Looking at Ultrasound Signal Processing on Low-Power GPUs

Anne C. Elster (*) and Bjørn Tungesvik

Dept. of Computer & Info. Science
Norwegian University of Science and Technology (NTNU)

(*) Currently on Sabbatical at
ICES (Inst. For Computational Science & Engineering)
University of Texas at Austin (until Aug 2016)
Acknowledgements

- My Master student **Bjørn Tungesvik** who did all the implementations!
Acknowledgements

• My Master student Bjørn Tungesvik who did all the implementations!

• Optimization ideas from my PhD student Rune Jensen

• Prof. Bjørn Angelsen and his SURF team including:
  – Ola Fineng Myhre, PhD student and mentor
  – Ole Martin Brende, PhD student
  – Johannes Kvam, PhD student (Elster is co-advisor)
  – Stian Solstad (Master student, 2015)
  – Ali Fatemi (Master student, 2015)
GPU history and HPC-Lab at NTNU

- Started working on GPUs for compute in 2006 with two of my master students
- Founded HPC-Lab in 2008, same year also got into NVIDIA's Professor Partnership program
- Elster has advised several PhD students and 30+ master theses on GPU computing
  (Elster has so far been main advisor for 66 master students)
- Finishing up CUDA book based on work with classes and students
- PI/Co-PI of NVIDIA CUDA/GPU Centers at both NTNU and UT Austin
Close collaboration with NTNU’s MedTech Imaging groups (since 2006)

HPC-Lab members and Tucker Taft, Spring 2014
Trondheim, Norway on the world map
NTNU Gløshaugen
(formerly Norwegian Institute of Technology)

U of Texas at Austin
Inspirational questions:

• Can we use embedded devices for High Performance Computing (HPC)?

• If so, how well do they do for some basic algorithms?

• How about filtering for bleeding edge ultrasound processing?
  – Q: Why do we care about this?
  – A: Move processing capability to the wand!!
What is Ultrasound?

- American Standards Instituted defines it to be > 20KHz

- Upper frequency limit of hearing by humans (may have auditory sensation of high-intensity ultrasound waves if feed sound directly to bone)
Ultrasound fun facts

- Bats can detect frequencies beyond 100kHz

- “Mosquito” devices
  - Teenagers 17.4KHz-20KHz anti-loitering.
  - Parent-avoiding ringtones..

- Polaroid introduced sonar based autofocus in 1978 with its Sonar One Step camera
  - The popular SX-70 uses same ultrasound tech later licensed for many applications
  - Later licensed for lot of other applications
3D ultrasound

Used for:

• Early detection of tumors
• Visualization of fetuses
• Blood flows in organ and fetuses

• http://www.ta.no/grenland/det-forste-portrettet/s/1-111-2263836
How does medical ultrasound work?

• Wand with array of piezo-electric elements
  – If applied voltage -> vibrate
  – If vibrate -> generate voltage

1. Transmit HF (1-5MHz) sound pulse
2. Pulse hits tissue boundaries
   E.g. fluid-soft tissue, soft-tissue-bone
3. Some wave reflected back to prove, some travel further
4. Reflect waves picked up by probe & relayed
5. Calculate dist from probe to tissue/organs using speed of sound in tissue (540m/s)
6. Machine displays distance and intensities of echoes as image
Beamforming

Direct ultrasound waves (signals) to some focus by delaying & combining signals sent to element
Beamforming

Direct ultrasound waves (signals) to some focus by delaying & combining signals sent to element

In ultrasound:
• Transmit with fixed focus
• Receive with either fixed or dynamic focus
• Standard beamforming: DAS (delay&sum)
Beam forming

Appearance in image
Scattering
Overlap
Irregular Wavefront

Irregular mixture of fat and tissue $\rightarrow$ Heterogenous characteristics

Ultrasound machines assumes 1$^{st}$ order scattering, so
Multiple scattering noise
SURF Ultrasound Imaging
(Second Order Ultrasound Field or dual-band)

- Normal pulse

- SURF pulse
Ultrasound issues contin.

• Using same transmit and receiver beam
  -> large point-spread function (blurring) at each depth
  -> limited ability to resolve scattering

• Reducing point-spread fn implies synthetic focus at each depth!
Dynamic Aperture Focusing

- Adjust aperture of beam as we receive ensuring have beam at each focus $P$

$$\Delta x = \frac{\lambda F}{D},$$

$\Delta x$ – beam width
$\lambda$ – wavelength
$F$ – focus point
$D$ – aperture
Ultrasound issues contin.

• Reducing point-spread fn implies synthetic focus at each depth!
  – Achieved by creating filter based on Westerwelt eqn.,
    -- simplified model of “Nonlinear Imaging with dual band pulse complexes” by Angelsen and Tangen

• Transversal filtering technique allows for synthetic depth variable for 1\textsuperscript{st} order scattering
What we achieved:

• Our initial goal was 20 FPS,
  – i.e 50 ms of processing per frame.

• Our synthetic dynamic focusing algorithm on the Jetson TK1 is able to process a frame in **24 milliseconds**!

• Our method also tested on more powerful GPU PC hardware --able to process same data set in **8.8 ms**.
MIMD Parallella and SIMT Kepler

SIMT

Instruction stream

Data streams

CU | CU | CU | CU

MIMD

Instruction stream

Data stream

CU | CU | CU | CU
## Memory bandwidth test
(Using NVIDIA Bandwidth test and STREAM)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Memory Module</th>
<th>Transfer speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W DRAM</td>
<td>Pageable</td>
<td>4964.3 MB/s</td>
</tr>
<tr>
<td>Copy to device</td>
<td>Pageable</td>
<td>1404.5 MB/s</td>
</tr>
<tr>
<td>Copy to device</td>
<td>Page-locked</td>
<td>998.2 MB/s</td>
</tr>
<tr>
<td>DEVICE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy from Device</td>
<td>Pageable</td>
<td>1447.7 MB/s</td>
</tr>
<tr>
<td>Copy from Device</td>
<td>Page-locked</td>
<td>5464.4 MB/s</td>
</tr>
<tr>
<td>Device to device</td>
<td>Pageable</td>
<td>11885 MB/s</td>
</tr>
<tr>
<td>Device to device</td>
<td>Page-locked</td>
<td>3127.7 MB/s</td>
</tr>
</tbody>
</table>

This test showed that the Jetson much faster than Parallella board.
Julia, Matrix mult & N-body
Testing -- 2D FFTs
64x64, 128x128, 256x256 and 512x512
Testing: Memory Layout

Bank 0
Code

Bank 1
Ping buffer

Bank 2
Pong buffer

Bank 3
Twiddle factors
Mailbox
FFTs and Batched FFTs (128x128)
RF data without & with adjustments
CIRS Phantom (Model 040GSE)

1. Near field – 5 targets
   - Depth 1-5mm
   - Diam. 100 microns
   - 1 mm spacing

2. Vertical group with 4 targets
   - 1-4cm
   - Diam. 1-100 microns
   - 10 mm spacing

3. Horizontal group with two gray scale targets
   - Contrast resol. +6 and > 15db, Diam 8mm

4. Horizontal group, 3 targets
   - Depth 4cm
   - Diam. 100 microns
   - Spacing 10 mm
Dataset

- Acquired using 40MHz sampling freq.
- Transducer with 128 channels
- Gave matrix of ca. 128 x 2080
- Divided into 40 windows (→ 52 samples/window)
- With overlap: 104 samples/window
- Adding padding to avoid circular convolution: 144
- Padding to nearest 2-factor: 256
- Pad also laterally: 128 to 256
- → need 40 FFTs, inv FFT and Hadamards products/frame
Convolution

\[ Y(n) = F(n)G(n) \]

\[ \text{IFFT2D}(Y) \]
4mm
Conclusions

• Ultrasound processing requires High Performance Computing

• HPC = Heterogenous and Parallel Comptuing

• Realt-time requirement met on the Tegra TK1 kit for our Ultrasound filtering for synthetic dynamic focusing
Furture work

- Look at the Tegra TX1!
- Move the processing to the transducer
TK1/Kepler

- GPU: SMX Kepler: 192 core
- CPU: ARM Cortex A15
  - 32-bit, 2 instr/cycle, in-order
  - 15GBs, LPDDR3, 28nm process
- GTX 690 and Tesla K10 cards have 3072 (2x1536) cores!
- Tesla K80 is 2.5x faster than K10
  - 5.6 TF TFLOPs single prec.
  - 1.87 TFLOPS Double prec.
- Nested kernel calls
- Hyper Q allowing up to 32 simultaneous MPI tasks

TX1/Maxwell

- GPU: SMX Maxwell: 256 cores
  - 1 TFLOPs/s
- CPU: ARM Cortex-A57
  - 64-bit, 3 instr/cycle, out-of-order
  - 25.6 GBs, LPDDR4, 20nm process
- Maxwell Titan with 3072 cores
- API and Libraries:
  - Open GL 4.4
  - CUDA 7.0
  - cuDNN 4.0
Thank you!

And to my Master student **Bjørn Tungesvik** who did all the implementations!

For further questions contact:

anne.elster@gmail.com