Abstract—This paper is about parallel region growing medical image segmentation on GPUs. Computer-aided diagnosis (CAD) in medical field is playing vital role in clinical practice. Accordingly, medical therapy using CAD is becoming a more challenging research task. Image segmentation for the identification of features and/or objects is a challenging data-dependent task in medical imaging. Some of the image segmentation approaches are based on edge detection, region growing, and thresholding. Region growing (RG) is one of the most conventional and efficient approaches in image segmentation, which is the first step of image analysis. Parallelizing region growing algorithms is essential towards achieving real time performance. In this paper, we have compared GPU based region growing methods and implemented PBF and PLMBF algorithms on Tesla K-40c GPU to segment cardiac images.

Index Terms—Region Growing (RG), Seeded Region Growing (SRG), Graphics Processing Unit (GPU)

I. INTRODUCTION

MEDICAL imaging is one of the prominent fields in today’s world. In medical imaging the identification of the connected region is a challenging task. Region growing starts with a set of pixels called seeds and grows a uniform, connected region from each seed. The key steps to region-growing methods are to define seed(s), number of seeds and a classifying criterion that relies on image properties and user interaction [11]. Among the various studies, seeded region growing (SRG) is a widely used image segmentation method in identifying identical objects due to its lesser computational complexity [11], [5]. However, in practice, it demands high computational cost to the large amount of dependent data to be processed in medical image analysis and still requires efficient solutions [25]. SRG detects the connected pixels, satisfying a certain condition. The neighboring pixels from the set of seed points are evaluated to determine if they have similar characteristics with respect to seed points. The similar neighboring pixels are added to the segmented region. The process continues for further collection of pixels and merging into the segmented region until no more pixels are left for evaluation. The computation time of SRG is proportional to the number of pixels [4], [1]. Because on every iteration, the segmented region is expanded by the number of connected pixels. However, SRG is efficient in identifying smaller region, it requires observable time to segment a larger region [1].

II. RELATED WORK

This section is about related works and their comparisons concerning region growing algorithms. Table I shows comparison table on different implementations from various publications. Chen et al. presents a sketch-based interface for seeded region growing volume segmentation [2]. Adams and Bischof present the algorithm, discusses briefly its properties, and suggest two ways in which it can be employed, namely, by using manual seed selection or by automated procedures [1]. Erdt et al. present a fully automatic approach to quickly enhance and extract the vascular system of the liver from CT datasets [3]. Pan et al. have proposed cuda based implementation of medical image segmentation algorithms indicating the optimum use of GPU [13].

Scherbonyd et al. presents a segmentation method to exploit the parallel processing capabilities of GPU hardware to fasten the computations [18]. In Schenke et al. [16], researchers survey and analyse GPU-based volume segmentation algorithms. They present a framework for the segmentation process and demonstrate the speed advantage using GPU. R. Adams and L.
Bischof presents a new algorithm for segmentation of intensity based images. The method, however, requires number of input seeds to control the formation and expansion of regions for segmentation [1].

Yang and Welch have introduced the use of GPU systems to perform real-time image segmentation [26]. Researchers have designed an automatic seeded region growing (SRG) algorithm, along with a boundary-oriented parallel pixel labeling technique and an automatic seed selection method. Moreover, a seed tracking algorithm is proposed for automatic moving object extraction [4]. In addition the presented GPU based method was tested against a CPU implementation to demonstrate the performance gain of using modern GPUs [3].

### III. REGION GROWING ALGORITHMS, IMPLEMENTATIONS AND PERFORMANCE COMPARISON

A parallel processing can be applied to accelerate region growing computations. One of the alternatives is to use highly computing device i.e. GPU. GPUs have many cores (more than 100) compared to CPUs. The blending functions in Yang and Welch are used to control how the source color (new color fragment from the OpenGL pipeline) and the destination color (usually the pixels in the frame buffer) are combined. Using these features, image segmentation and subsequent image morphology operations can be performed, completely within the graphics hardware [26].

Seeded region growing (SRG) algorithm is very attractive for semantic image segmentation by involving high-level knowledge of image components in the seed selection procedure. However, the SRG algorithm also suffers from the problems of pixel sorting orders for labeling and automatic seed selection [4]. An obvious way to improve the SRG algorithm is to provide more effective pixel labeling technique and automate the process of seed selection [4]. The region seeds, which are located inside the temporal change mask, are selected for generating the regions of moving objects [16].

Table II shows performance comparison table of different implementations from various publications. Basic implementation of Yang and Welch, running on an NVIDIA GeForce4, is 35 times faster for image thresholding and over 30% faster for image morphological operations, compared to a highly optimized software implementation running on a 2.2 Ghz Intel P4 CPU [26].

There have been very few proposed approaches utilizing CUDA to implement image segmentation algorithms. Vineet and Narayanan [23] have presented a fast implementation (10-12 times faster than the best sequential algorithm) of the push-relabel algorithm for mincut/maxflow algorithm for graph-cuts using CUDA. They have used 640x480 size benchmark images and 1024x1024 size synthetic images on an NVIDIA GTX 280 for the performance comparison. Pan et al. have implemented SRG in CUDA. They have compared the SRG implementations in CUDA, Cg, and serial CPU. Experimental results show that the CUDA implementation is 1.6 times faster than the serial CPU implementation [12]. Parallelizing segmentation algorithms using GPUs is one of the most researched area in medical imaging. Yang and Welch have used register combiners to perform thresholding and basic convolutions on 2D color images. On 2.2GHz Intel Pentium 4 CPU, GPU implementation with NVIDIA GeForce4 has demonstrated a 30% speedup over a CPU implementation [26]. Viola et al. have proposed a 3D segmentation method using thresholding.
combined with an interactive visualization, observing nearly an eight-fold speedup over a CPU implementation [24].

<table>
<thead>
<tr>
<th>Publ/Platform</th>
<th>Dataset</th>
<th>Algo.</th>
<th>Result (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>[14] GPU-C2050, 3GB</td>
<td>150MP Image</td>
<td></td>
<td>0.79</td>
<td>x1.2</td>
</tr>
<tr>
<td>[6] GT9600 64 Cores 1GB</td>
<td>max 2.4kx2.4k PBF</td>
<td></td>
<td>4.3-6.2</td>
<td></td>
</tr>
<tr>
<td>[6] GT9600 64 Cores 1GB</td>
<td>max 2.4kx2.4k PLMBF</td>
<td></td>
<td>3.9-8.5</td>
<td></td>
</tr>
<tr>
<td>[6] GT480 480 Cores 1.5GB</td>
<td>max 2.4kx2.4k PBF</td>
<td></td>
<td>9-14.6</td>
<td></td>
</tr>
<tr>
<td>[6] GT480 480 Cores 1.5GB</td>
<td>max 2.4kx2.4k PLMBF</td>
<td></td>
<td>10.2-19</td>
<td></td>
</tr>
<tr>
<td>[13] GeForce 6800 (16 seeds, 60 iter.)</td>
<td>Brain-256x256x256</td>
<td></td>
<td>4.42</td>
<td></td>
</tr>
<tr>
<td>[12] AMD ANTHLON 64 X2 3800 with GeForce 7800GT 256MB</td>
<td>Brain (MRI-152x154x181)</td>
<td>SRG</td>
<td>0.384</td>
<td></td>
</tr>
<tr>
<td>[8] Linux &amp; Tesla C1060, 240 Cores 4GB, Intel Xeon 2.5 GHz 7.8 GB</td>
<td>Aerial Images of different sizes</td>
<td></td>
<td>6.86</td>
<td></td>
</tr>
<tr>
<td>[16] GeForce 6800GT 256 MB</td>
<td>Brain- Dataset (256x256x192)</td>
<td>Diffusion based RG</td>
<td>204 MVoxels/s</td>
<td></td>
</tr>
<tr>
<td>[3] GPU(8800 GTS, 640 MB VRAM)</td>
<td>CT datasets (278x308x99)</td>
<td>Auto. RG</td>
<td>1.59</td>
<td>15</td>
</tr>
<tr>
<td>[20] AMD 64 Quadcore CPU, 8 GB with GeForce GTX 570</td>
<td>Stained Colon Tissue Images</td>
<td>Split &amp; Merge</td>
<td>0.25ms</td>
<td></td>
</tr>
<tr>
<td>[17] 8500 GT with Intel Core i5-3570 3.4GHz quad core 8 GB</td>
<td>Lung</td>
<td>SRG</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>[17] GTX285 1 GB, Intel Core i5-3570 3.4GHz quad core 8 GB</td>
<td>Lung</td>
<td>SRG</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td>[17] 8500 GT with Intel Core i5-3570 3.4GHz quad core 8 GB</td>
<td>Colon</td>
<td>SRG</td>
<td>1.3</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II: Performance Comparison Table: Publ. represents referred publication.

Preliminary results by Yang and Welch show a performance increase of over 30% using an NVIDIA GeForce4 when compared to an implementation using Intel MMX optimized code on 2.2 Ghz Intel P4 [26].

IV. RESULTS AND CONCLUSION

The cardiac images [19] are segmented using region growing on GPU (Tesla K40c). The test images are shown in Figure 1a & 2a with its segmented images using PBF (Figure 1b and 2b) and PLMBF (Figure 1c and 2c) algorithms [6].

In this paper, the performance and parameter comparison is done based on literature and methodology is proposed to evaluate the platform and algorithm. The method reduces platform selection problem for computationally effective algorithms for various datasets. The paper has included results of different implementations of RG based segmentation of cardiac images. The region growing algorithm can be further improved for obtaining more accurate segmented area. The parameters such as number of seeds, seed selection, neighbourhood criteria, threshold can be studied and modified to get better performance results.

ACKNOWLEDGMENT

This work is supported by High Performance Soft-tissue Navigation ( HIPERNAV - H2020-MSCA-ITN-2016 ) funded through a Marie Skodowska-Curie grant.

REFERENCES

Fig. 2: Cardiac second image with segmented images using PBF and PLMBF


