registration is the process of determining a transformation that maps points from one image (source image) to their homologous points in another image (reference image). It is generally formalized as a cost optimization problem. Its cost function measures the similarity degree between two images. Therefore, the optimization process is attributed as the search for a transform that minimizes the cost function (i.e. maximizes similarity):

$$\begin{aligned}&\textit{Calculate } \boldsymbol{Transform}\\&\textit{such that } \boldsymbol{similarity}(\boldsymbol{A},\ \boldsymbol{B}) \textbf{ is maximized}\\&\textit{where } A = reference\_image,\\&B = Transform(source\_image)\end{aligned}$$

Fig. 1 illustrates the process of image registration. The *Transformed Image* is produced by applying the transform function on the *Source Image*. The similarity between the *Transformed Image* and the *Reference Image* is then calculated, which is returned to the optimizer. Based on the similarity, the optimizer iteratively tunes the transform function until finally the *Transformed Image* and the *Reference Image* show the best similarity.

In order to tune the transform function, we need to parameterize it. In this paper, *affine registration* is considered, so the transform is *affine transform*, which can be expressed as:

$$transformed\_image = M \times source\_image + \vec{b}$$

where $M$ is a $3 \times 3$ matrix; $\vec{b}$ is a vector. The $3 \times 4$ matrix $[M b]$ is labeled as a *transform matrix* that uniquely defines a transform function. Therefore, the transform parameter shown in Fig. 1 is in fact a transform matrix.

During the search process various searching strategies are employed to enhance the possibility of obtaining an optimal transform, while reducing search time. These strategies comprise a searching framework.

### 2.2. FLIRT framework

FLIRT algorithm [3,4] is one of such searching frameworks. It is composed of four stages – each stage focuses on a specific resolution, from 8 mm, 4 mm, 2 mm to 1 mm progressively. A stage contains a series of local searches in which four spaces are traversed: *rotation, translation, scale* and *skew*. Each space is three dimensional $(X, Y, Z)$, so if one dimension is represented by one degree of freedom (DOF), at maximum a 12-DOF search can be performed.

The primary 8 mm searching stage first executes a rotation space searching with a stride of 60 degrees, thus $6 \times 6 \times 6$ times to cover the whole space (360 degrees for all three dimensions). For each checkpoint, a 4-DOF (i.e. rotation and global scale) local search is done. Then another rotation space search with a finer stride of 18 degrees is executed. This time, $(360/18)^3 = 8000$ trials are required. However, unlike the coarse grain search, for every checkpoint, we only evaluate that specific spot instead of initiating a complete local search. Afterwards, three transformation matrices that generate the minimum cost are selected to execute a 7-DOF (i.e. rotation, translation and global scale) full search. The obtained matrices are marked as candidates for the next stage.

In the second 4 mm stage with 4 mm resolution, a 7-DOF (i.e. rotation, translation and global scale) search is applied to the three candidates together with their 30 neighbors (for each candidate, two perturbations on each rotation dimension with 9 degree deviation, four perturbations on scaling with zoom in and zoom out by a factor of 0.1 and 0.2). The best transformation is found out as input for the next stage.

In the 2 mm stage, a 7-DOF (i.e. rotation, translation and global scale), 9-DOF (i.e. rotation, translation and scale) and 12-DOF (i.e. rotation, translation, scale and skew) local search are performed alternately, further approaching the global optimal.