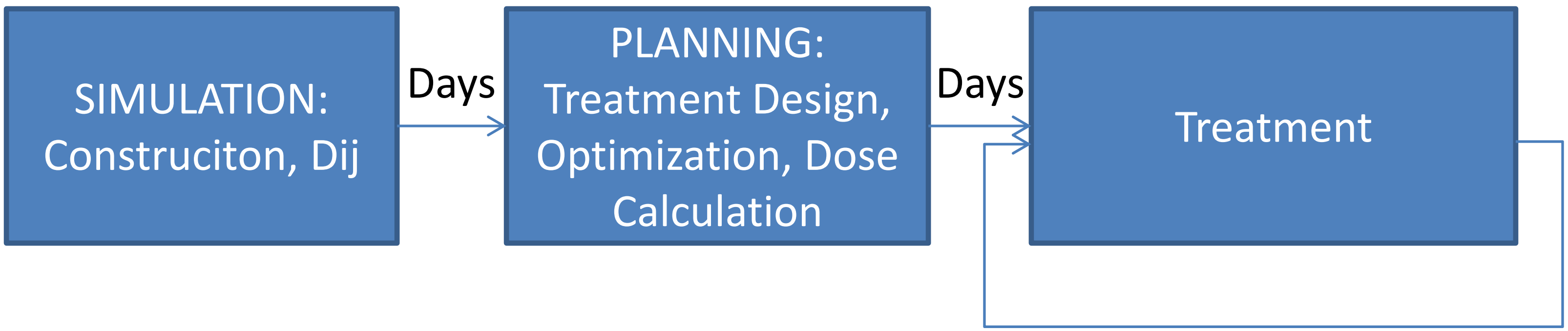
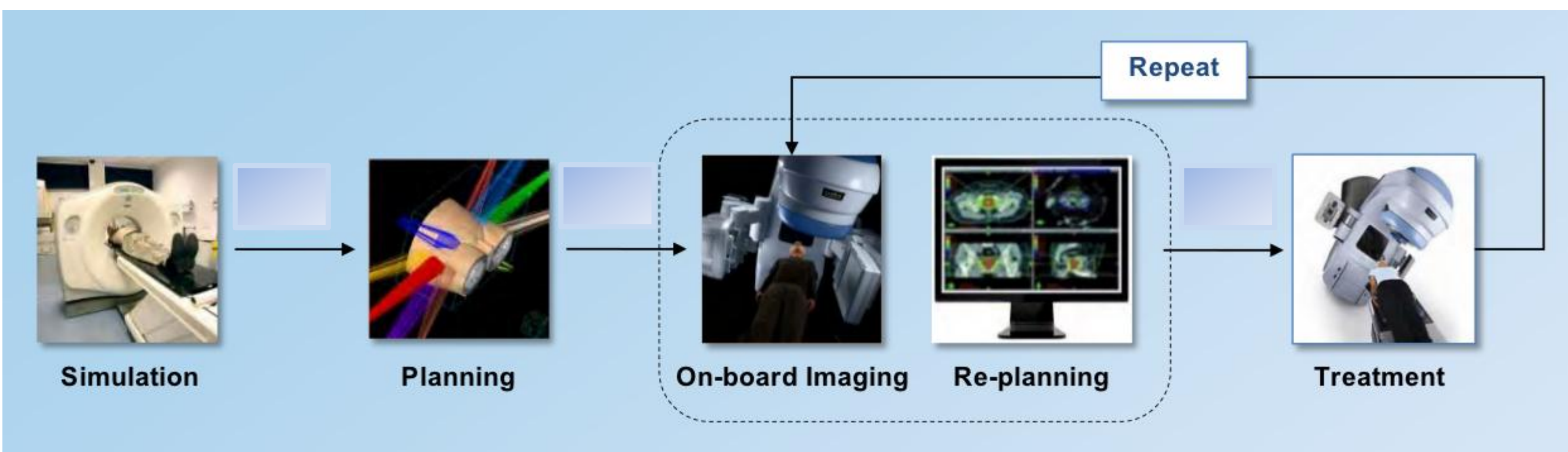
GPU applications in Cancer Radiation Therapy at UCSD

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

Conventional Radiotherapy

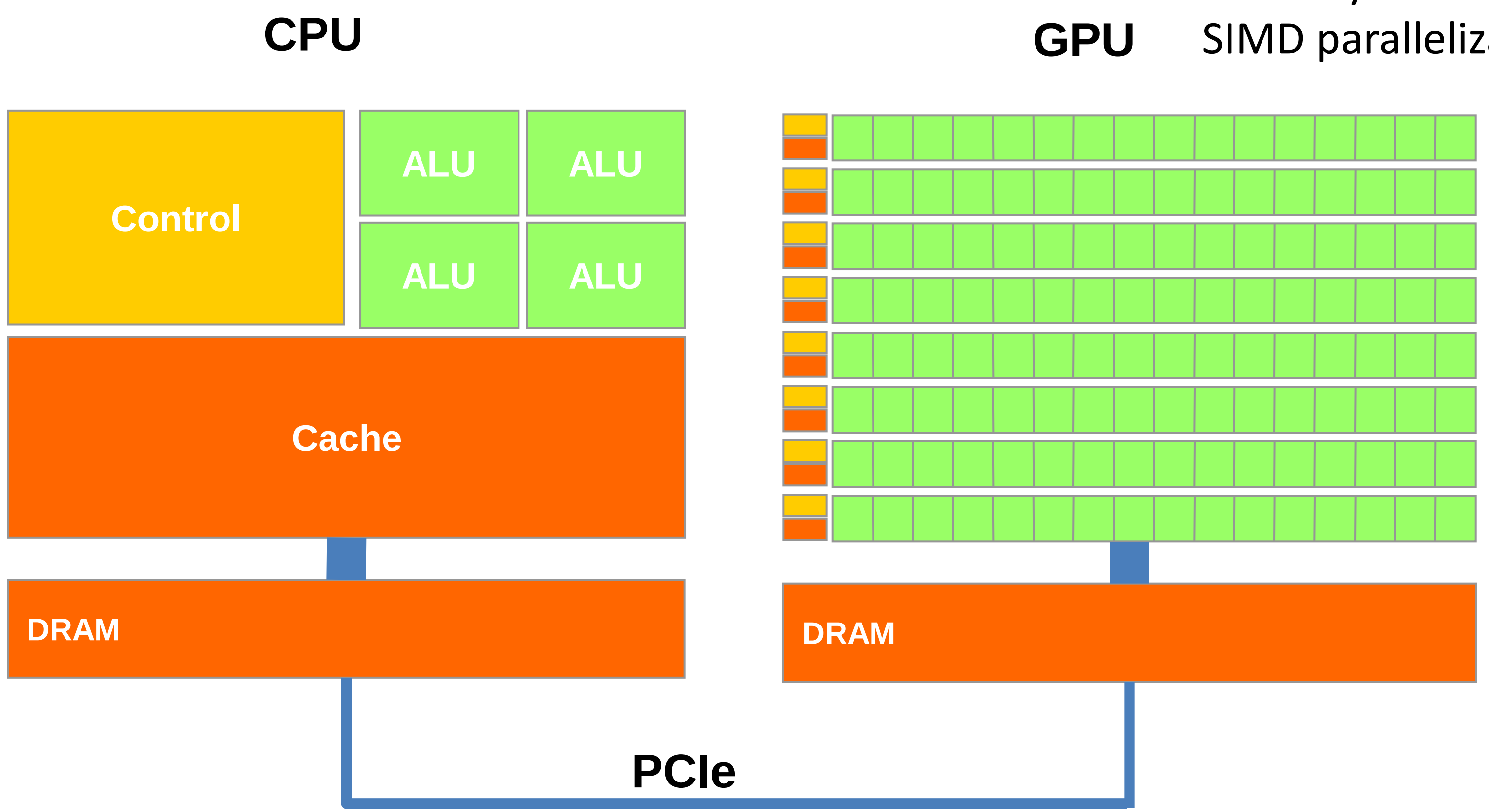


Online ART (Adaptive Radiation) We Want

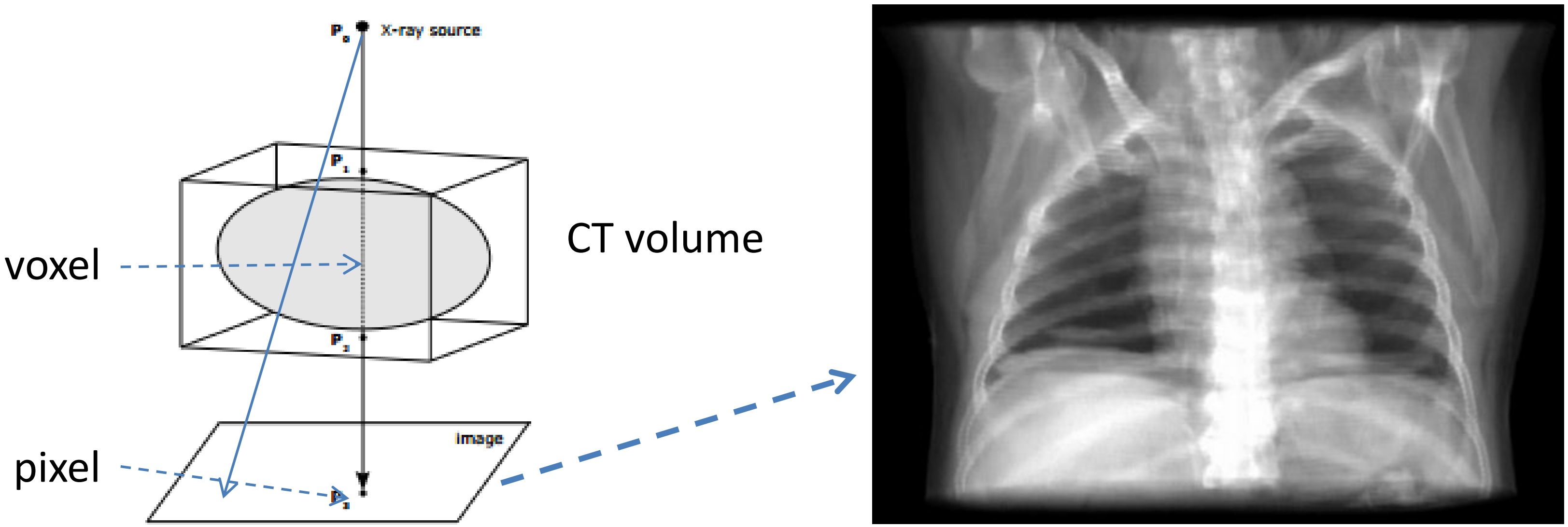


CPU vs GPU

Memory
SIMD parallelization



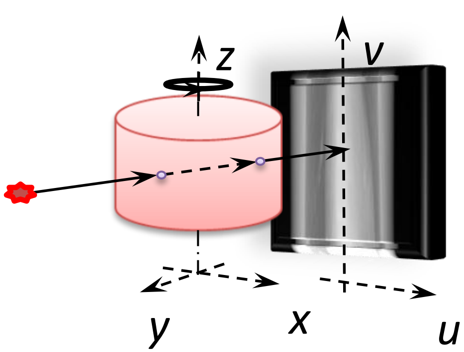
Simple Radiation Application in GPU: DRR (Digitally Reconstructed Radiographs)

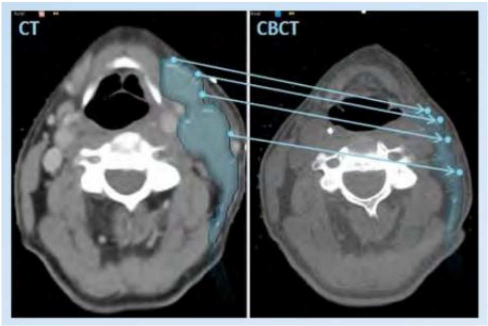


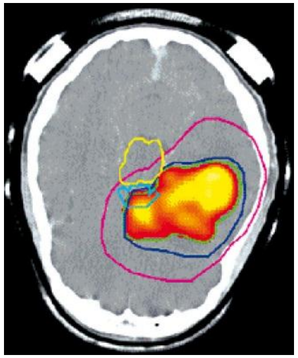
- Image pixel calculated in each thread
- CT volume in Texture memory
- Few seconds in CPU becomes few 1/10s seconds in GPU

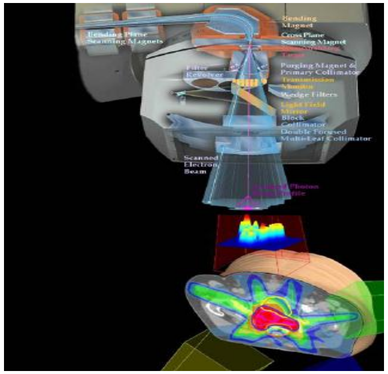
Online Re-planning Process Needs..

- CBCT (Cone-Beam CT) Reconstruction
- Deformable Image Registration
- Re-contours
- Dose Calculation
- Plan Re-Optimization



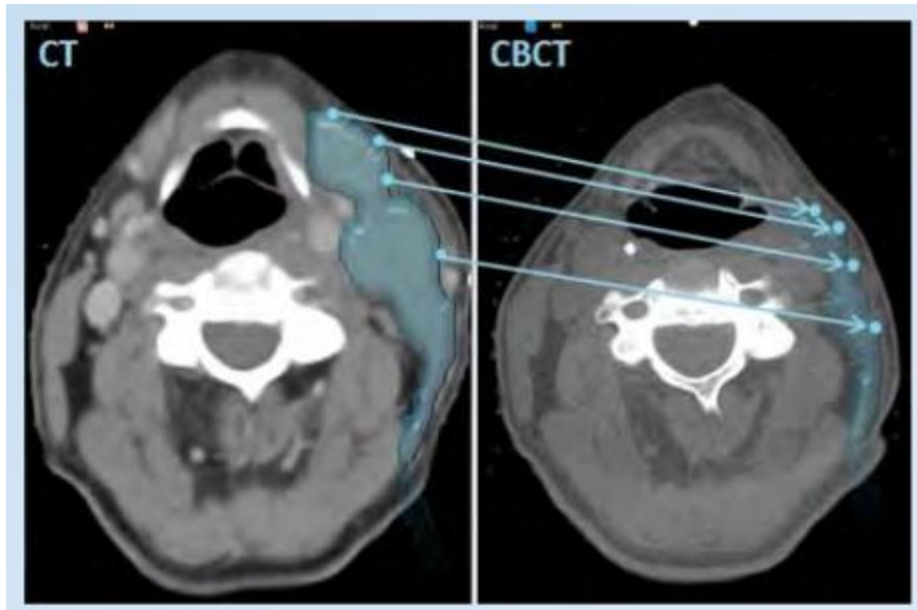






Deformable Image Registration with 'Demons'

- 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)



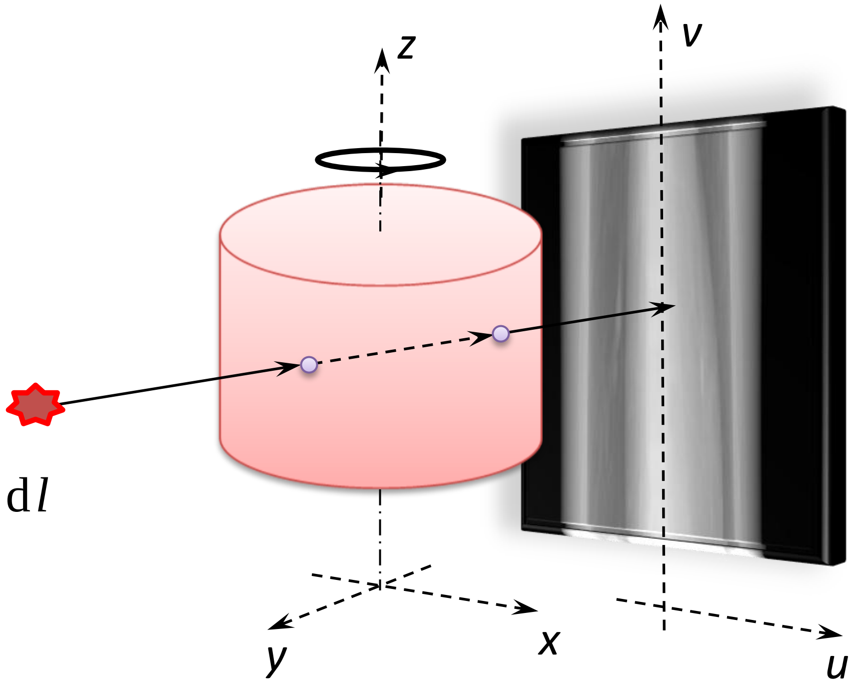
$$\vec{d}(\boldsymbol{x}) = \frac{2[I_0(x) - I_1(x)][\vec{\nabla} I_0(x) + \vec{\nabla} I_1(x)]}{\|\vec{\nabla} I_0(x) + \vec{\nabla} I_1(x)\|^2 + [I_0(x) - I_1(x)]^2/K^2},$$

Cone Beam CT

- CBCT reconstruction problem

$$I(u,v) = I_0 e^{-\int_S^{(u,v)} f(x,y,z)\mathrm{d}l}$$

$$g(u,v) = -\log\left(\frac{I}{I_0}\right) = \int_S^{(u,v)} f(x,y,z)\mathrm{d}l$$



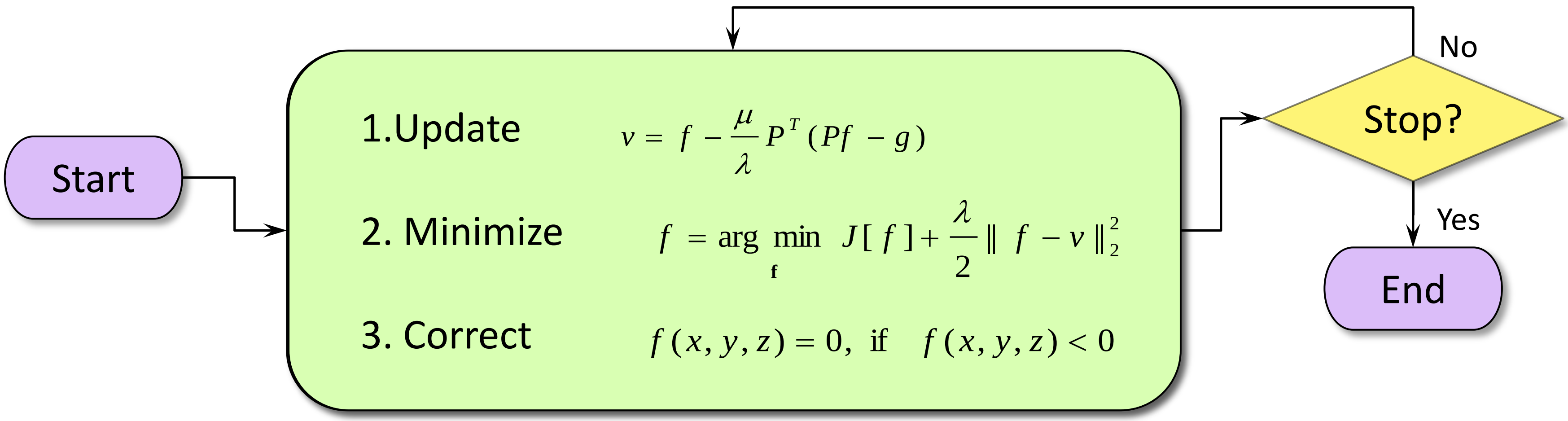
- Mathematical problem

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

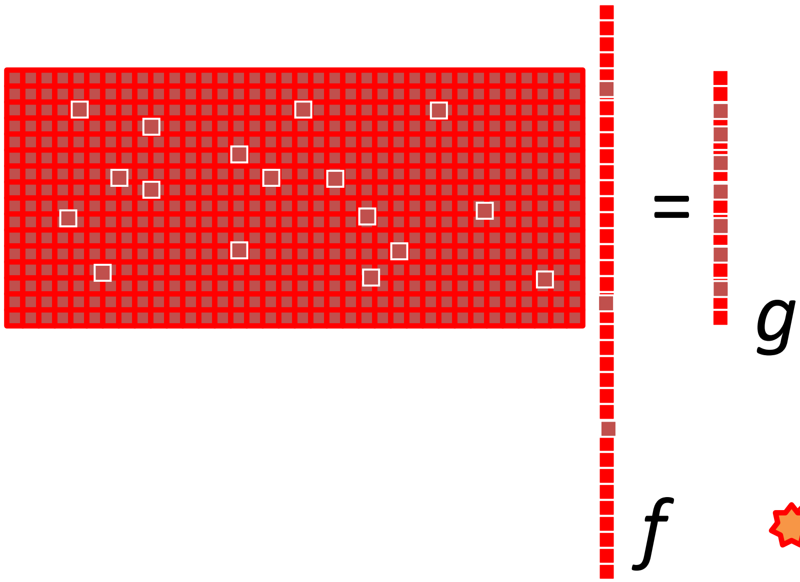
Algorithm

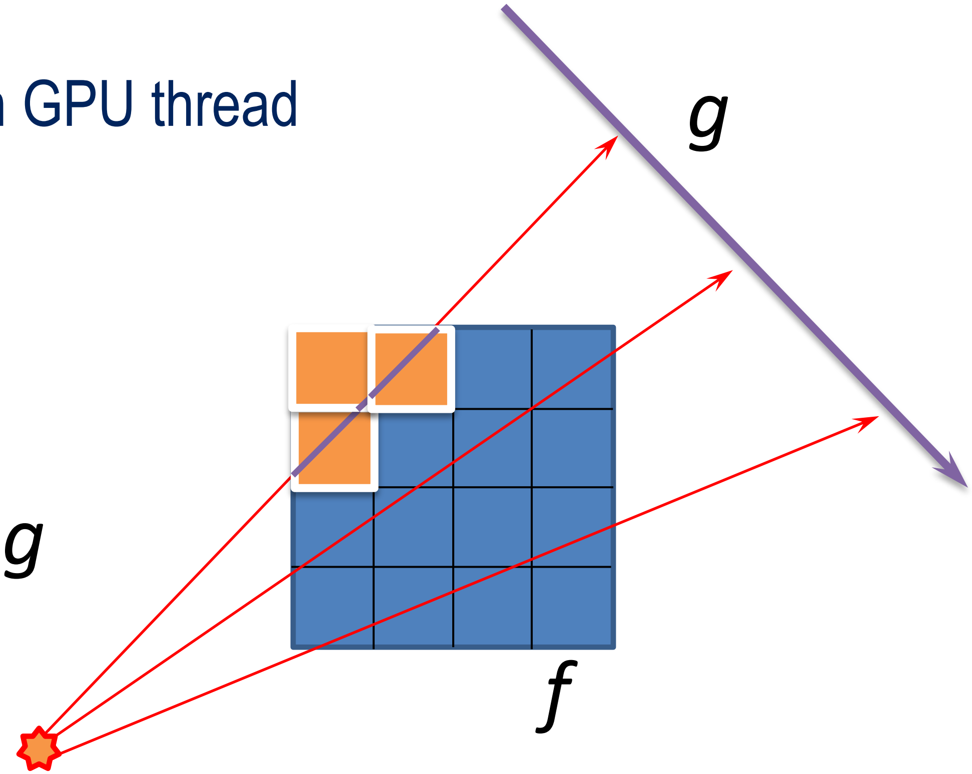
- Optimization procedure



Computation of $g = Pf$

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

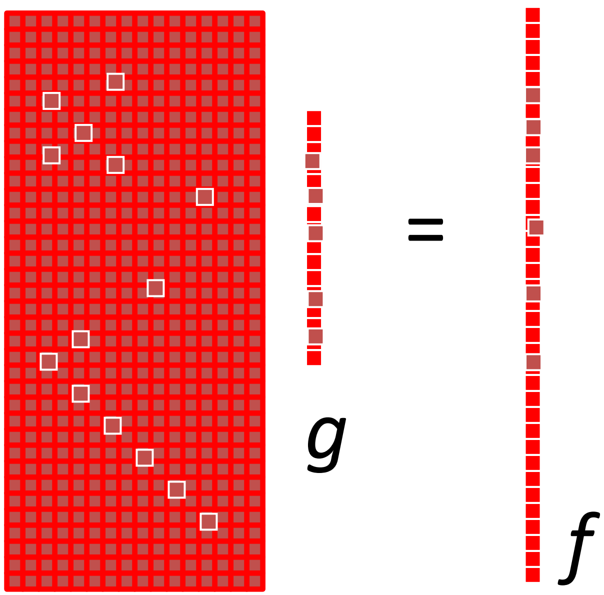


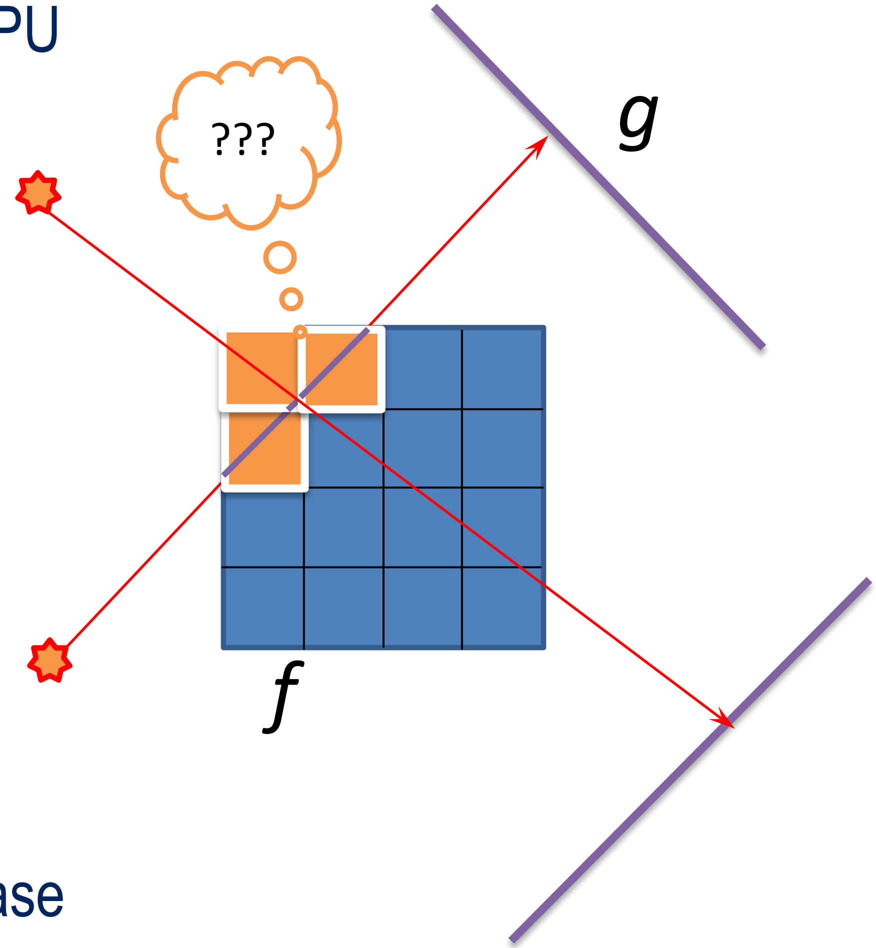


- Multiple threads run in parallel ~100 times faster than CPU implementation

Computation of $f = P^T g$

- Backward calculation
- Ray tracing algorithm on each GPU thread





- Multiple threads run in parallel, however…Slow!!
- *T = 46 min* for a typical clinical case

GPU-friendly Algorithm for P^T

- Consider $g = Pf$ in a continuum case

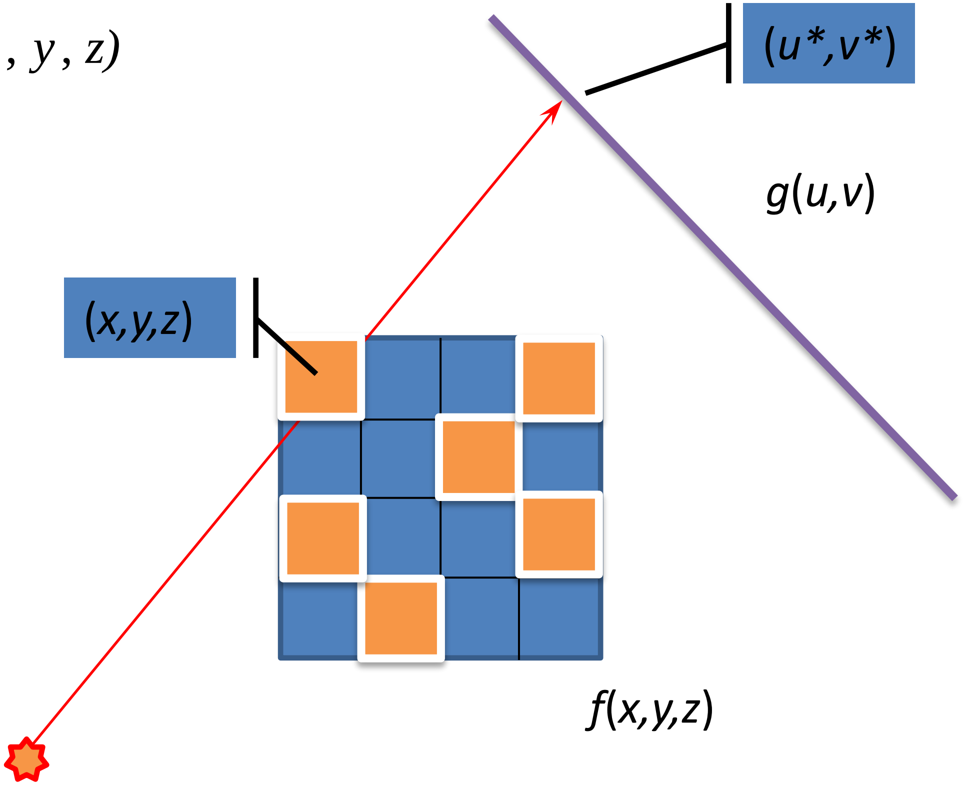
$$P[f(x,y,z)](u,v) = \int_L \mathrm{d}l \; f(x,y,z)$$

- 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 ~20

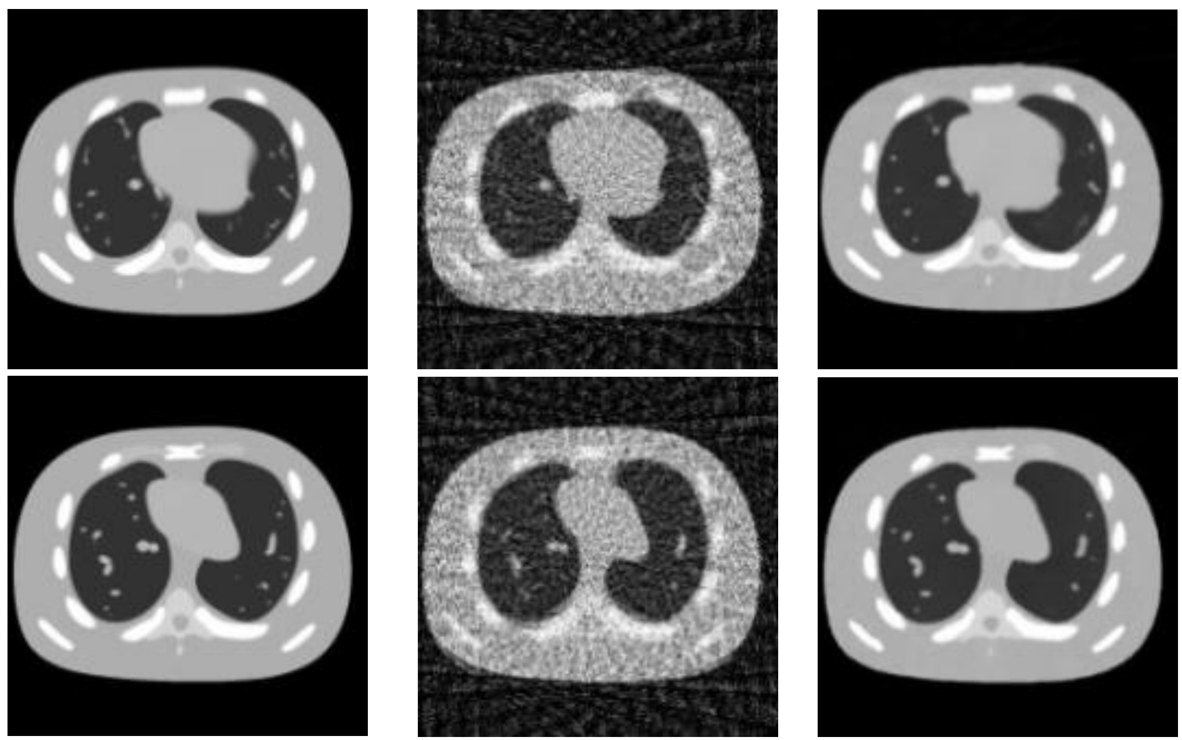
Results

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

Ground Truth FBP TNLM

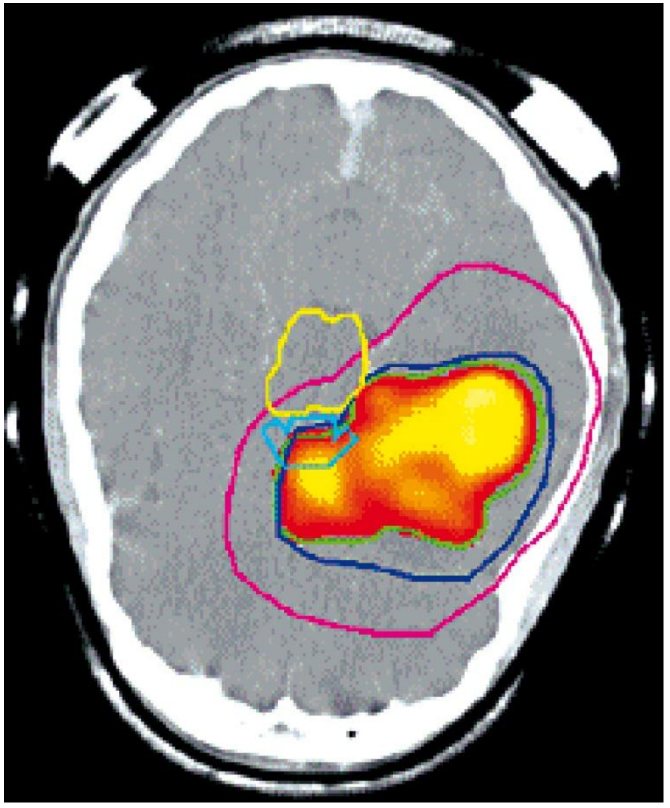
40%

70%



Online Re-planning Process Needs..

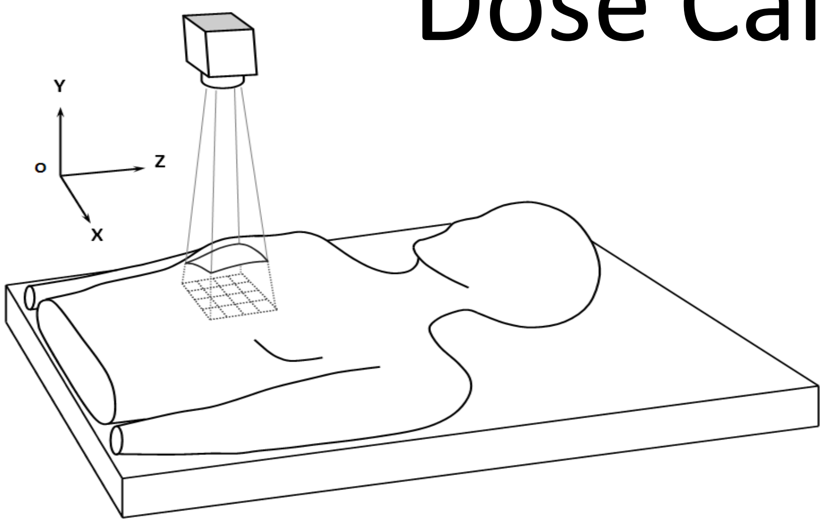
- ~~CBCT Reconstruction~~
- ~~Deformable Image Registration~~
- Re-contours
- Dose Calculation
- Plan Re-Optimization

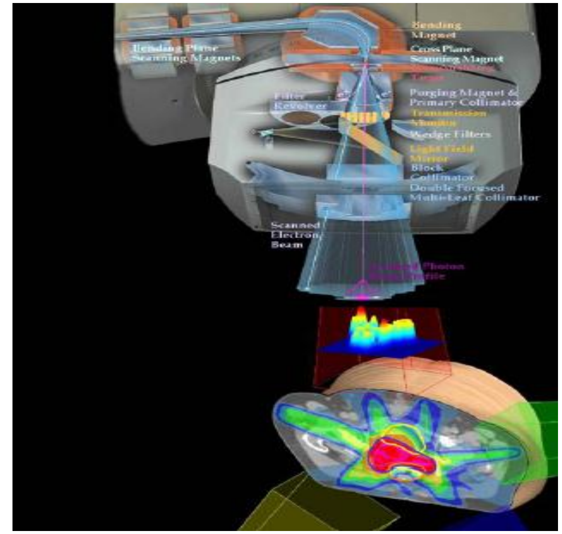


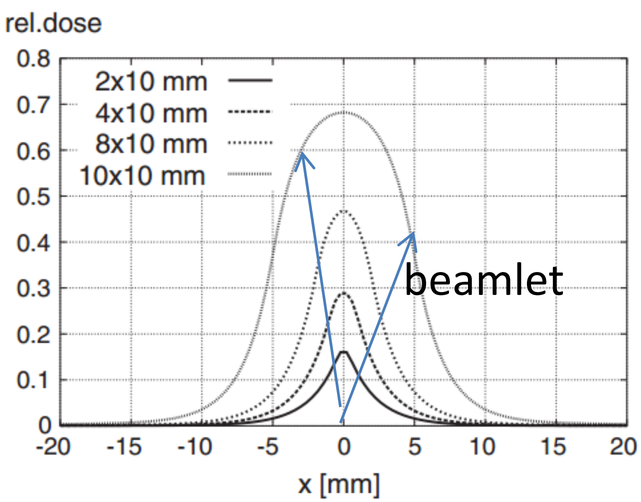
GPU-based Treatment Planning @ UCSD

- 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~~

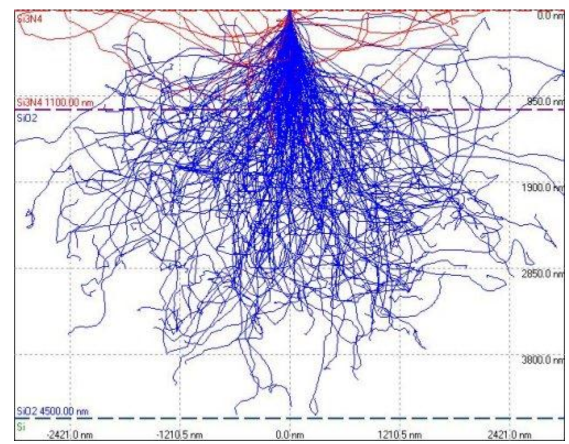
Dose Calculation





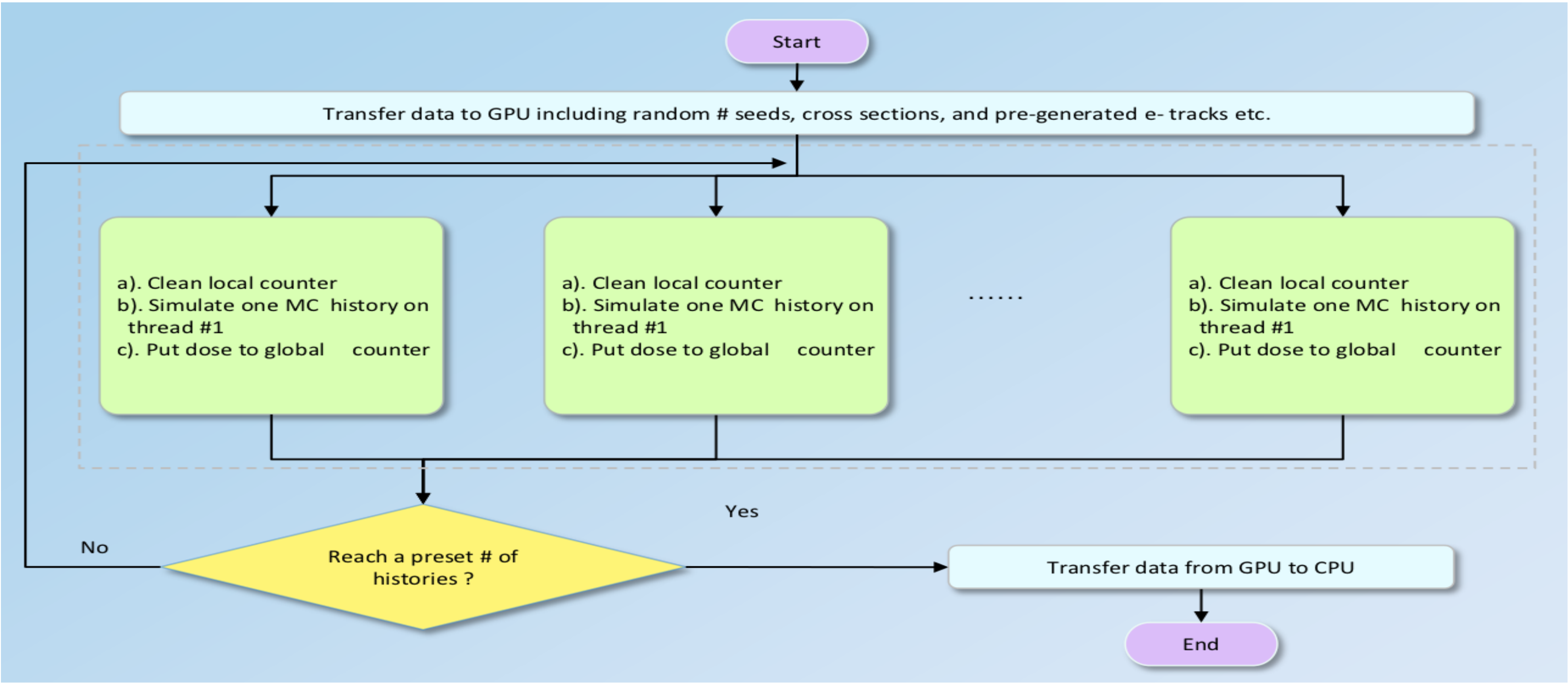


FSPB Beam Cross-profiles



Monte Carlo histories

Monte Carlo Dose Calculation on GPU



Results (photon point source, 6MV on head)

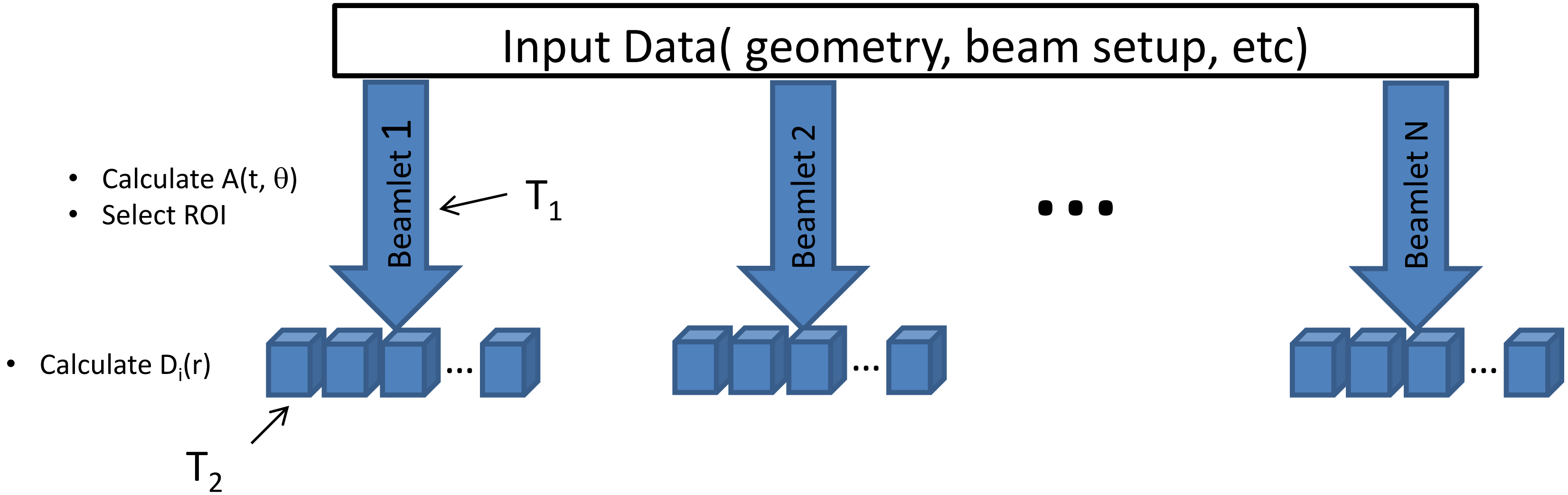
Execution time T, and speed-up factor T_{CPU}/T_{GPU} for four different testing cases.

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

CPU: Intel Xeon processor with 2.27GHz

GPU: NVIDIA Tesla C2050

Finite-size Pencil Beam Model Data-Parallel Task



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

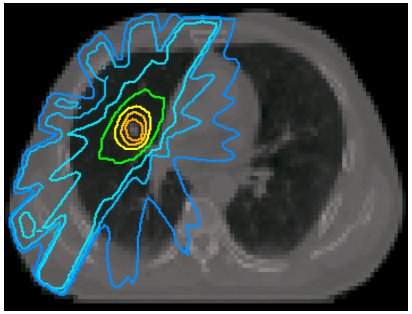
Patient Cases

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

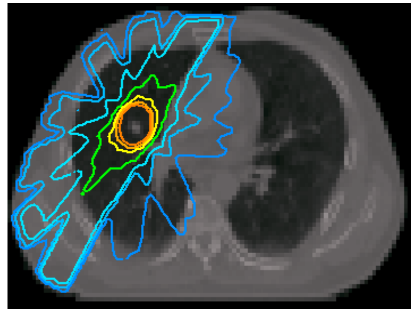
Case	Tumor Site	# of Beams	# of Beamlets	# of Voxels
H1	Parotid	8 (non-coplanar)	7,264	128×128x72
H2	Hypopharynx	7 (non-coplanar)	4,429	128x128x72
H3	Nasal Cavity	8 (non-coplanar)	3,381	128x128x72
H4	Parotid	5 (coplanar)	4,179	128x128x72
H5	Larynx	7 (non-coplanar)	10,369	128x128x72
L1	Left lung, low lobe(close to pleura)	6 (coplanar)	637	128x128x80
L2	Right lung, low lobe (paravertebral)	6 (coplanar)	1,720	128x128x103
L3	Left lung, upper lobe (close to pleura)	5 (coplanar)	921	128x128x80
L4	Right lung, upper lobe (close to heart)	7 (coplanar)	841	128x128x80
L5	Left lung (middle)	5 (coplanar)	686	128x128x80

Lung Cancer Patient Case (L4)

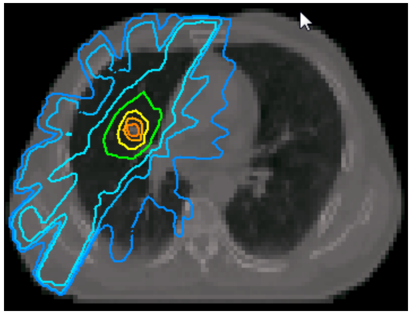
Monte Carlo

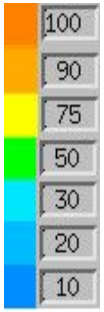


g-FSPB

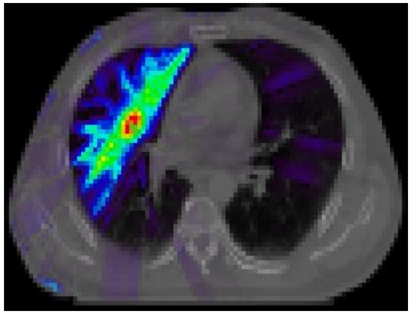


DC(Density Correction)-FSPB

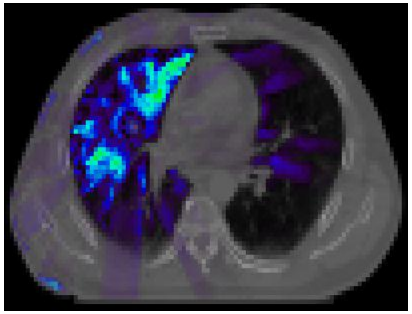


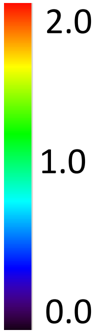


γ: g-FSPB

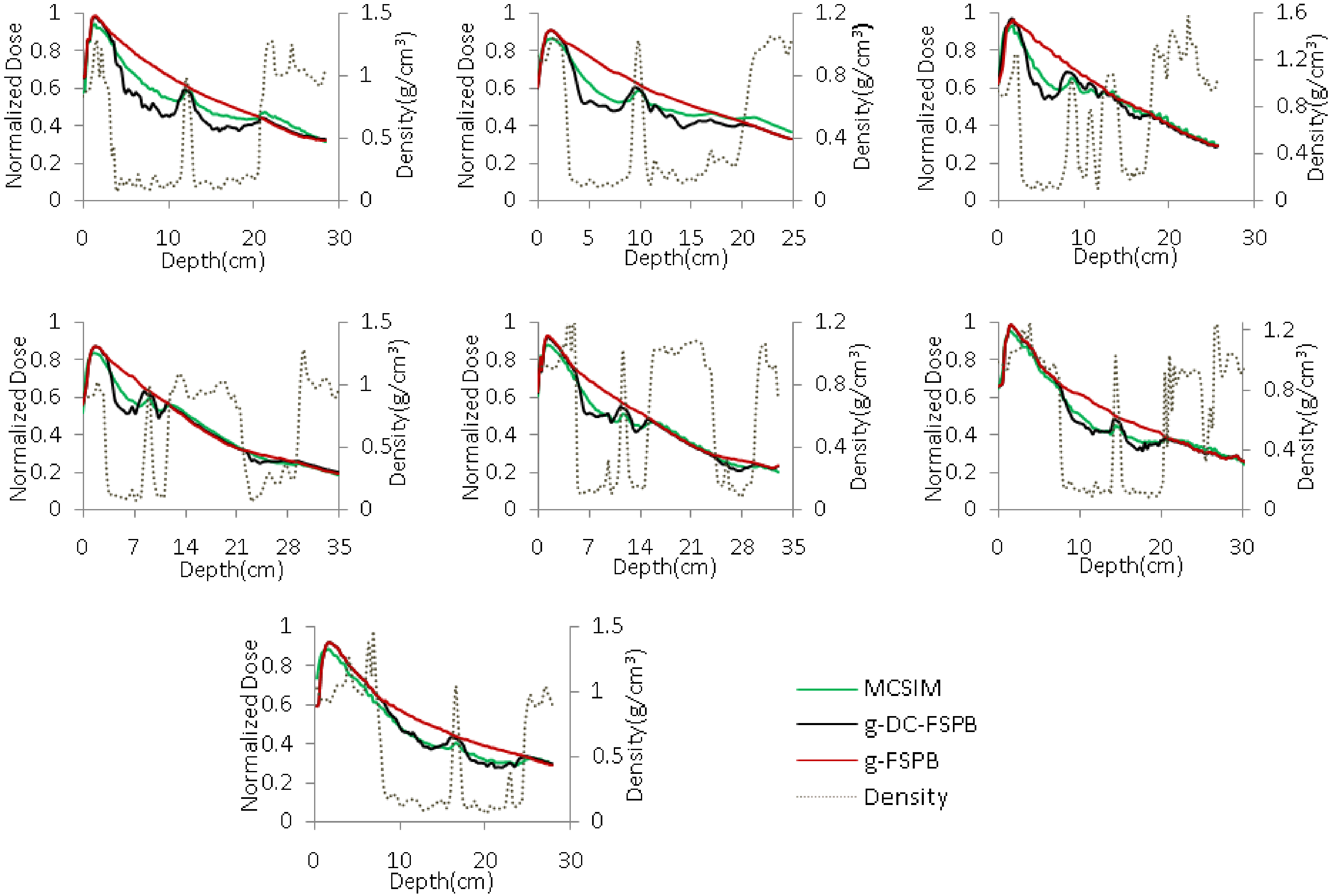


γ: g-DC-FSPB





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.

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

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

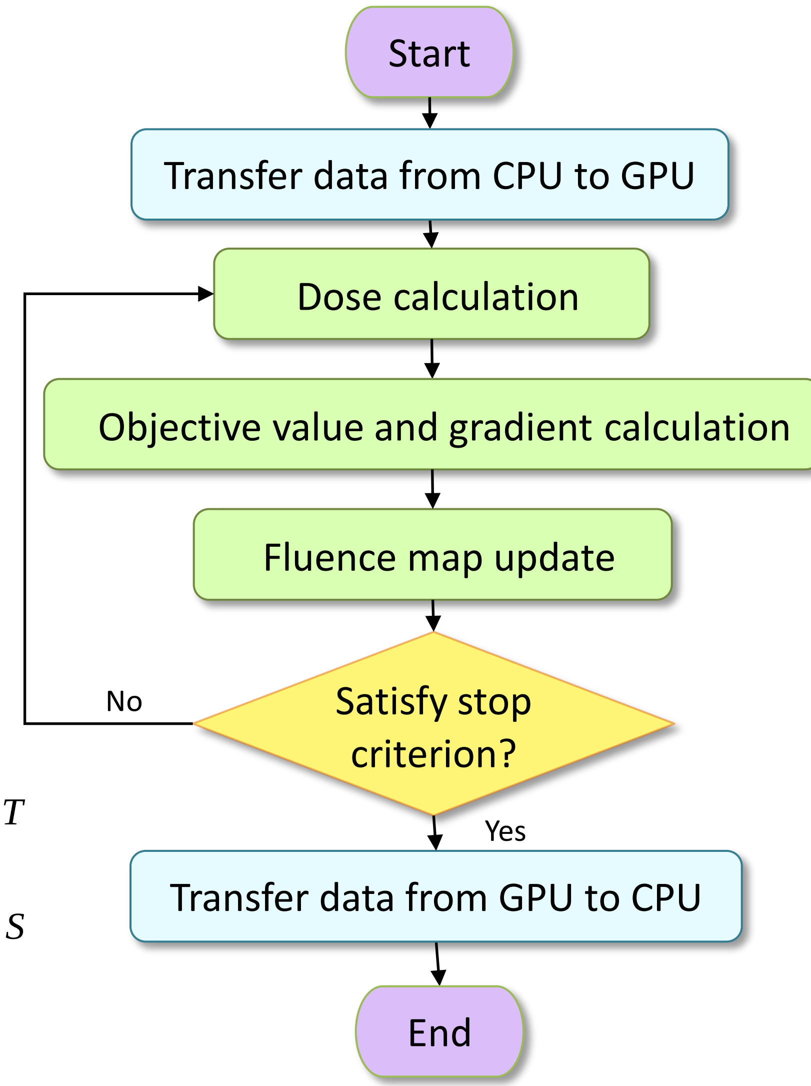
Voxel dose $$z_j^l = \sum_{i \in N} D_{ij}^l x_i \qquad j \in V$$

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

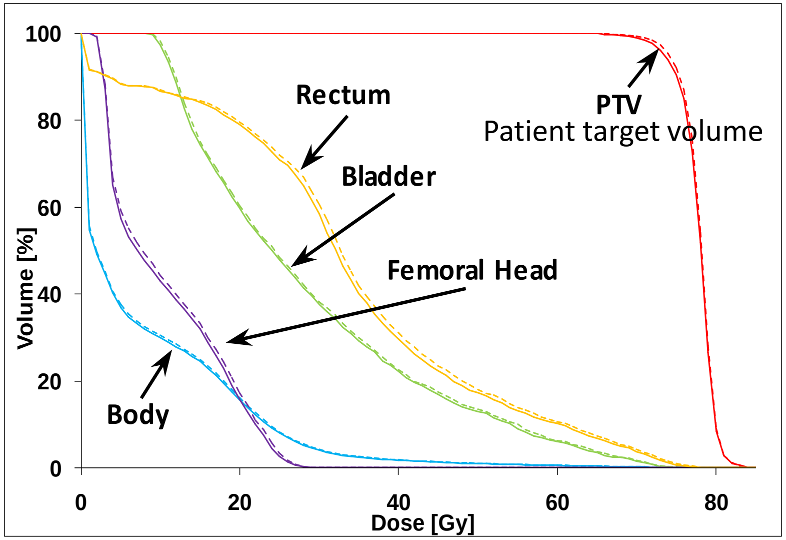
Where

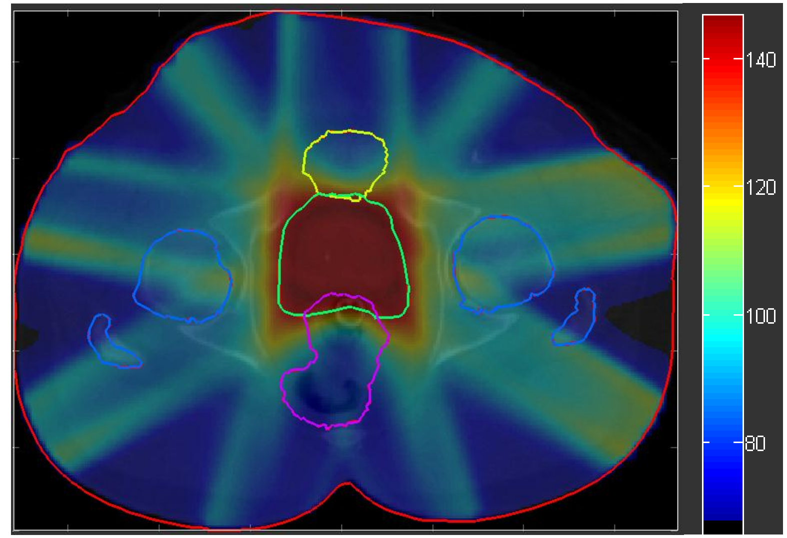
$$F_{s-}(\mathbf{z}) = \sum_{j \in V_s} (\max(0, z_j - z_j^l))^2 \qquad s \in T$$

$$F_{s+}(\mathbf{z}) = \sum_{j \in V_s} (\max(0, z_j^l - z_j))^2 \qquad s \in S$$



Results for GPU-based FMO Algorithm





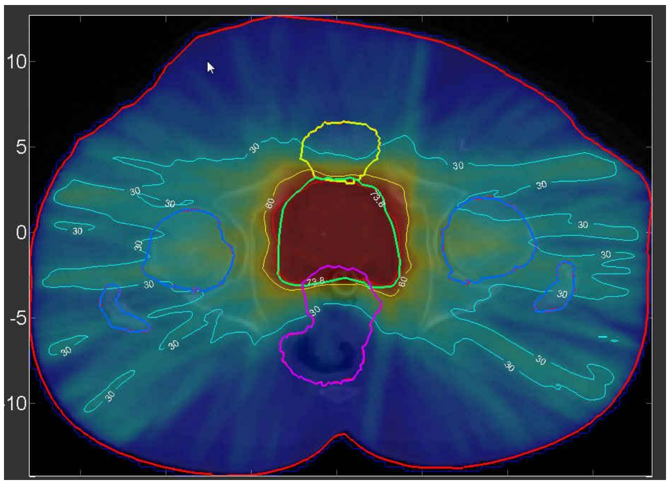
# Case	beamlet size (mm²)	# beamlets	voxel size (mm³)	# voxels (×10⁴)	# non-zero D_{ij}'s (×10⁶)	GPU time (s)
1	10×10	2,055	4×4×4	3.6	3.1	0.2
2	5×5	6,433	4×4×4	3.6	10.6	0.5
3	5×5	6,433	2.5×2.5×2.5	14.0	43.3	2.8

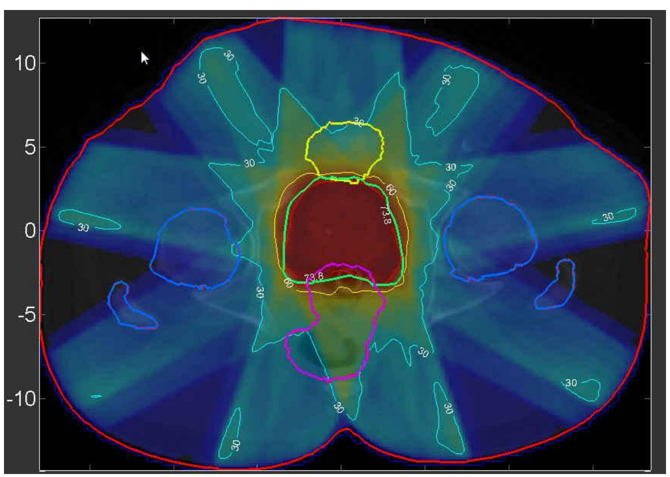
~40x speedup compared to an Intel Xeon 2.27 GHz CPU

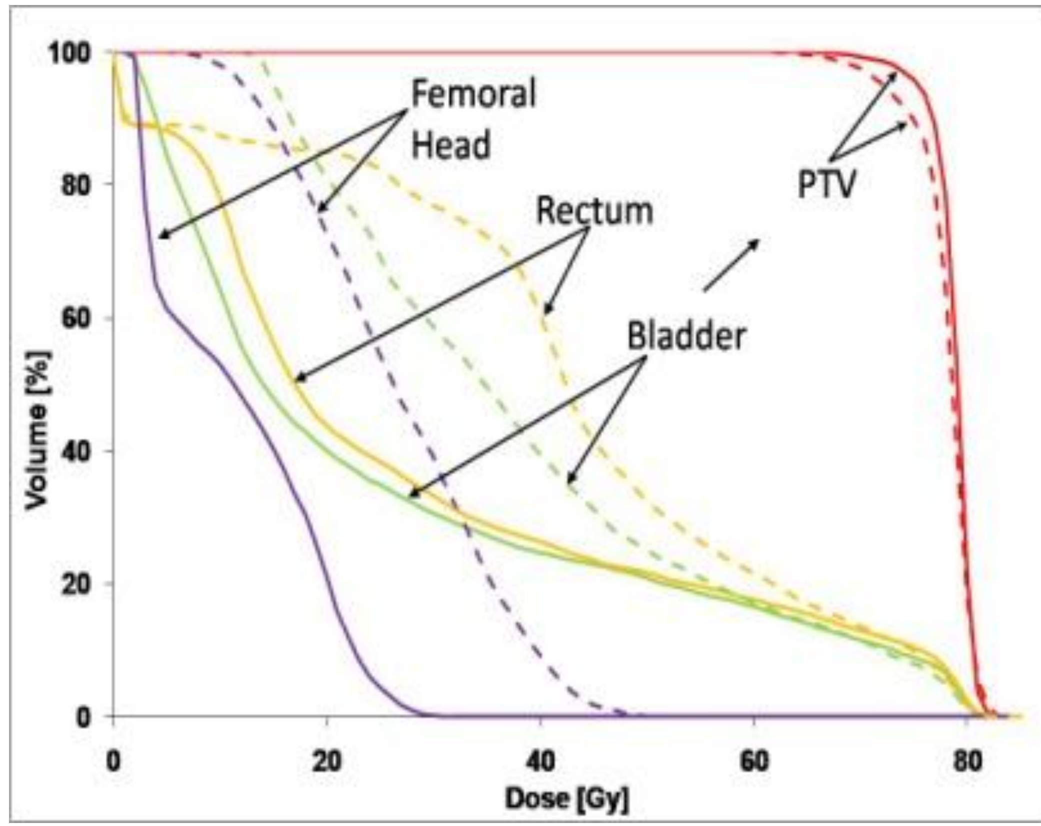
~0.5 sec for re-optimizing a 9-field prostate IMRT plan

Men et al *Phys Med Biol* 54(21):6565-6573, *2009*

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