Real-Time-Capable GPU-Framework for Depth-Aware Rigid 2-D/3-D Registration

Matthias Utzschneider[1], Jian Wang[2], Roman Schaffert[1,2], Anja Borsdorf[2], Andreas Maier[1,3]

[1]Pattern Recognition Lab, FAU Erlangen-Nuremberg, Erlangen
[2]Siemens Healthineers, Forchheim
[3]Erlangen Graduate School in Advanced Optical Technologies, Erlangen
matthias.utzschneider@fau.de

Abstract. 2-D/3-D image fusion is used for a variety of interventional procedures. Overlays of 2-D images with perspective-correctly rendered 3-D images provide the physicians additional information during the interventions. In this work, a real-time capable 2-D/3-D registration framework is presented. An adapted parallelization using GPU is investigated for the depth-aware registration algorithm. The GPU hardware architecture is specially taken into account by optimizing memory access patterns and exploiting CUDA-texture memory. The real-time capability is achieved with a median runtime of one 2-D/3-D registration iteration of 86.1 ms with an median accuracy of up to 1.15 mm.

1 Introduction

Imaging gained more and more importance in interventional medicine over the last decades. In the family of diverse medical imaging modalities, X-ray fluoroscopy is the standard routine for many interventional procedures. The fluoroscopic images are acquired intra-operatively by the interventional C-arm system. Intra-operative images are often combined with pre-operative images acquired by Computed Tomography (CT) or Magnetic Resonance Tomography (MRT) to give the physicians information about the 3-D position of interventional devices to provide guidance. The interventional fluoroscopies are combined with 3-D images by rendering a perspective-correct view of the 3-D image and superimpose it onto the 2-D fluoroscopy. This procedure is called 2-D/3-D image fusion. For an accurate overlay of fluoroscopy and CT image, the application has to compensate for misalignments often introduced by patient movement. To maintain the accuracy of the overlay, 2-D/3-D registration algorithms are applied. Furthermore, any interventional application has to meet time constraints. These circumstances make it necessary for the implementations to be performance-oriented and adjusted to the processing architecture. In the recent years, the use of special hardware like Graphic Processing Units (GPUs) has been investigated for computationally intensive tasks in high performance computing. For example, rendering of Digitally Reconstructed Radiographs (DRRs) is an often used approach in 2-D/3-D registration, which is well parallelizable and computational

intensive. GPUs became very common for implementations of 2-D/3-D registration algorithms [1].

In this paper a real-time registration implementation of an algorithm [2] based on the point-to-plane correspondence (PPC) model [3] for rigid registration is presented. The proposed approach makes use of the high performance of GPUs for applications with a potential for high parallelism and exploits special hardware optimizations for GPUs. This paper is structured as follows. In Sec. 2, an overview of the registration algorithm is given and the parallelization method for the overall algorithm and for computationally intensive sub-steps is described. Runtime performance is evaluated and results are presented in Sec. 3. In Sec. 4, results are summarized and future development possibilities are outlined.

2 Materials and methods

2.1 Depth-aware registration algorithm

To correct the misalignments and preserve the accuracy of the 2-D/3-D overlay, a transformation $\mathbf{T}_{\mathrm{Reg}} \in \mathbb{R}^{4\times4}$ has to be estimated to compensate for the motion. $\mathbf{T}_{\mathrm{Reg}}$ is determined by depth-aware registration based on the feature-based PPC model [3]. With this model, a linear system of equations is formulated using a set of feature points $\{\mathbf{w}\}_{\mathrm{corr}}$ for the observed volume and a set of corresponding 2-D points $\{\mathbf{p}'\}_{\mathrm{corr}}$ on the fluoroscopic image I_0 [3]. For the proposed approach, an initial set of feature points with an high 3-D image gradient, e.g. on bone surfaces $\{\mathbf{w}\}_{\mathrm{init}}$ is extracted from the volume using the 3-D Canny filter [4]. In the preselection step, occluding contour points $\{\mathbf{w}\}_{\mathrm{sel}}$ are selected from the current viewing direction [3]. In the tracking step, correspondences for $\{\mathbf{w}\}_{\mathrm{sel}}$ are searched with Depth-layer-gradient Images (DLIs) [5] of the volume and gradient images of the fluoroscopy using a patch-matching approach. This patch-matching routine [2] determines the set of feature points and correspondences $\{\mathbf{w}, \mathbf{p}'\}_{\mathrm{corr}}$ for the estimation of $\mathbf{T}_{\mathrm{Reg}}$. The transformation $\mathbf{T}_{\mathrm{Reg}}$ is computed by solving the system of equations using iterative-reweighted least square optimization.

2.2 GPU-parallelization of the registration algorithm

The above algorithm offers the ease of parallelization to reduce the overall runtime of the approach. To harness the high parallel processing potential of GPUs, the algorithm is implemented in Nvidia CUDA for Nvidia GPUs. All sub-steps are implemented with the parallelization adapted to the sub-steps properties. The rendering of DLIs is similar to the DRR generation, which is well parallelizable for GPUs [1]. For the DLI-rendering, the volume is initially stored in a 3-D CUDA-texture. The advantage of using CUDA-textures for sampling is the efficient hardware-implemented interpolation and 3-D spatial caching. The preselection as well as the patch-matching routine are executable for every feature point independently. Therefore, the GPU-implementation yields great potential for a runtime improvement. The well parallelizable steps, i.e. preselection and

patch-matching, are performed on the GPU for all initial feature points in parallel. Furthermore, all feature points are processed in the GPU memory to avoid CPU-GPU memory transfer latencies. The final estimation of $\mathbf{T}_{\mathrm{Reg}}$ is performed on the CPU and is the sole step executed on the CPU. Therefore, only feature points with correspondences as well as the correspondences themselves are transferred from the VRAM to the processors main-memory. An overview of the implemented GPU registration framework is given in Fig. 1.

2.3 Patch-matching parallelization

To improve the performance of the patch-matching step, two approaches are investigated. The routine uses Gradient Correlation [6] on multiple patches in the vicinity of each feature point to find corresponding points between DLIs and the gradient image of the fluoroscopy. A feature-based and a patch-based parallelization is examined to achieve real-time performance. The feature-based method distributes all patch-matching computations necessary for one single feature point to one single CUDA-thread each. This approach increases the simultaneously executed computations significantly compared to a multi-core CPU implementation with at least 32 threads processing in parallel. In the patch-based routine, each patch processed for a feature point position is assigned to one CUDA-thread. Patches processed for one feature point overlap strongly and therefore GPU memory is accessed in close spatial vicinity. To benefit from caching and to coalesce memory accesses, all patches for one feature points are processed in parallel by concurrently working threads on the GPU. To further improve data access read-only data caching is used for DLI-images and fluoroscopy gradient images, which enables L1-caching of all accessed data.

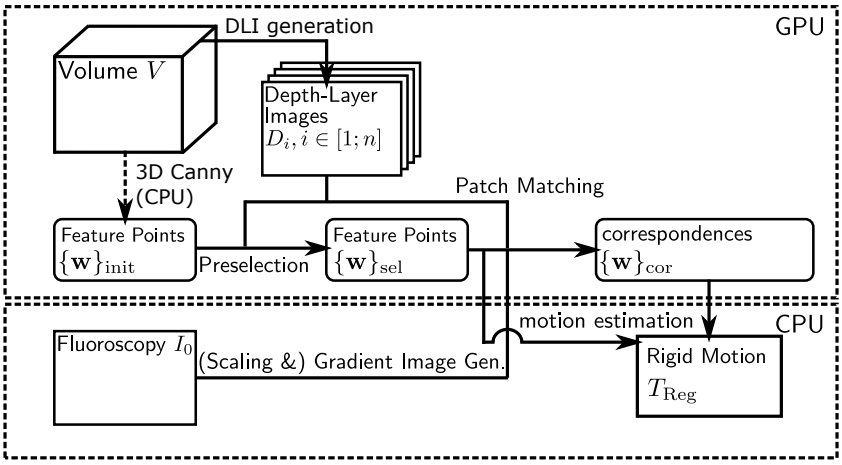

Fig. 1. Overview of the GPU-based registration Framework. All data and steps within the dashed lines reside respectively are executed on the denoted hardware architecture.

2.4 Depth-layer-image sampling

DLIs are used to find correspondences for feature points in a patch-matching routine. The DLI-rendering is implemented in a CUDA kernel using a ray-based parallelization. The memory access patterns of the kernel is adapted to the spatial caching of CUDA-textures. To further improve the runtime-performance of the kernel, the sampling step k_s is varied. By increasing k_s, the necessary number of volume-texture evaluations decreases, but it can worsen the accuracy due to lack of evaluated sampling points. The optimal sampling step is determined, which does not deteriorate the mean Target Registration Error (mTRE) [7] of the overall approach and yields the best runtime result for DLI-rendering.

3 Results

The evaluation of the runtime performance is done on a high performance test-system using a modern Nvidia GPU for professional rendering (Intel Xeon E4-1620v3@3.5GHz, 8GB DDR4-RAM, NVIDIA Quadro K2200 (640 CUDA-Cores, 1 GHz, 5 SMs). The evaluation setup from [3] with ten image sequences of a thorax phantom is used for the runtime evaluation. All runtime measurements are averaged over ten runs. The accuracy of the approach is indicated by the mTRE and the mean Projection Error (mPE) [3]. In the first evaluation, the different patch-matching methods are evaluated for the best runtime. The best volume sampling step for rendering DLIs is determined by an accuracy and runtime evaluation. The overall runtime-performance is evaluated using the determined sampling step and patch-matching approach.

The volume sampling step k_s is initially chosen as 1.0 voxel per sampling step. The evaluation is done by increasing k_s by 0.2 per evaluation step. The runtime and the mTRE for the chosen k_s are measured. With $k_s = 2.0$ voxels, the best runtime is achieved without increasing the overall mTRE, see Fig. 2 and Fig. 3. The evaluation results are averaged over all image sequences (#1-#10). The run-time performance of the DLI-rendering step is increased by 224%.

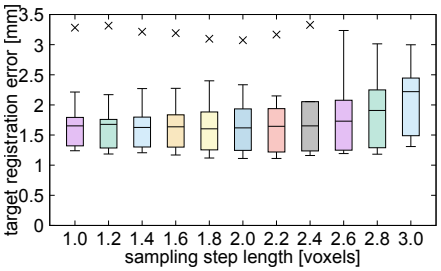

Fig. 2. Error evaluation for different k_s [sampling step length].

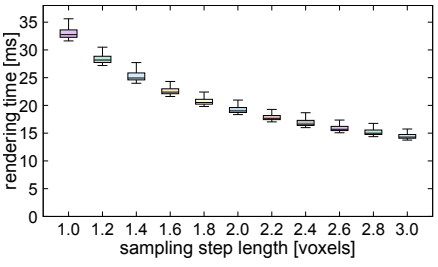

Fig. 3. Runtime evaluation for different k_s [sampling step length].

Table 1. Average and best runtimes of the patch-matching implementations.

Implementation	Average [ms]	Best [ms]	Speed-Up (Average)
GPU (feature-based):	92.11	84.82	-
GPU (patch-based):	40.31	34.53	$\times 2.28$

The evaluation for the patch-matching methods is performed with image sequence #10. The patch-based approach outperforms the feature-based implementation by around 228%, see Tab. 1.

For the evaluation of the overall framework, the patch-based implementation and a sampling step of $k_\mathrm{s} = 2.0$ voxels per sampling step is used. An average runtime of $86,1$ ms and a best runtime of $73,1$ ms is reached in the evaluation setup, which is considered real-time capable for dynamic 2-D/3-D registration [8]. In comparison with a single-core CPU implementation using an OpenGL DLI-rendering a speed-up of $\times 41.2$ is achieved, see Tab. 3. The mTRE depending on the image sequence is 0.86 mm at best and 1.21 mm in average and does not exceed 2.0 mm for 8 out of 10 sequences, see Tab. 2. The mPE does not exceed 2.0 mm for all sequences and is 1.15 mm in average and 0.80 mm at best, see Tab. 2.

4 Discussion

In this paper, a GPU-based real-time 2-D/3-D registration framework is presented. A depth-aware registration approach using the PPC model is implemented for Nvidia GPUs. The framework makes use of the high parallelism of GPUs and is optimized for Nvidia GPU architecture exploiting CUDA-texture memory and read-only data-cached memory access. The implementation is adjusted for the special architectural demands of GPUs by coalescing GPU memory accesses and taking spatial caching-strategies of CUDA-texture memory into

Sequence	#frames	$mTRE$ [mm]	$mRPE$ [mm]
1	33	4.39 ± 0.35	1.18 ± 0.12
2	93	1.23 ± 0.40	1.03 ± 0.22
3	111	1.16 ± 0.20	1.13 ± 0.21
4	111	1.08 ± 0.28	1.06 ± 0.19
5	110	1.53 ± 0.24	1.17 ± 0.28
6	101	0.83 ± 0.23	0.80 ± 0.12
7	105	3.48 ± 2.92	1.27 ± 0.47
8	117	1.77 ± 0.68	1.29 ± 0.14
9	114	0.86 ± 0.24	0.80 ± 0.16
10	121	1.19 ± 0.25	1.49 ± 0.23

Table 2. $mTRE$ and $mRPE$ and standard deviation of the implemented approach.

Table 3. Average and best runtimes of one registration iteration for the different implementations.

Implementation	Average [ms]	Best [ms]	Speed-Up (Average)
CPU (single-core$\|k_s = 1.0$)	3553.27	2633.44	-
GPU (feature-based$\|k_s = 1.0$)	174,36	171,93	$\times 20,38$
GPU (patch-based$\|k_s = 2.0$)	86.11	73.47	$\times 41,2$

account. Real-time capability [8] and a runtime performance of 13 frames per second at best is achieved with the GPU-framework and an adapted volume sampling step. Only one approach for 2-D/3-D registration reaches a similar framerate and a higher runtime-performance is not reached by approaches reported in literature [9] to the best knowledge of the author.

For further development, the GPU framework is to be completed by implementing the final motion estimation step on the GPU. With the achieved real-time capability multi-start approaches similar to [10] are feasible to be investigated.

References

1. Abdellah M, Eldeib A, Owis MI. GPU Acceleration for Digitally Reconstructed Radiographs using Bindless Texture Objects and CUDA/OpenGL Interoperability. Conf Proc IEEE Eng Med Biol Soc. 2005; p. 4242–5.
2. Borsdorf A, Wang J. Verfahren zur 2D-3D-Registrierung, Recheneinrichtung und Computerprogramm. DE Patent 102015208929, May 13; 2016.
3. Wang J, Borsdorf A, Heigl B. Gradient-based differential approach for 3-D motion compensation in interventional 2-D/3-D image fusion. Proc Int Conf 3D Vis. 2014;1:293–300.
4. Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell. 1986;8(6):679–98.
5. Wang J, Borsdorf A, Hornegger J. Depth-layer-based patient motion compensation for the overlay of 3D volumes onto x-ray sequences. Proc BVM. 2013;3(13):128–22.
6. Wein W, Roeper B, Navab N. 2D/3D registration based on volume gradients. Proc SPIE 2005. 2005;5747:144–50.
7. van de Kraats E, Penney GB, Tomaževič D, et al. Standardized evaluation methodology for 2-D-3-D registration. IEEE Trans Med Imaging. 2005;24(9):1177–89.
8. Heimann T, Mountey T, John M, et al. Real-time ultrasound transducer localization in fluoroscopy images by transfer learning from synthetic training data. Med Image Anal. 2014;18(8):1320–8.
9. Liao R, Zhang L, Sun Y, et al. A Review of Recent Advances in Registration Techniques Applied to Minimally Invasive Therapy. Med Image Anal 2013. 2013;5(15):983–98.
10. Otake Y, Wang A, Stayman JW, et al. Robust 3D-2D image registration: application to spine interventions and vertebral labeling in the presence of anatomical deformation. Phys Med Biol 2013. 2013;1(58):8535–53.