GPU applications in Cancer Radiation Therapy at UCSD

Steve Jiang, UCSD Radiation Oncology
Amit Majumdar, SDSC
Dongju (DJ) Choi, SDSC
Conventional Radiotherapy

**SIMULATION:** Construction, Dij

**PLANNING:** Treatment Design, Optimization, Dose Calculation

**Treatment**

Online ART (Adaptive Radiation) We Want

1. Simulation
2. Planning
3. On-board Imaging
4. Re-planning
5. Treatment

Repeat
CPU vs GPU

CPU
- Control
- ALU
- ALU
- ALU
- Cache
- DRAM
- PCIe

GPU
- Memory
- SIMD parallelization
- DRAM
Simple Radiation Application in GPU: DRR (Digitally Reconstructed Radiographs)

- Image pixel calculated in each thread
- CT volume in Texture memory
- Few seconds in CPU becomes few 1/10s seconds in GPU
Online Re-planning Process Needs..

• CBCT (Cone-Beam CT) Reconstruction
• Deformable Image Registration
• Re-contours
• Dose Calculation
• Plan Re-Optimization
Deformable Image Registration with ‘Demons’

- Morphing one image (volume) into another
- Displacement vector on each pixel in each thread
- Iterate until displacement vector becomes less than criteria
- For CT size 256X256X100, the speed up was ~50X (C1060 vs. Intel Xeon 2.27GHz)

\[
\tilde{d}(x) = \frac{2[I_0(x) - I_1(x)][\nabla I_0(x) + \nabla I_1(x)]}{\|\nabla I_0(x) + \nabla I_1(x)\|^2 + [I_0(x) - I_1(x)]^2/K^2},
\]
Cone Beam CT

- CBCT reconstruction problem

\[ I(u, v) = I_0 e^{-\int_{s}^{(u,v)} f(x, y, z) \, dl} \]

\[ g(u, v) = -\log\left(\frac{I}{I_0}\right) = \int_{s}^{(u,v)} f(x, y, z) \, dl \]

- Mathematical problem

Find a volumetric image \( f(x, y, z) \) given \( g(u, v) \) s.t. \( Pf \sim g \)

- \( P \) --- projection operator in cone beam geometry
- \( f(x, y, z) \) --- volumetric image to be reconstructed
- \( g(u, v) \) --- measured projections on imager
Algorithm

- Optimization procedure

1. Update
   \[ v = f - \frac{\mu}{\lambda} P^T (Pf - g) \]

2. Minimize
   \[ f = \arg \min_r J[f] + \frac{\lambda}{2} \| f - v \|^2 \]

3. Correct
   \[ f(x, y, z) = 0, \text{ if } f(x, y, z) < 0 \]
Computation of $g = Pf$

- Forward calculation
- Ray tracing algorithm on each GPU thread for one element in $g$

- Multiple threads run in parallel ~100 times faster than CPU implementation
Computation of $f = P^T g$ 

- Backward calculation
- Ray tracing algorithm on each GPU thread
- Multiple threads run in parallel, however…Slow!!
- $T = 46 \text{ min}$ for a typical clinical case
GPU-friendly Algorithm for $P^T$

- Consider $g = Pf$ in a continuum case

$$P[f(x, y, z)](u, v) = \int_L dl \, f(x, y, z)$$

- Mathematically, $P^T$ defined as

$$\langle f, P^T g \rangle = \langle Pf, g \rangle \text{ for } \forall f, g$$

- Solve for $P^T$

$$P^T [g(u, v)](x, y, z) = \frac{L^3(u^*, v^*)}{L_0 l^2(x, y, z)} g(u^*, v^*)$$

- Multiple threads run in parallel. Speed up by a factor of $\sim 20$
Results

- Digital NCAT phantom in a realistic GE CT geometry
- Case #3: Low (500) projections acquired with low (20) mAs protocol

![Ground Truth, FBP, TNLM images](image-url)
Online Re-planning Process Needs..

- CBCT Reconstruction
- Deformable Image Registration
- Re-contours
- Dose Calculation
- Plan Re-Optimization
GPU-based Treatment Planning @ UCSD

- Dose calculation
  - gFSPB: finite size pencil beam model
  - gDPM: DPM Monte Carlo code to GPU
- Plan optimization
  - gFMO: fluence map optimization
  - gDAO: direct aperture optimization
  - gVMAT: optimization for volumetric modulated arc therapy
Dose Calculation

FSPB Beam Cross-profiles

Monte Carlo histories
Monte Carlo Dose Calculation on GPU

Start

Transfer data to GPU including random # seeds, cross sections, and pre-generated e-tracks etc.

a). Clean local counter
b). Simulate one MC history on thread #1
c). Put dose to global counter

......

a). Clean local counter
b). Simulate one MC history on thread #1
c). Put dose to global counter

No

 Reach a preset # of histories?

Yes

Transfer data from GPU to CPU

End
# Results

(Photon point source, 6MV on head)

Execution time $T$, and speed-up factor $T_{CPU}/T_{GPU}$ for four different testing cases.

<table>
<thead>
<tr>
<th>Source type</th>
<th># of Histories</th>
<th>Case</th>
<th>$T_{CPU}$ (sec)</th>
<th>$T_{GPU}$ (sec)</th>
<th>$T_{CPU}/T_{GPU}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20MeV Electron</td>
<td>$2.5 \times 10^6$</td>
<td>water-lung-water</td>
<td>117.5</td>
<td>2.05</td>
<td>57.3</td>
</tr>
<tr>
<td>20MeV Electron</td>
<td>$2.5 \times 10^6$</td>
<td>water-bone-water</td>
<td>127.0</td>
<td>1.97</td>
<td>64.5</td>
</tr>
<tr>
<td>6MV Photon</td>
<td>$2.5 \times 10^8$</td>
<td>water-lung-water</td>
<td>1403.7</td>
<td>18.6</td>
<td>75.5</td>
</tr>
<tr>
<td>6MV Photon</td>
<td>$2.5 \times 10^8$</td>
<td>water-bone-water</td>
<td>1741.0</td>
<td>24.2</td>
<td>71.9</td>
</tr>
<tr>
<td>6MV Photon</td>
<td>$2.5 \times 10^8$</td>
<td>VMAT HN patient</td>
<td>N/A</td>
<td>36.7</td>
<td>N/A</td>
</tr>
<tr>
<td>6MV Photon</td>
<td>$2.5 \times 10^8$</td>
<td>VMAT Prostate patient</td>
<td>N/A</td>
<td>39.6</td>
<td>N/A</td>
</tr>
<tr>
<td>6MV Photon</td>
<td>$2.5 \times 10^8$</td>
<td>IMRT HN patient</td>
<td>N/A</td>
<td>36.1</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**CPU:** Intel Xeon processor with 2.27GHz  
**GPU:** NVIDIA Tesla C2050
Finite-size Pencil Beam Model  Data-Parallel Task

Input Data( geometry, beam setup, etc)

• Calculate $A(t, \theta)$
• Select ROI

• Calculate $D_i(r)$

- Uses Thrust library for sorting
- Beams in loop (cpu), beamlets in loop (gpu), voxels in threads
## Patient Cases

Table 1. Tumor site, number of beams, and case dimension for 5 head-and-neck (H1-H5) cases and 5 lung (L1-L5) cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Tumor Site</th>
<th># of Beams</th>
<th># of Beamlets</th>
<th># of Voxels</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>Parotid</td>
<td>8 (non-coplanar)</td>
<td>7,264</td>
<td>128x128x72</td>
</tr>
<tr>
<td>H2</td>
<td>Hypopharynx</td>
<td>7 (non-coplanar)</td>
<td>4,429</td>
<td>128x128x72</td>
</tr>
<tr>
<td>H3</td>
<td>Nasal Cavity</td>
<td>8 (non-coplanar)</td>
<td>3,381</td>
<td>128x128x72</td>
</tr>
<tr>
<td>H4</td>
<td>Parotid</td>
<td>5 (coplanar)</td>
<td>4,179</td>
<td>128x128x72</td>
</tr>
<tr>
<td>H5</td>
<td>Larynx</td>
<td>7 (non-coplanar)</td>
<td>10,369</td>
<td>128x128x72</td>
</tr>
<tr>
<td>L1</td>
<td>Left lung, low lobe (close to pleura)</td>
<td>6 (coplanar)</td>
<td>637</td>
<td>128x128x80</td>
</tr>
<tr>
<td>L2</td>
<td>Right lung, low lobe (paravertebral)</td>
<td>6 (coplanar)</td>
<td>1,720</td>
<td>128x128x103</td>
</tr>
<tr>
<td>L3</td>
<td>Left lung, upper lobe (close to pleura)</td>
<td>5 (coplanar)</td>
<td>921</td>
<td>128x128x80</td>
</tr>
<tr>
<td>L4</td>
<td>Right lung, upper lobe (close to heart)</td>
<td>7 (coplanar)</td>
<td>841</td>
<td>128x128x80</td>
</tr>
<tr>
<td>L5</td>
<td>Left lung (middle)</td>
<td>5 (coplanar)</td>
<td>686</td>
<td>128x128x80</td>
</tr>
</tbody>
</table>
Lung Cancer Patient Case (L4)

Monte Carlo  g-FSPB  DC(Density Correction)-FSPB

γ: g-FSPB  γ: g-DC-FSPB
Lung Cancer Patient Case (L4)
### Accuracy and Efficiency

Table 2. Gamma index evaluation results and dose calculation computation time for 10 testing cases using the g-DC-FSPB algorithm. The corresponding g-FSPB results are given in parenthesis for comparison purpose.

<table>
<thead>
<tr>
<th>Case #</th>
<th>$\gamma_{\text{max}}$</th>
<th>$\gamma_{50}^{\text{avg}}$</th>
<th>$P_{50}$</th>
<th>$T_{\text{tr}}$ (sec)</th>
<th>$T_{\text{gpu}}$ (sec)</th>
<th>$T_{\text{tot}}$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>2.12 (2.16)</td>
<td>0.30 (0.31)</td>
<td>97.53% (97.32%)</td>
<td>0.20</td>
<td>0.64 (0.55)</td>
<td>0.84 (0.75)</td>
</tr>
<tr>
<td>H2</td>
<td>3.44 (4.11)</td>
<td>0.28 (0.28)</td>
<td>97.80% (97.01%)</td>
<td>0.20</td>
<td>0.40 (0.35)</td>
<td>0.60 (0.55)</td>
</tr>
<tr>
<td>H3</td>
<td>2.27 (2.36)</td>
<td>0.46 (0.52)</td>
<td>92.29% (86.39%)</td>
<td>0.20</td>
<td>0.38 (0.34)</td>
<td>0.58 (0.54)</td>
</tr>
<tr>
<td>H4</td>
<td>3.08 (3.11)</td>
<td>0.61 (0.63)</td>
<td>82.96% (81.56%)</td>
<td>0.19</td>
<td>0.35 (0.32)</td>
<td>0.54 (0.51)</td>
</tr>
<tr>
<td>H5</td>
<td>3.33 (3.37)</td>
<td>0.61 (0.61)</td>
<td>86.19% (86.09%)</td>
<td>0.20</td>
<td>1.31 (1.10)</td>
<td>1.51 (1.30)</td>
</tr>
<tr>
<td>L1</td>
<td>1.53 (1.92)</td>
<td>0.24 (0.45)</td>
<td>99.35% (94.81%)</td>
<td>0.21</td>
<td>0.22 (0.20)</td>
<td>0.43 (0.41)</td>
</tr>
<tr>
<td>L2</td>
<td>2.35 (3.30)</td>
<td>0.36 (0.71)</td>
<td>96.64% (76.38%)</td>
<td>0.22</td>
<td>0.40 (0.36)</td>
<td>0.62 (0.58)</td>
</tr>
<tr>
<td>L3</td>
<td>1.68 (3.07)</td>
<td>0.32 (0.75)</td>
<td>99.16% (76.60%)</td>
<td>0.21</td>
<td>0.30 (0.25)</td>
<td>0.51 (0.46)</td>
</tr>
<tr>
<td>L4</td>
<td>2.70 (4.59)</td>
<td>0.63 (1.53)</td>
<td>81.33% (28.55%)</td>
<td>0.18</td>
<td>0.25 (0.23)</td>
<td>0.43 (0.41)</td>
</tr>
<tr>
<td>L5</td>
<td>2.19 (4.34)</td>
<td>0.49 (1.13)</td>
<td>90.24% (57.03%)</td>
<td>0.21</td>
<td>0.33 (0.29)</td>
<td>0.54 (0.50)</td>
</tr>
</tbody>
</table>
GPU-based Treatment Planning @ UCSD

• Dose calculation
  – gFSPB: finite size pencil beam model
  – gDPM: DPM MC code to GPU

• Plan optimization
  – gFMO: fluence map optimization
  – gDAO: direct aperture optimization
  – gVMAT: optimization for volumetric modulated arc therapy
DAO Model

- Optimize w.r.t both aperture and intensity

VMAT Model

- Optimize w.r.t both aperture and intensity
- One aperture is at one beam angle
- Aperture shapes at neighboring angles satisfy MLC mechanical constraints
- Smoothness of intensity changes between neighboring angles included in the objective function.
FMO model and flow chart

$$\min \sum_{j \in V} F_j (z_j^l)$$

Subject to

Voxel dose
$$z_j^l = \sum_{i \in N} D_{ij}^l x_i \quad j \in V$$

$$x_i \geq 0 \quad i \in N$$

Where

$$F_{s-} (z) = \sum_{j \in V_s} (\max(0, z_j^l - z_j^l))^2 \quad s \in T$$

$$F_{s+} (z) = \sum_{j \in V_s} (\max(0, z_j^l - z_j^l))^2 \quad s \in S$$
Results for GPU-based FMO Algorithm

<table>
<thead>
<tr>
<th>Case</th>
<th>beamlet size (mm²)</th>
<th># beamlets</th>
<th>voxel size (mm³)</th>
<th># voxels (×10⁴)</th>
<th># non-zero $D_{ij}$'s (×10⁶)</th>
<th>GPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 × 10</td>
<td>2,055</td>
<td>4 × 4 × 4</td>
<td>3.6</td>
<td>3.1</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>5 × 5</td>
<td>6,433</td>
<td>4 × 4 × 4</td>
<td>3.6</td>
<td>10.6</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5 × 5</td>
<td>6,433</td>
<td>2.5 × 2.5 × 2.5</td>
<td>14.0</td>
<td>43.3</td>
<td>2.8</td>
</tr>
</tbody>
</table>

~40x speedup compared to an Intel Xeon 2.27 GHz CPU
~0.5 sec for re-optimizing a 9-field prostate IMRT plan
Prostate Cancer Case

- VMAT
- IMRT
Summary – GPU-based Treatment Planning

• We have developed GPU-based computational tools for real-time treatment planning
  – Conventional problem size have been well fit to one GPU hardware

• For a typical prostate case
  – The dose calculation takes less than 1 second with FSPB with 3D density correction, less than 40 seconds with Monte Carlo
  – The plan optimization takes less than 1 second with FMO, 2 seconds with DAO (intensity, aperture), and 30 seconds with VMAT

➢ Next step

❑ Faster and Finer → algorithm improvement, larger problems, multiple GPUs
❑ Software integration → Analysis/Data integration
❑ Clinical implementation and evaluation