Evaluation of Image Processing Algorithms on ARM Powered Mobile Devices

Damir Demirović [1] Amira Šerifović-Trbalić [1] Naser Prljača [1] Philippe C. Cattin [2]

[1]University of Tuzla, Faculty of Electrical Engineering, Tuzla, Bosnia and Herzegovina

[2]Medical Image Analysis Center, University of Basel, Basel, Switzerland

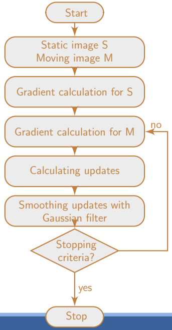
29. May 2014

Introduction

- Image enhancement plays an important role in different research fields such as medical image analysis.
- The most popular methods for image processing and analysis are very resource hungry.
- We investigate the possibilities and limitations of well known Demons algorithm implemented for mobile Android device.
- To our knowledge there is no implementation of Demons registration method for Android platform.
- The contributions of this work are:
  - Implementation of the Thirions Demons registration algorithm on ARM CPUs running Android.
  - Speeding-up the implementation using hardware optimizations
  - Performance evaluation for Gaussian parameters.

Demons registration

- Image registration tries to find a mapping from one coordinate system of one image to the coordinate system of the other image.
- The one of the most widely used deformable registration algorithm is Demons registration algorithm.
- Advantages: widely known, simple to use, linear computational complexity, easily parallelizable.



Start

Static image S
Moving image M

Gradient calculation for S

Gradient calculation for M

Calculating updates

Smoothing updates with Gaussian filter

Stopping criteria?

no

yes

Stop

Figure : Flowchart of Demons algorithm

Implementation for an Android device

- Android platform is now a world dominant in mobile platform market.
- Applications are developed in the Java language using Android Software Development Kit (SDK).
- Resouce consuming parts can be with C/C++ language using Native Development Kit (NDK)
- With NDK we can exploit Single Instruction, Multiple Data (SIMD) optimizations
- For Tegra 3 and Samsung Exynos platforms optimizations for ARM V7 instructions with NEON instructions can be used.

Hardware and Software

- Used devices were: desktop (small laptop, home PC, server machine), cell phones and tablet.
- GCC 4.6.3 compiler with optimizations using GNU/Linux.

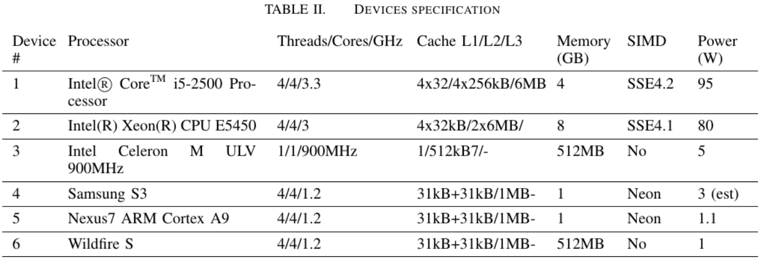


TABLE II. DEVICES SPECIFICATION

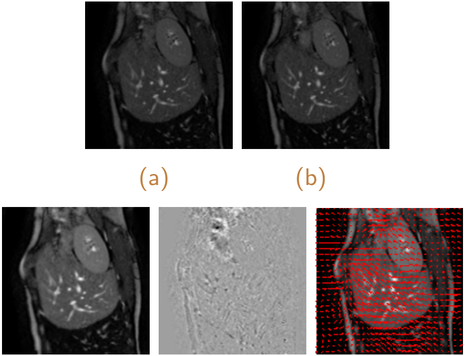
Device #	Processor	Threads/Cores/GHz	Cache L1/L2/L3	Memory (GB)	SIMD	Power (W)
1	Intel® Core™ i5-2500 Processor	4/4/3.3	4x32/4x256kB/6MB	4	SSE4.2	95
2	Intel(R) Xeon(R) CPU E5450	4/4/3	4x32kB/2x6MB/	8	SSE4.1	80
3	Intel Celeron M ULV 900MHz	1/1/900MHz	1/512kB7/-	512MB	No	5
4	Samsung S3	4/4/1.2	31kB+31kB/1MB-	1	Neon	3 (est)
5	Nexus7 ARM Cortex A9	4/4/1.2	31kB+31kB/1MB-	1	Neon	1.1
6	Wildfire S	4/4/1.2	31kB+31kB/1MB-	512MB	No	1

Table : Optimization parameters for GCC compilers

Platform	Optimization string
Desktop	-O2
ARMv7	-march=armv7-a -mtune=cortex-a9 -mfloat-abi=softfp -mfpu=neon
ARMv6	-march=armv6 -marm -mfloat-abi=softfp -mfpu=vfp

Results

- Two MRI liver images 256 × 256 were used in all experiments.
- We varied Gaussian parameters in Demons: kernel width, and σ.



(a) (b)

(c) (d) (e)

Figure : Images used for registration a) fixed b) moving c) registered d) difference after registration e) overlayed deformation field

Results (2)

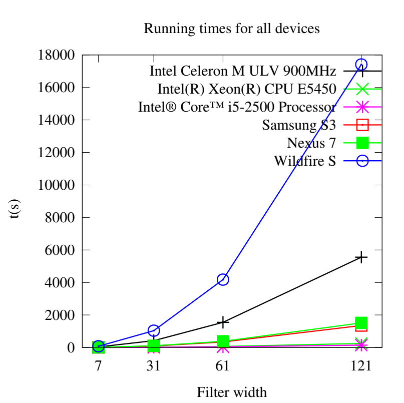


Figure : Registration results for all devices

	Parameters		Device #					
Image size	width	σ	1	2	3	4	5	6
256 × 256	7	1.0	1.5	1.2	34.3	8.6	9.7	30.1
	31	5.0	12.8	18.2	422.7	91.2	103.3	1033.8
	61	10.0	42.2	63.2	1549.9	338.8	379.6	4175.5
	121	20.0	144.7	245.9	5558.1	1351.2	1514.3	17414.5

Discussion and Future Work

- Results shows, in general, that mobile devices cannot compete with desktop CPUs in higly demanding tasks like image registrations.
- Can be used as an alternative with smaller Gaussian kernel widths, (i.e. smaller deformations).
- The best results were about 9 times slower than desktop.
- Further research in this direction can evaluate tradeoff between registration precision and execution times regarding floating point optimizations.
- Another possibility is multicore implementation of Demons algorithm on the ARM CPUs, i.e. implementation using Threading Building Blocks (TBB).

Acknowledgement

This work has been supported by the SCOPES project of the Swiss National Science Foundation (http://www.snf.ch)

Thank you for your attention