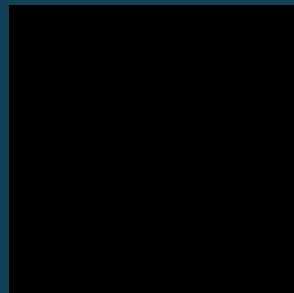


3D Motion Blur For PET Using GPGPU

Author: **Dóra Varnyú**

Supervised by: **László Szirmay-Kalos**

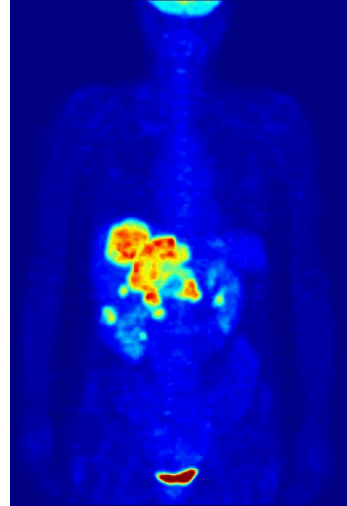
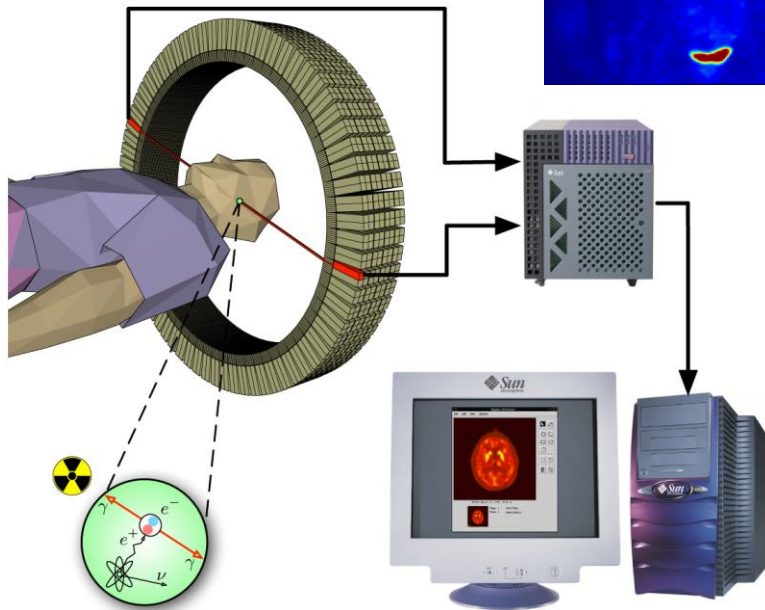
Budapest University of Technology, Hungary





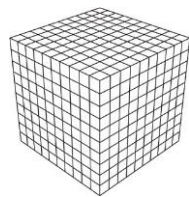
Positron Emission Tomography (PET)

- Cancer
- Coronary Artery Disease
- Alzheimer's disease
- Epilepsy





PET reconstruction

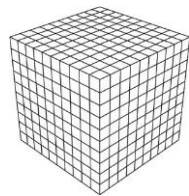


Homogeneous 3D voxels

Voxel activity: number of radioactive decays that occurred in the voxel

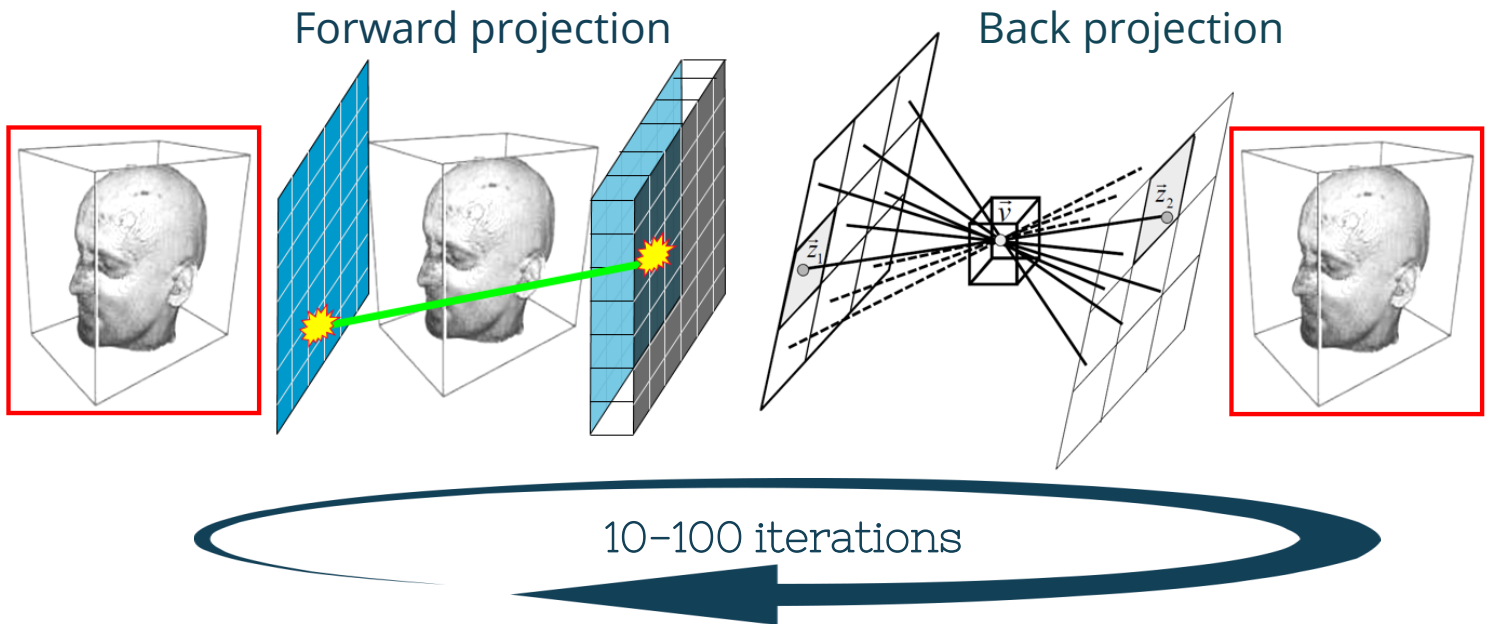


PET reconstruction



Homogeneous 3D voxels

Voxel activity: number of radioactive decays that occurred in the voxel





Motion Blur



Created by Diana Juncher, [CC BY 4.0](#)



Created by E01, [CC BY-SA 2.0](#)

With motion blur



Created by Niabot, [CC BY-SA 3.0](#)

Without motion blur



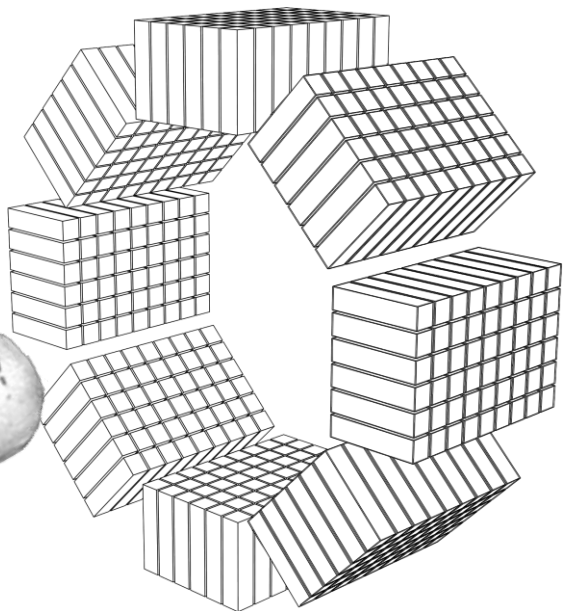


Overview 1/2

Measuring movement real-time using markers

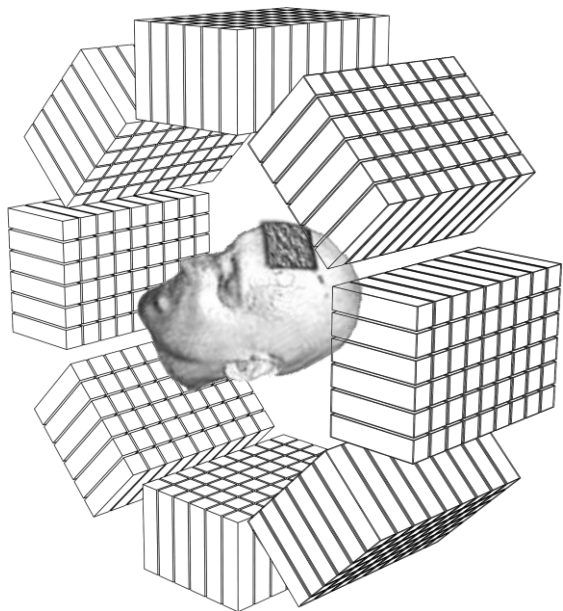
$\mathbf{v}(x, y, z)$ translation vector

$\mathbf{q}(x, y, z, w)$ quaternion for rotation





Overview 1/2



Measuring movement real-time using markers

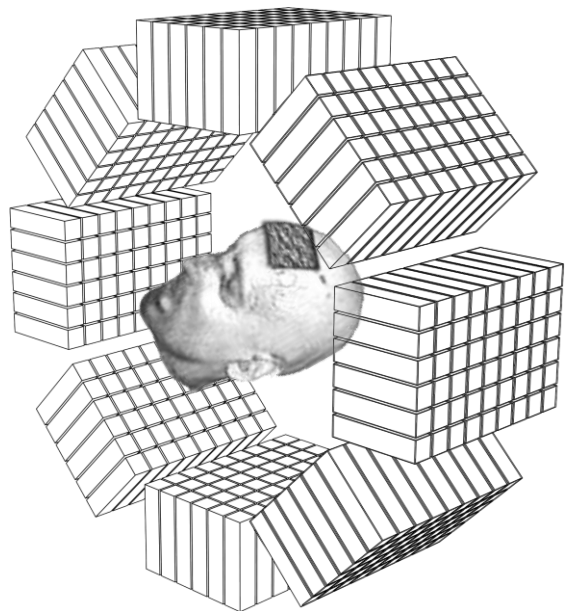
$v(x, y, z)$ translation vector

$q(x, y, z, w)$ quaternion for rotation



Overview 1/2

Operating on two voxel arrays:



Measuring movement real-time using markers

$v(x, y, z)$ translation vector

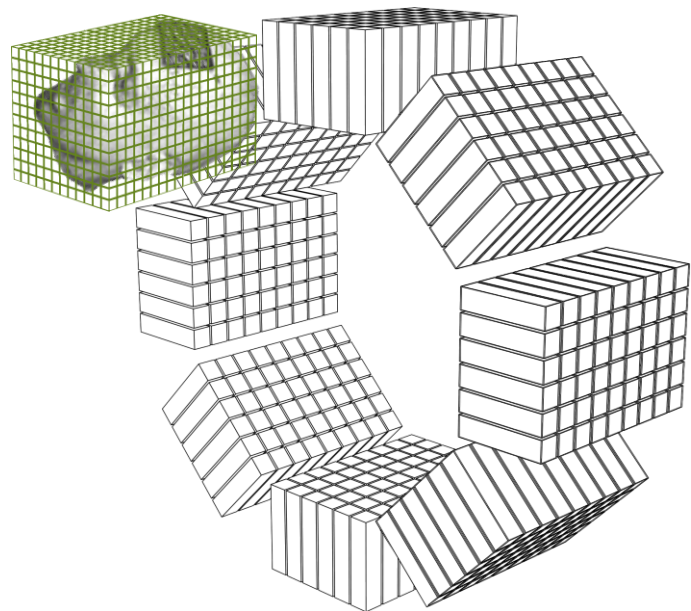
$q(x, y, z, w)$ quaternion for rotation



Overview 1/2

Operating on two voxel arrays:

1. Object-space array



Measuring movement real-time using markers

$v(x, y, z)$ translation vector

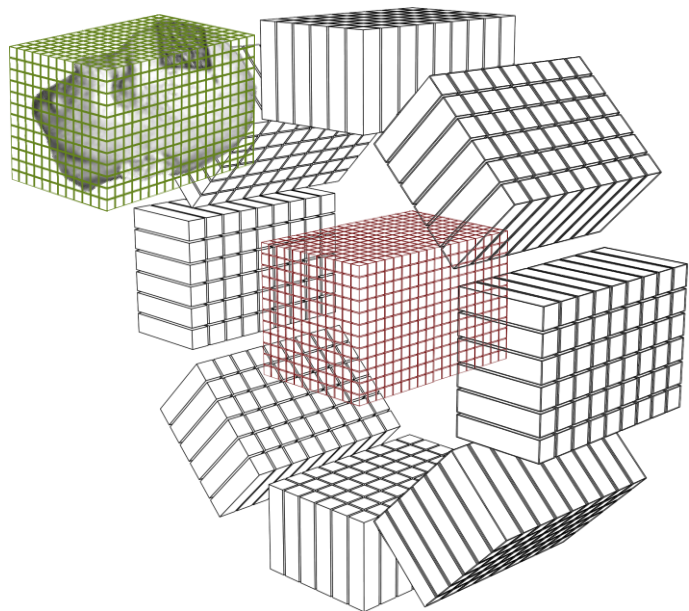
$q(x, y, z, w)$ quaternion for rotation



Overview 1/2

Operating on two voxel arrays:

1. Object-space array
2. Tomograph-space array



Measuring movement real-time using markers

$v(x, y, z)$ translation vector

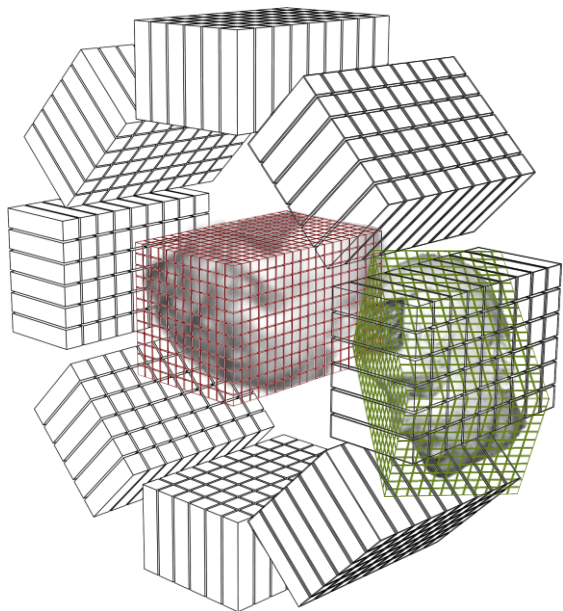
$q(x, y, z, w)$ quaternion for rotation



Overview 1/2

Operating on two voxel arrays:

1. Object-space array
2. Tomograph-space array



Measuring movement real-time using markers

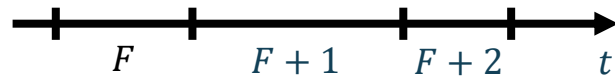
$v(x, y, z)$ translation vector

$q(x, y, z, w)$ quaternion for rotation

$\int$

Overview 2/2

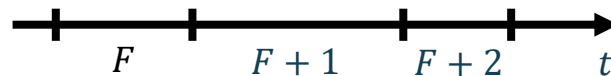
Time is divided into frames



∫

Overview 2/2

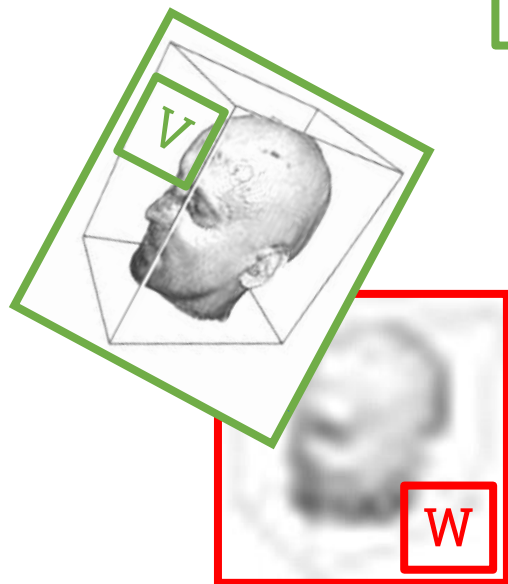
Time is divided into frames



Object-space voxel



Tomograph-space voxel



$x_{V,F}$: activity of voxel V in frame F

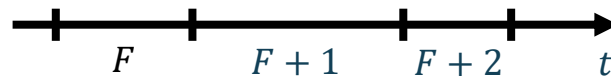
$o_{V,W,F}$: average overlap factor of voxels V and W in frame F ($o_{V,W,F} \in [0; 1]$, $\sum_W o_{V,W,F} = 1$)

$$x_{W,F} = \sum_V o_{V,W,F} \cdot x_{V,F}$$



Overview 2/2

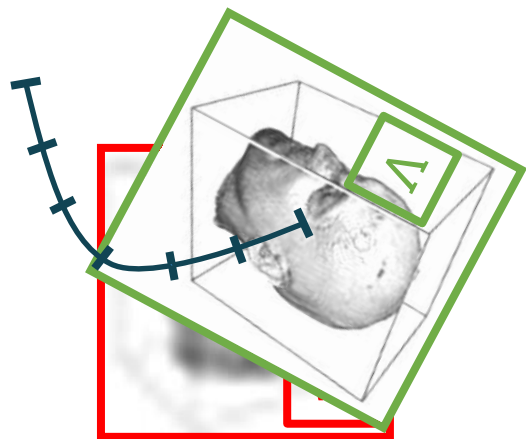
Time is divided into frames



Object-space voxel



Tomograph-space voxel



$x_{V,F}$: activity of voxel V in frame F

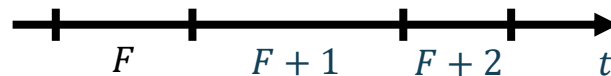
$o_{V,W,F}$: average overlap factor of voxels V and W in frame F ($o_{V,W,F} \in [0; 1]$, $\sum_W o_{V,W,F} = 1$)

$$x_{W,F} = \sum_V o_{V,W,F} \cdot x_{V,F}$$



Overview 2/2

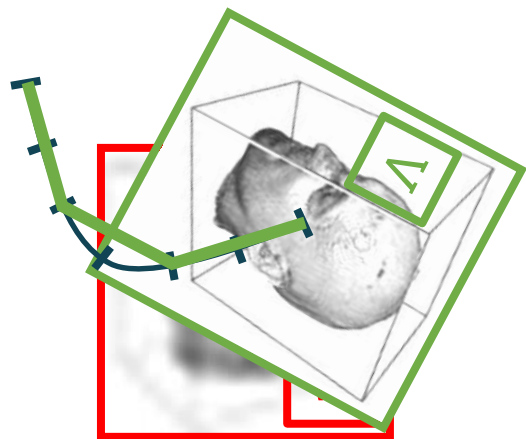
Time is divided into frames



Object-space voxel



Tomograph-space voxel

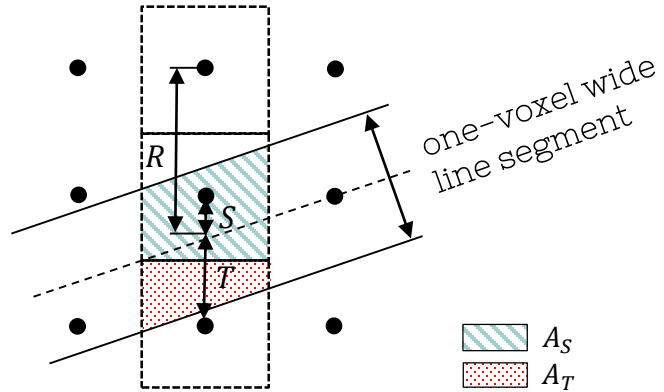


$x_{V,F}$: activity of voxel V in frame F

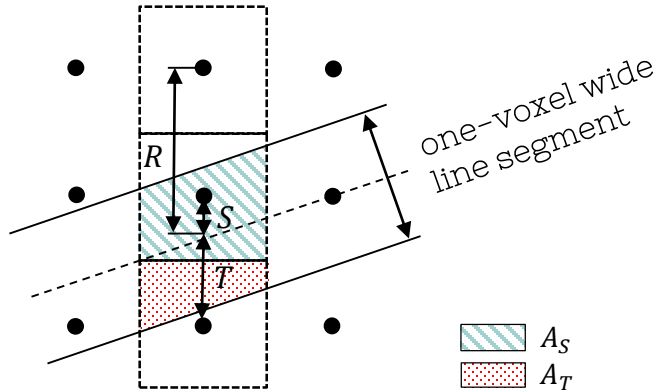
$o_{V,W,F}$: average overlap factor of voxels V and W in frame F ($o_{V,W,F} \in [0; 1]$, $\sum_W o_{V,W,F} = 1$)

$$x_{W,F} = \sum_V o_{V,W,F} \cdot x_{V,F}$$

Antialiased Bresenham Algorithm



Antialiased Bresenham Algorithm



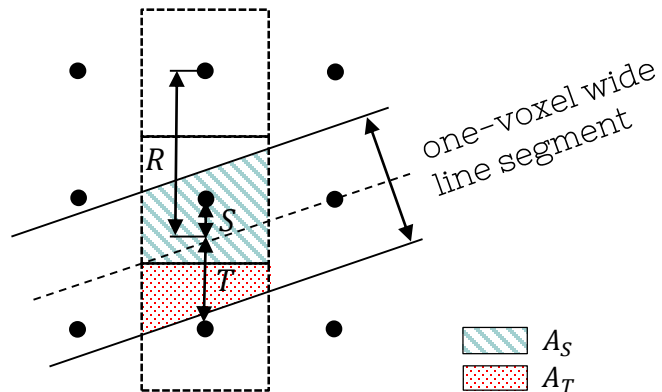
Areas of intersection:

$$A_S \approx (1 - S)$$

$$A_T \approx S$$

$$A_R \approx 0 \text{ (furthest voxel disregarded)}$$

Antialiased Bresenham Algorithm

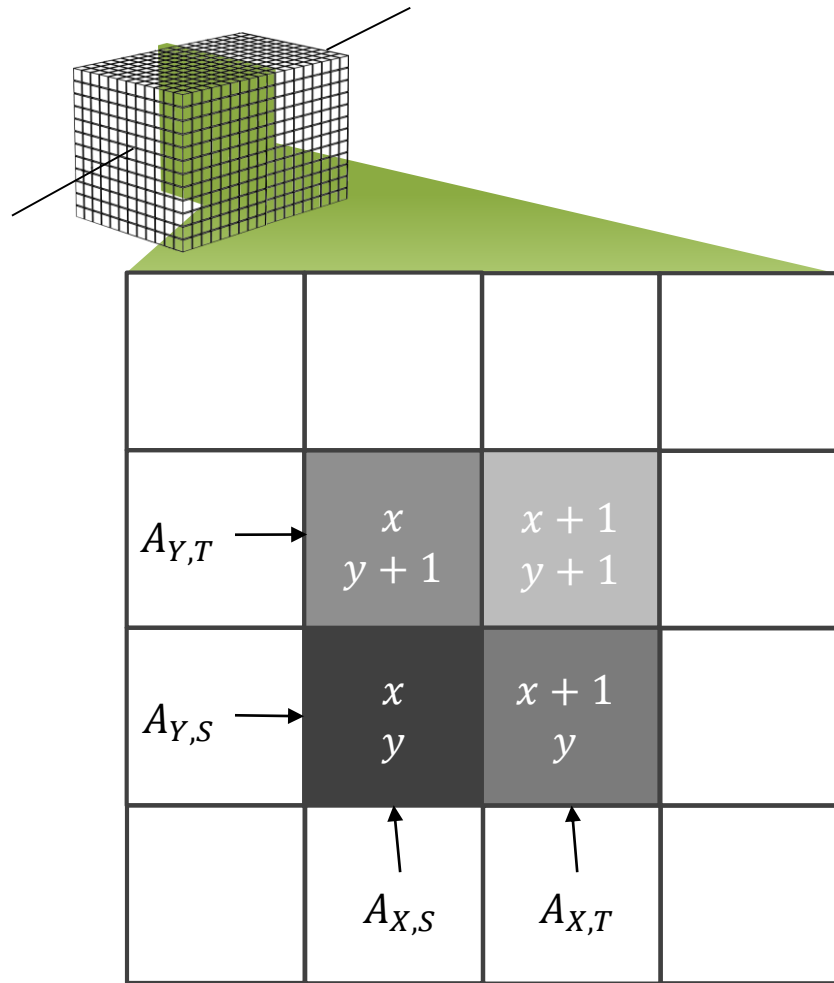


Areas of intersection:

$$A_S \approx (1 - S)$$

$$A_T \approx S$$

$$A_R \approx 0 \text{ (furthest voxel disregarded)}$$





GPU acceleration



CUDA: parallel computing platform for general purpose computing on GPU (GPGPU)



GPU acceleration



CUDA: parallel computing platform for general purpose computing on GPU (GPGPU)

Gather type, i.e. tomograph-space centric approach:

- Tomograph-space voxels are mapped to object space
- Thread-safe, no write conflicts



GPU acceleration



CUDA: parallel computing platform for general purpose computing on GPU (GPGPU)

Gather type, i.e. tomograph-space centric approach:

- Tomograph-space voxels are mapped to object space
- Thread-safe, no write conflicts

Memory operations are expensive → texture memory

- Fast access
- Built-in trilinear interpolator units
- Number of memory accesses decrease from 4 to 1 per slice
- Cache specialized for spatial locality



GPU acceleration



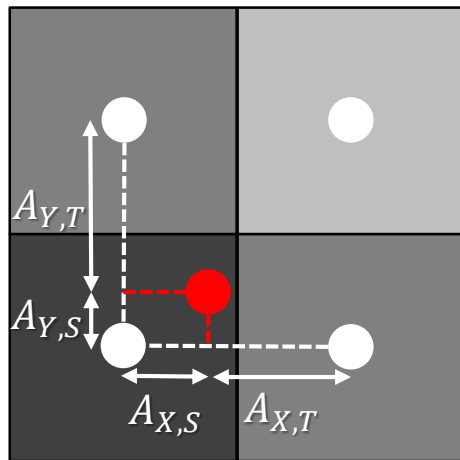
CUDA: parallel computing platform for general purpose computing on GPU (GPGPU)

Gather type, i.e. tomograph-space centric approach:

- Tomograph-space voxels are mapped to object space
- Thread-safe, no write conflicts

Memory operations are expensive → texture memory

- Fast access
- Built-in trilinear interpolator units
- Number of memory accesses decrease from 4 to 1 per slice
- Cache specialized for spatial locality





Examination setup

Questions:

1. The impact of approximating paths with polylines
2. If the path is a polyline, how good is the Antialiased Bresenham Algorithm?

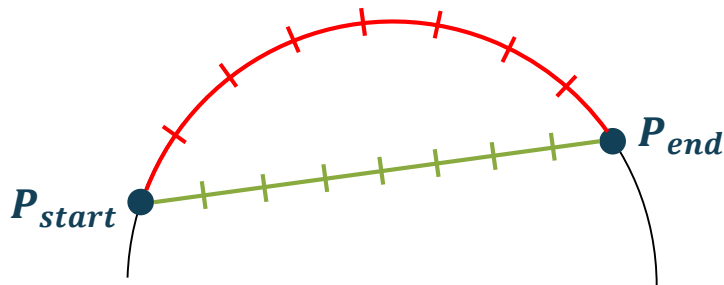


Examination setup

Questions:

1. The impact of approximating paths with polylines
2. If the path is a polyline, how good is the Antialiased Bresenham Algorithm?

Compared algorithms:



- Classical motion blur (CL)
- Ray marching (RM)
- Antialiased Bresenham Algorithm (AB)



Examination setup

- 3 mathematical phantoms
- each in 8 motion scenarios

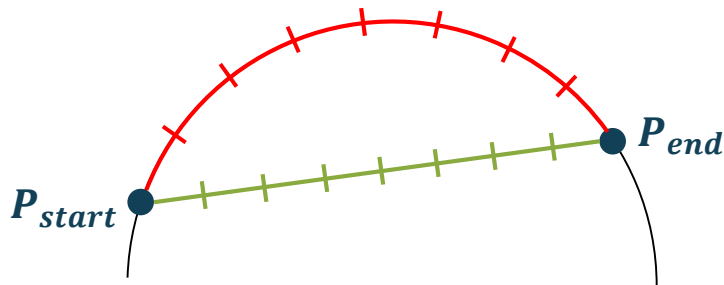
Total: $3 \times 8 = 24$ measurements

NVIDIA GeForce GTX 960M graphics card

Questions:

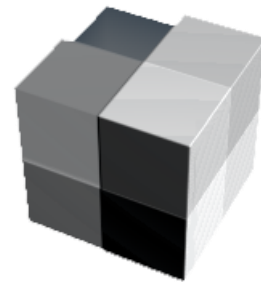
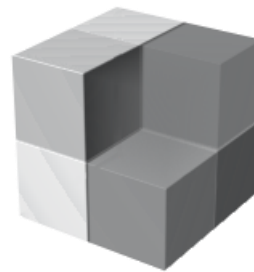
1. The impact of approximating paths with polylines
2. If the path is a polyline, how good is the Antialiased Bresenham Algorithm?

Compared algorithms:

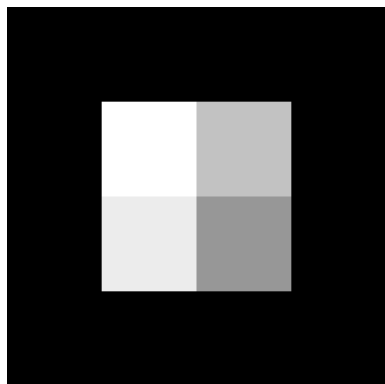


- Classical motion blur (CL)
- Ray marching (RM)
- Antialiased Bresenham Algorithm (AB)

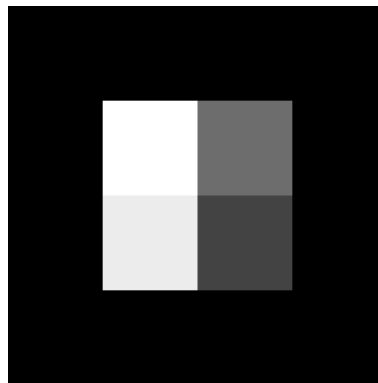
Homogeneity Phantom



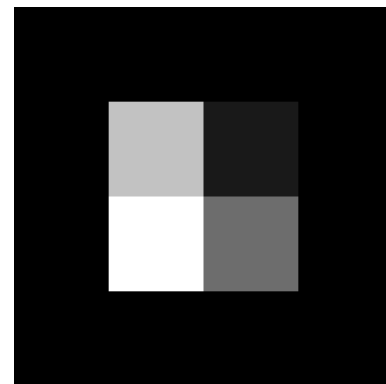
8 constant activity cubes, $256 \times 256 \times 256$ voxels



Sagittal

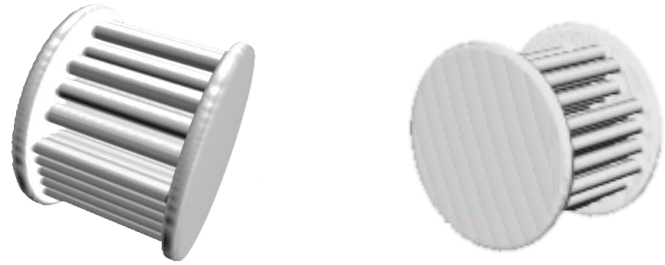


Coronal

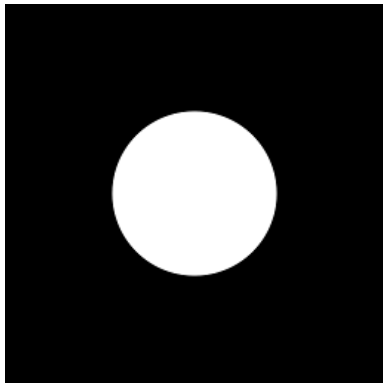


Transverse

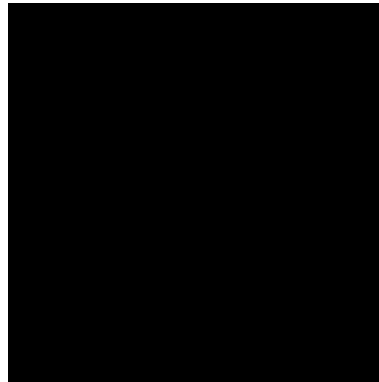
Derenzo Phantom



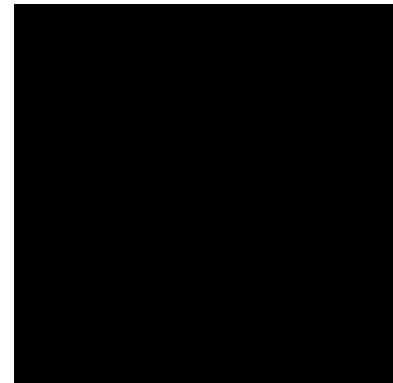
Rods with varying diameters between two plates, **256 × 256 × 256** voxels



Sagittal

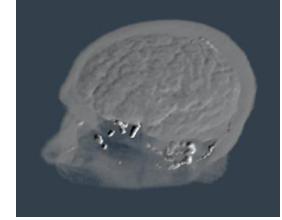
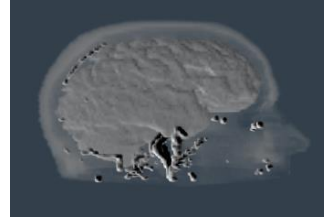


Coronal

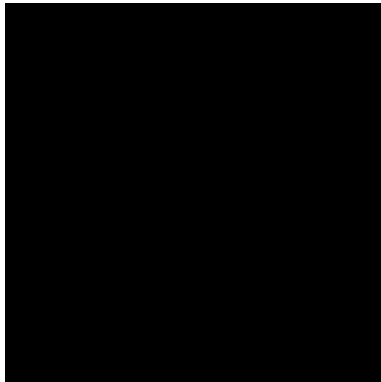


Transverse

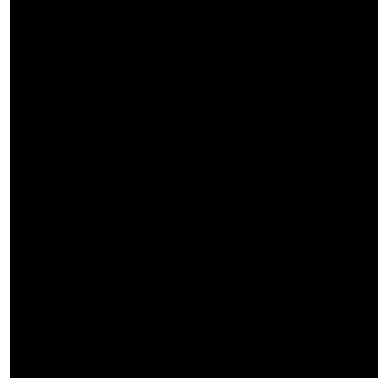
Zubal Brain Phantom



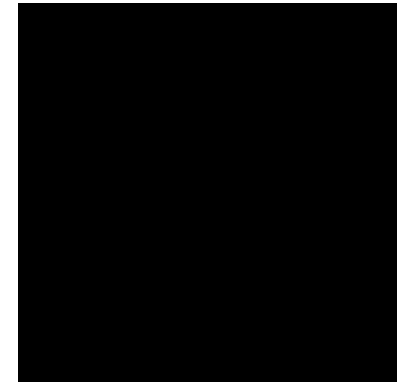
a human's brain inside its skull, **128 × 128 × 64** voxels



Sagittal



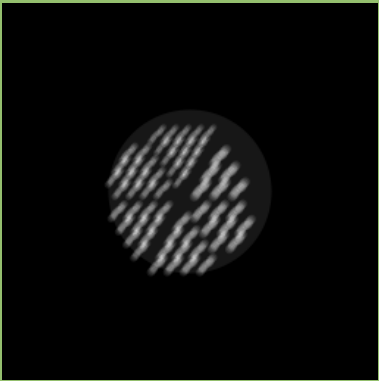
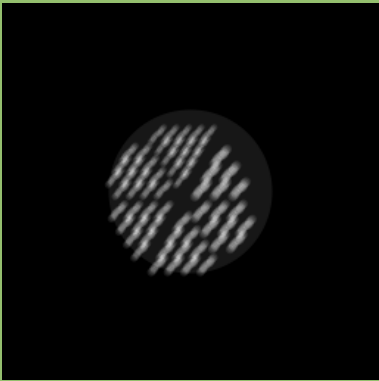
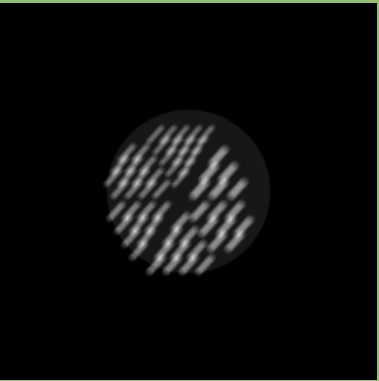
Coronal



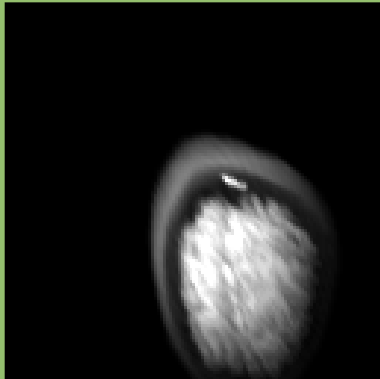
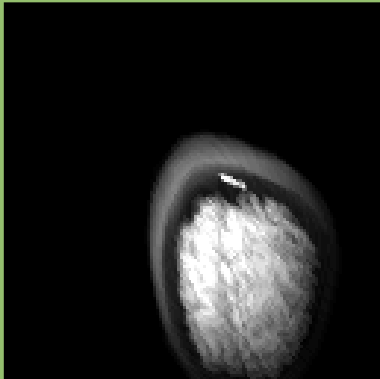
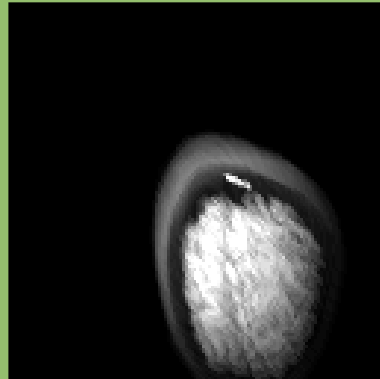
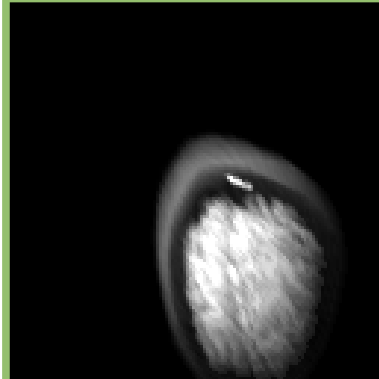
Transverse

Results

Derenzo Phantom, Scenario 2 (-12, 10, -11) voxel translation

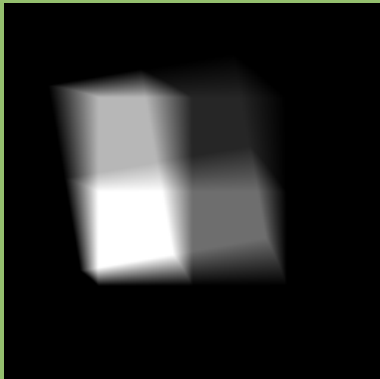
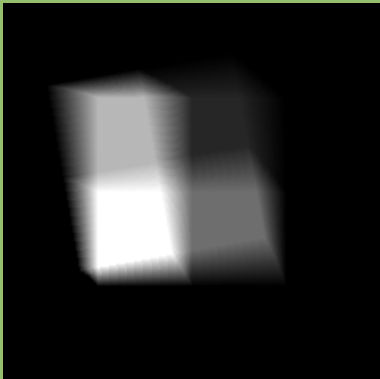
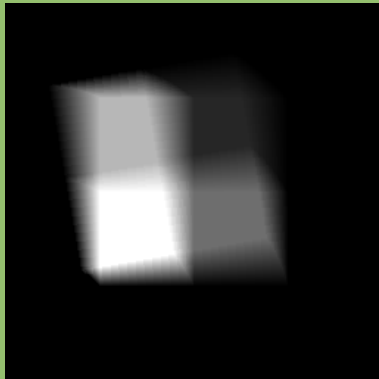
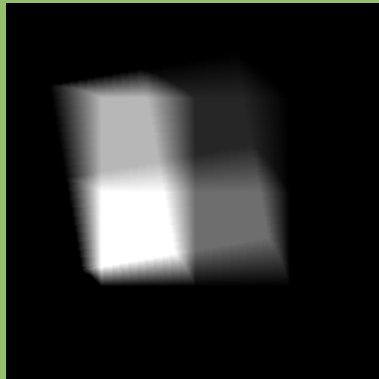
			
Reference	Classical	Ray Marching	Antialiased Bresenham
Error:	0.8664%	0.8664%	0.5819%
Execution time:	167.6756 ms	52.9707 ms	53.7349 ms

Zubal Brain Phantom, Scenario 3 (-20, 30, 15) voxel translation

			
Reference	Classical	Ray Marching	Antialiased Bresenham
Error:	0.5874%	0.5876%	0.0898%
Execution time:	39.7277 ms	11.0532 ms	12.7923 ms

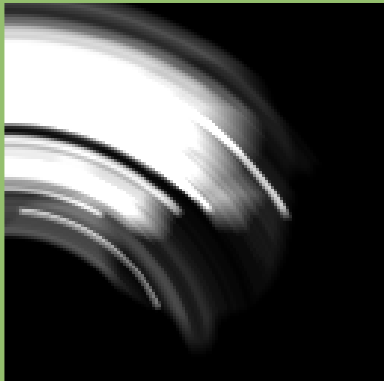
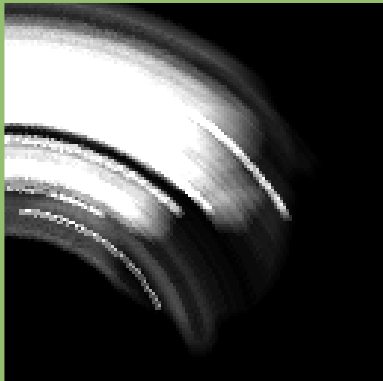
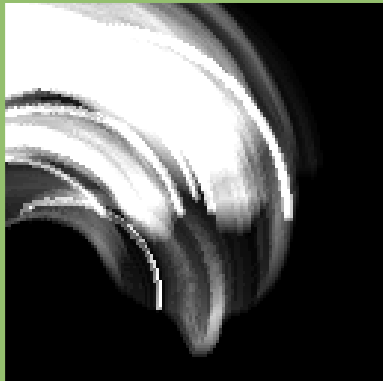
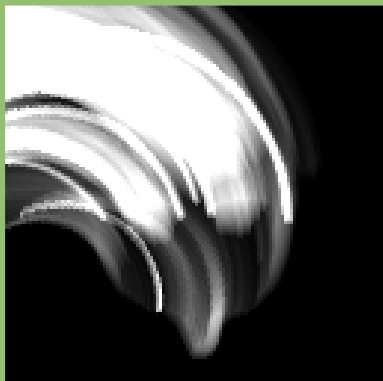
Homogeneity Phantom, Scenario 5

10° rotation

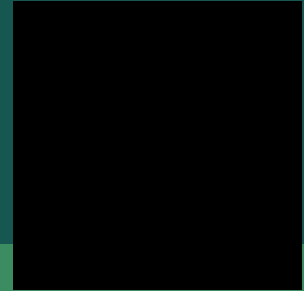
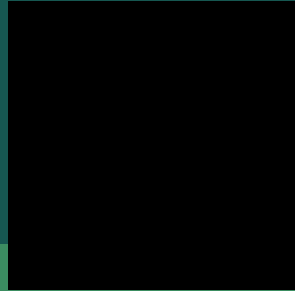
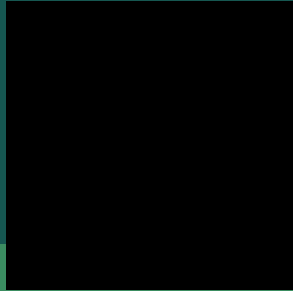
			
Reference	Classical	Ray Marching	Antialiased Bresenham
Error:	0.0278%	0.0474%	0.0473%
Execution time:	454.1396 ms	100.6399	113.5833 ms

Zubal Brain Phantom, Scenario 6

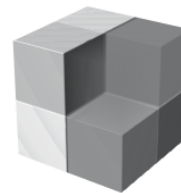
60° rotation

			
Reference	Classical	Ray Marching	Antialiased Bresenham
Error:	0.2541%	27.8996%	27.5332%
Execution time:	2478.4565 ms	278.5847 ms	376.0340 ms

Thank you for your attention!



Homogeneity Phantom



CL: Classical Technique

RM: Ray Marching

AB: Antialiased Bresenham

Scenario	Object motion		Error [%]			Execution time [ms]		
#	Translation	Rotation	CL	RM	AB	CL	RM	AB
1	(2, 2, 2)	0	0.0611	0.0611	0.0611	68.5384	33.6528	34.9513
2	(12, 10, -11)	0	0.0355	0.0355	0.0316	168.5775	52.9623	53.7336
3	(-20, 30, 15)	0	0.0211	0.1337	0.0087	4620.0122	1035.0973	984.2599
4	(-20, 30, 0)	0	0.0229	0.0238	0.0110	335.5022	84.4063	108.6518
5	(0, 0, 0)	10	0.0278	0.0474	0.0473	454.1396	100.6399	113.5833
6	(0, 0, 0)	60	0.0093	17.6623	17.6603	2472.8540	278.6437	377.6001
7	(2.5, -3.2, -0.7)	-5	0.0587	0.0734	0.0787	229.5226	59.7012	61.3481
8	(42, 21, 12)	20	0.0146	0.5404	0.5393	1059.9015	182.7321	218.4056

Derenzo Phantom



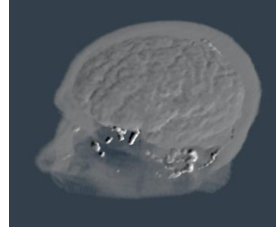
CL: Classical Technique

RM: Ray Marching

AB: Antialiased Bresenham

Scenario	Object motion		Error [%]			Execution time [ms]		
#	Translation	Rotation	CL	RM	AB	CL	RM	AB
1	(2, 2, 2)	0	0.7696	0.7696	0.7696	68.5586	34.8277	34.9368
2	(12, 10, -11)	0	0.8664	0.8664	0.5819	167.6756	52.9707	53.7349
3	(-20, 30, 15)	0	0.4557	0.4655	0.1534	327.9224	84.4359	108.6342
4	(-20, 30, 0)	0	0.1105	0.1105	0.0963	338.2086	86.5922	112.1805
5	(0, 0, 0)	10	0.4108	0.5632	0.4818	458.1706	100.6931	113.6382
6	(0, 0, 0)	60	0.1754	50.4148	50.2363	2478.4565	278.5847	376.0340
7	(2.5, -3.2, -0.7)	-5	1.2496	1.4567	1.1130	222.5278	59.4808	61.4398
8	(42, 21, 12)	20	0.2196	10.1651	9.7513	1062.7415	182.6560	217.8306

Zubal Brain Phantom



CL: Classical Technique

RM: Ray Marching

AB: Antialiased Bresenham

Scenario	Object motion		Error [%]			Execution time [ms]		
#	Translation	Rotation	CL	RM	AB	CL	RM	AB
1	(2, 2, 2)	0	2.2777	2.2777	2.2777	7.8501	4.0458	4.0384
2	(12, 10, -11)	0	1.1337	1.1337	0.3329	19.2592	7.3780	7.2972
3	(-20, 30, 15)	0	0.5874	0.5876	0.0898	39.7277	11.0532	12.7923
4	(-20, 30, 0)	0	0.1854	0.1888	0.0599	43.5365	11.6059	13.9375
5	(0, 0, 0)	10	0.6587	1.0414	0.7998	32.1842	7.9588	9.9130
6	(0, 0, 0)	60	0.2541	27.8996	27.5332	165.0488	19.4201	27.1628
7	(2.5, -3.2, -0.7)	-5	2.5813	2.7822	1.9601	14.9824	6.0723	6.1744
8	(42, 21, 12)	20	0.3330	1.4790	1.0891	99.6831	17.2297	20.1701