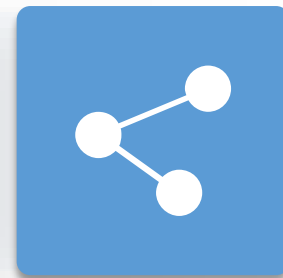




毕业设计终期答辩

——锥形束CT快速重建算法的技术研究

仪器科学与光电工程学院



答辩人：马煜华 14171002

指导老师：周付根



研究背景

RESEARCH BACKGROUNDS

算法实现

ALGORITHM REALIZATION

总结展望

CONCLUSION AND EXPECTATION

1

2

3

4

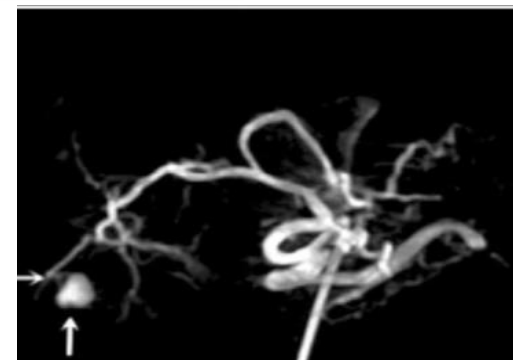
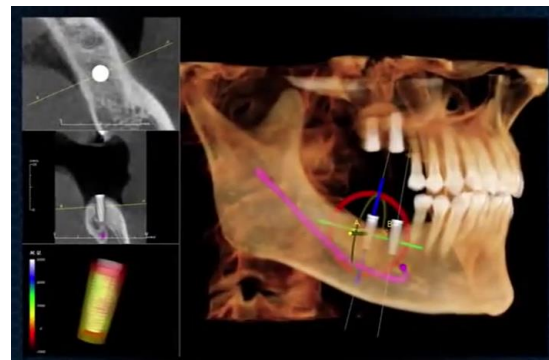
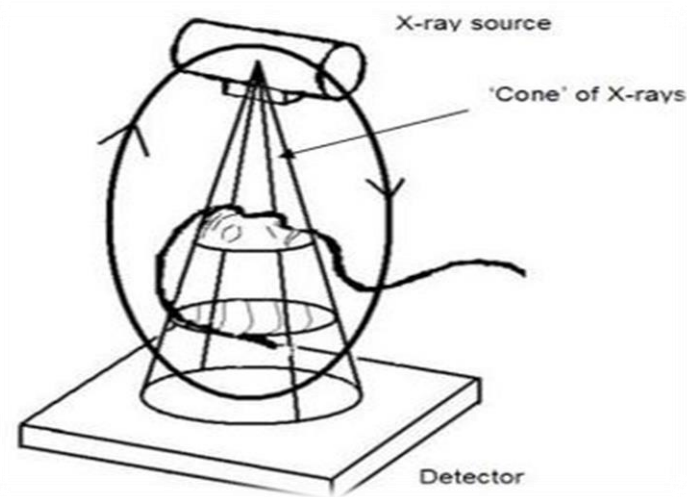
5

理论基础

THEORETIC BASIS

性能分析

PERFORMANCE ANALYSIS



CBCT重建

迭代类算法



精度高，重建速度慢

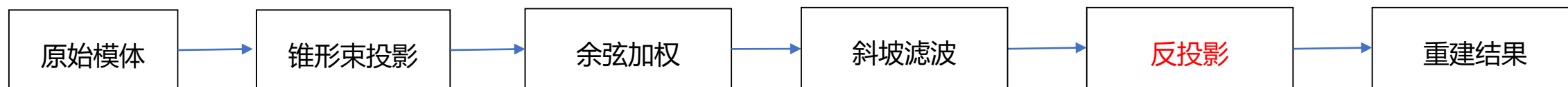
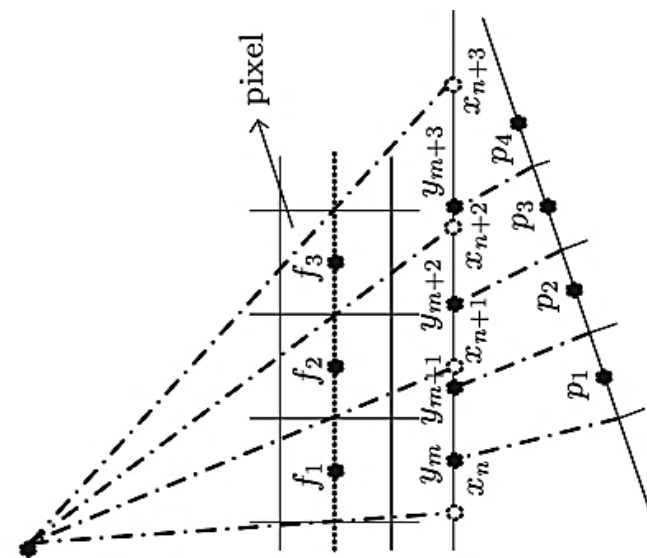
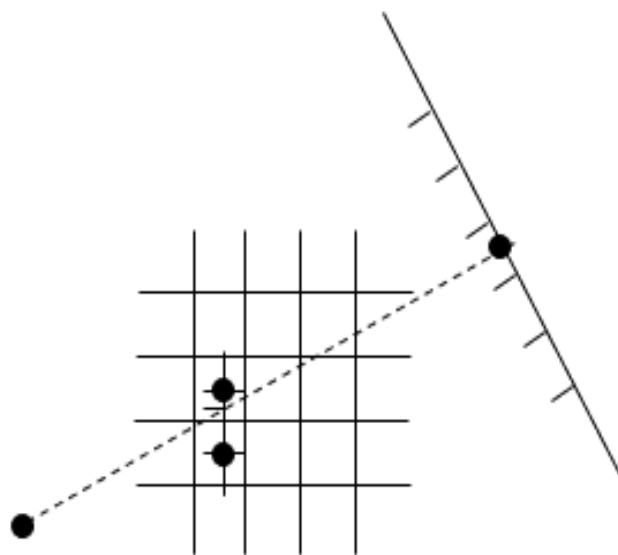
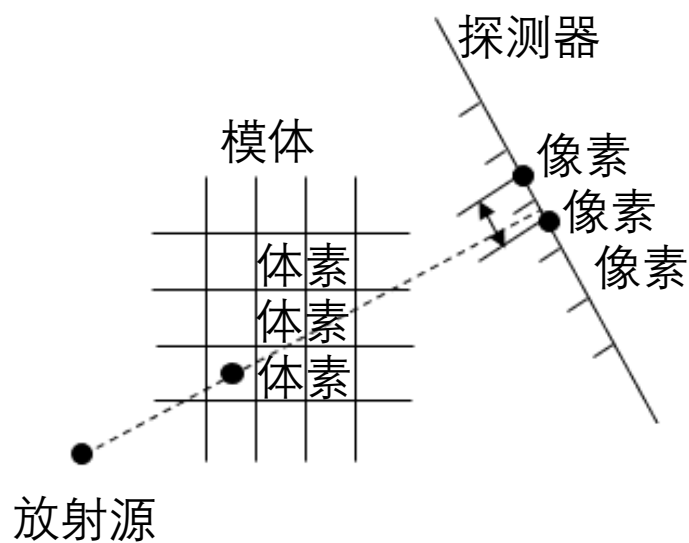
反投影重建算法



精度满足使用需要，重建速度适中

重点：提升反投影速度

反投影重建算法-FDK

体素驱动
Voxel Driven像素驱动
Pixel Driven距离驱动
Distance Driven



距离驱动算法的实现

$$V = \sum_{projs=1}^n \left[\left(\frac{1}{w^2 \sum S_{R_i}} \right) \sum_{i=1}^m P_{R_i} \times S_{R_i} \right]$$

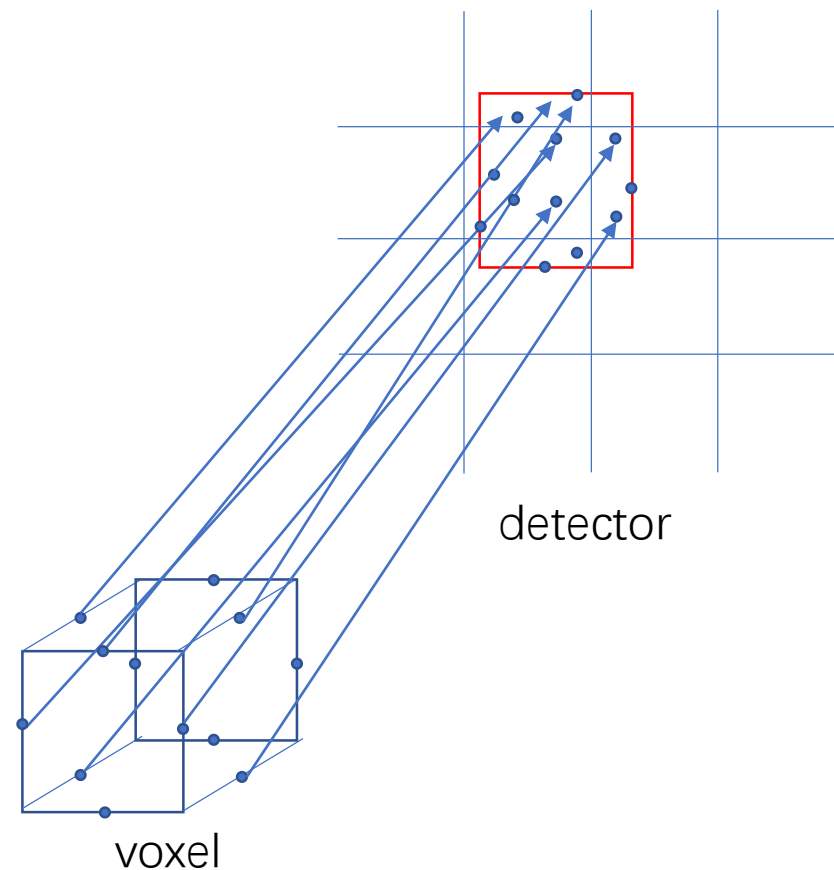
V : 反投影值

$\frac{1}{w^2 \sum S_{R_i}}$: 加权系数

R_i : 第 i 个重叠单元

P_{R_i} : R_i 对应的探测器单元值

S_{R_i} : R_i 对应的重叠单元面积



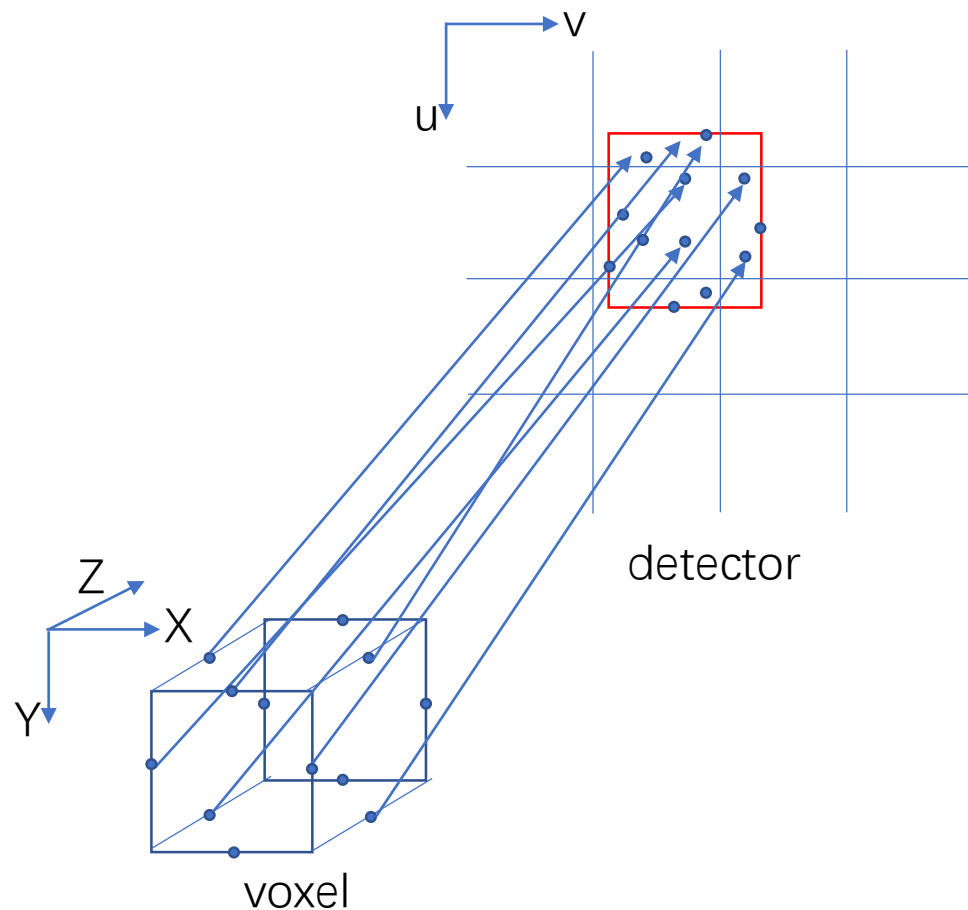
将体素棱边中点投影至投影图平面上
使用近似的重叠面积作为反投影权重



距离驱动算法的实现

$$V = \sum_{projs=1}^n \left[\left(\frac{1}{w^2 \sum S_{R_i}} \right) \sum_{i=1}^m P_{R_i} \times S_{R_i} \right]$$

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_p^{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



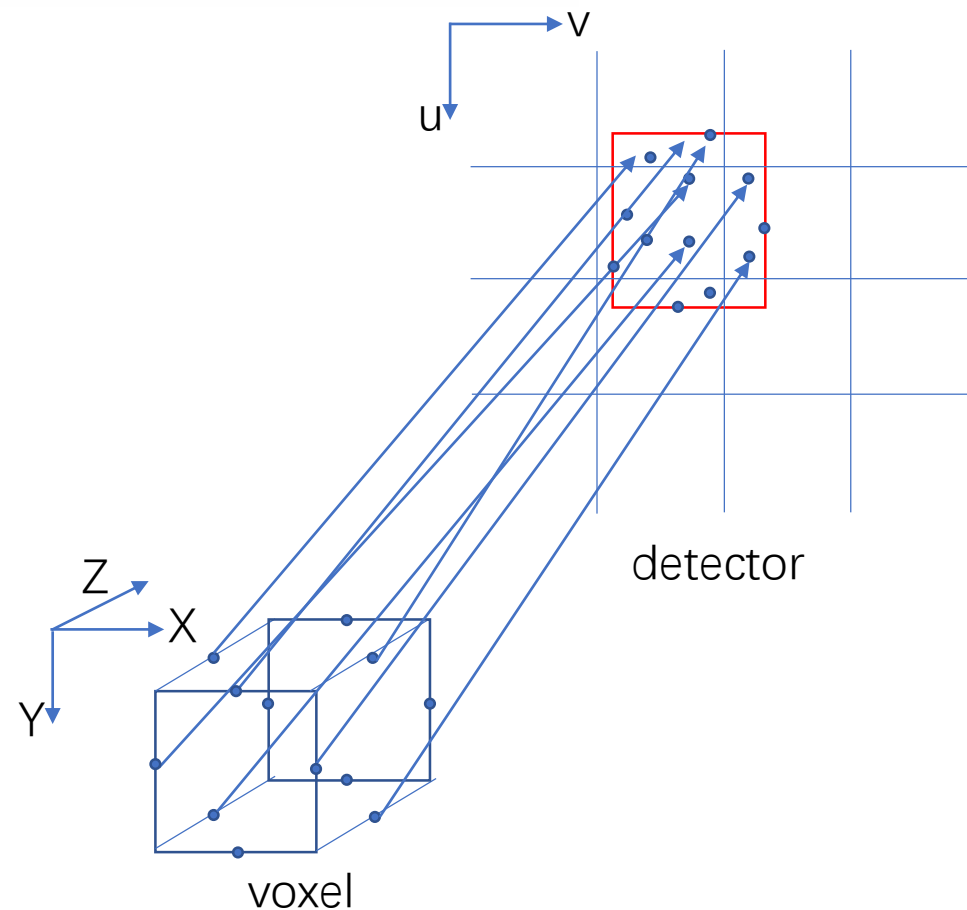
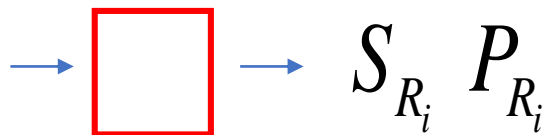
将体素棱边中点投影至投影图平面上
使用近似的重叠面积作为反投影权重



距离驱动算法的实现

$$V = \sum_{projs=1}^n \left[\left(\frac{1}{w^2 \sum S_{R_i}} \right) \sum_{i=1}^m P_{R_i} \times S_{R_i} \right]$$

12个边界点的投影值
在u, v方向上的极值

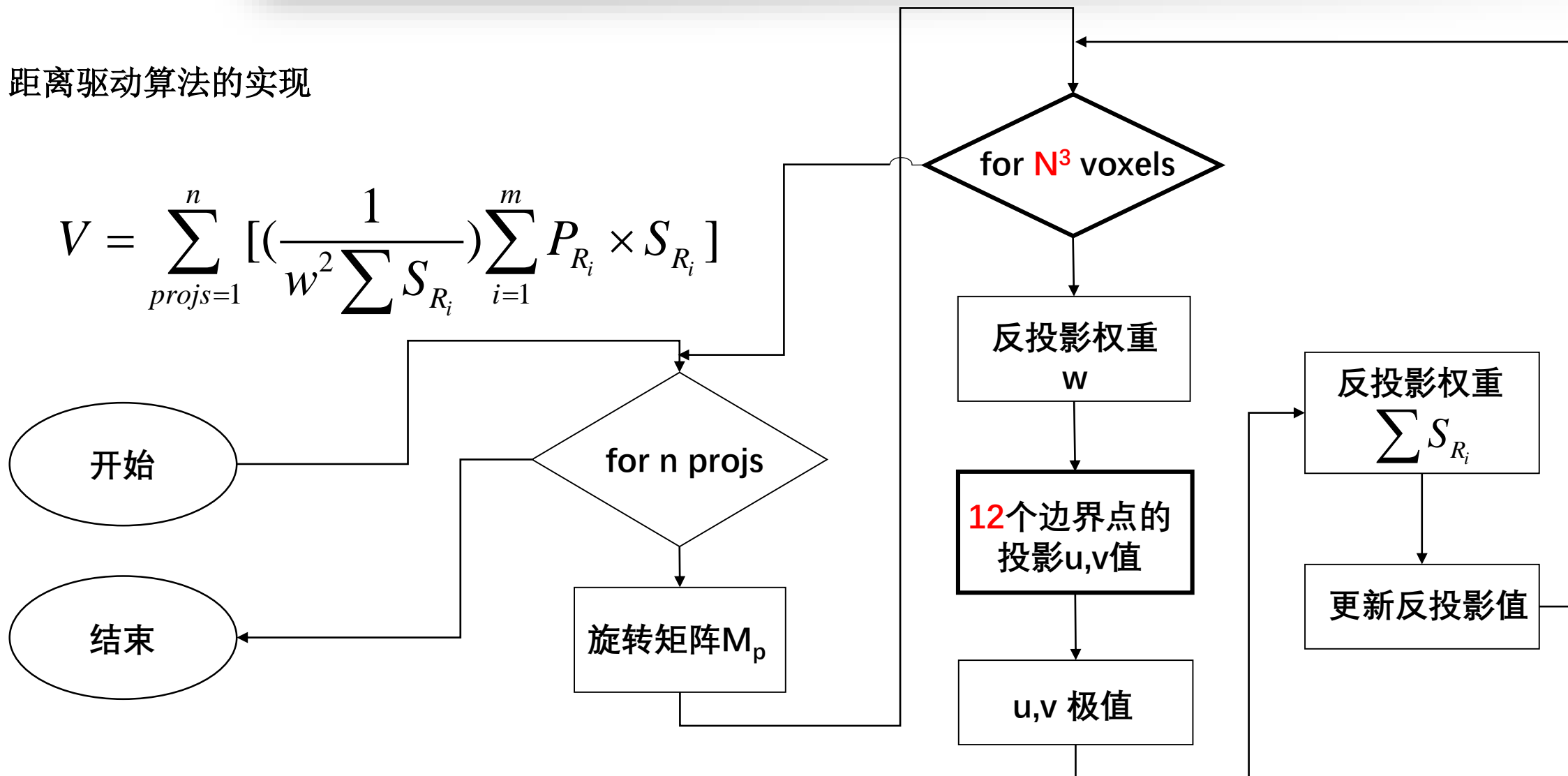


将体素棱边中点投影至投影图平面上
使用近似的重叠面积作为反投影权重

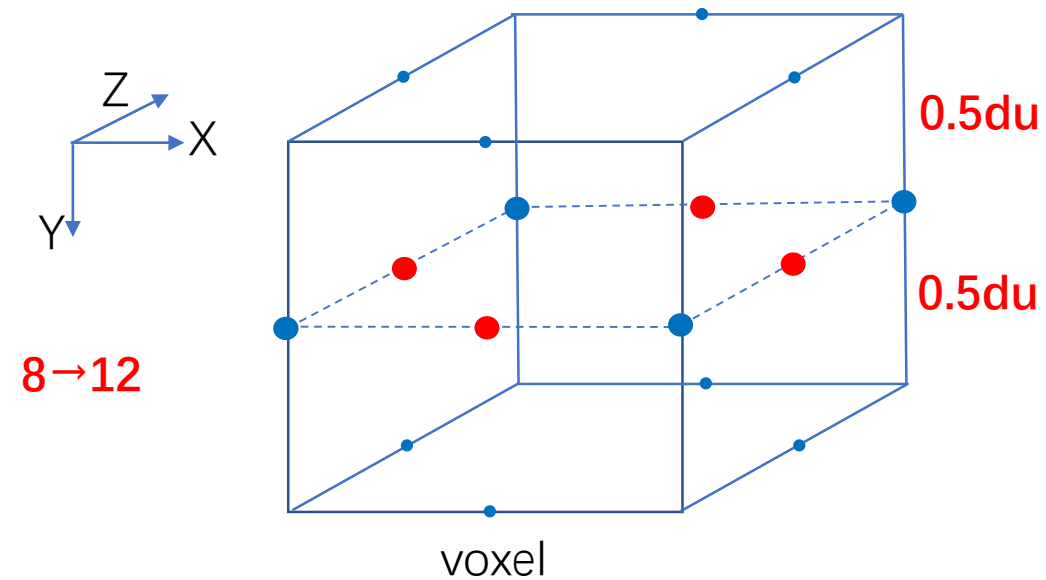
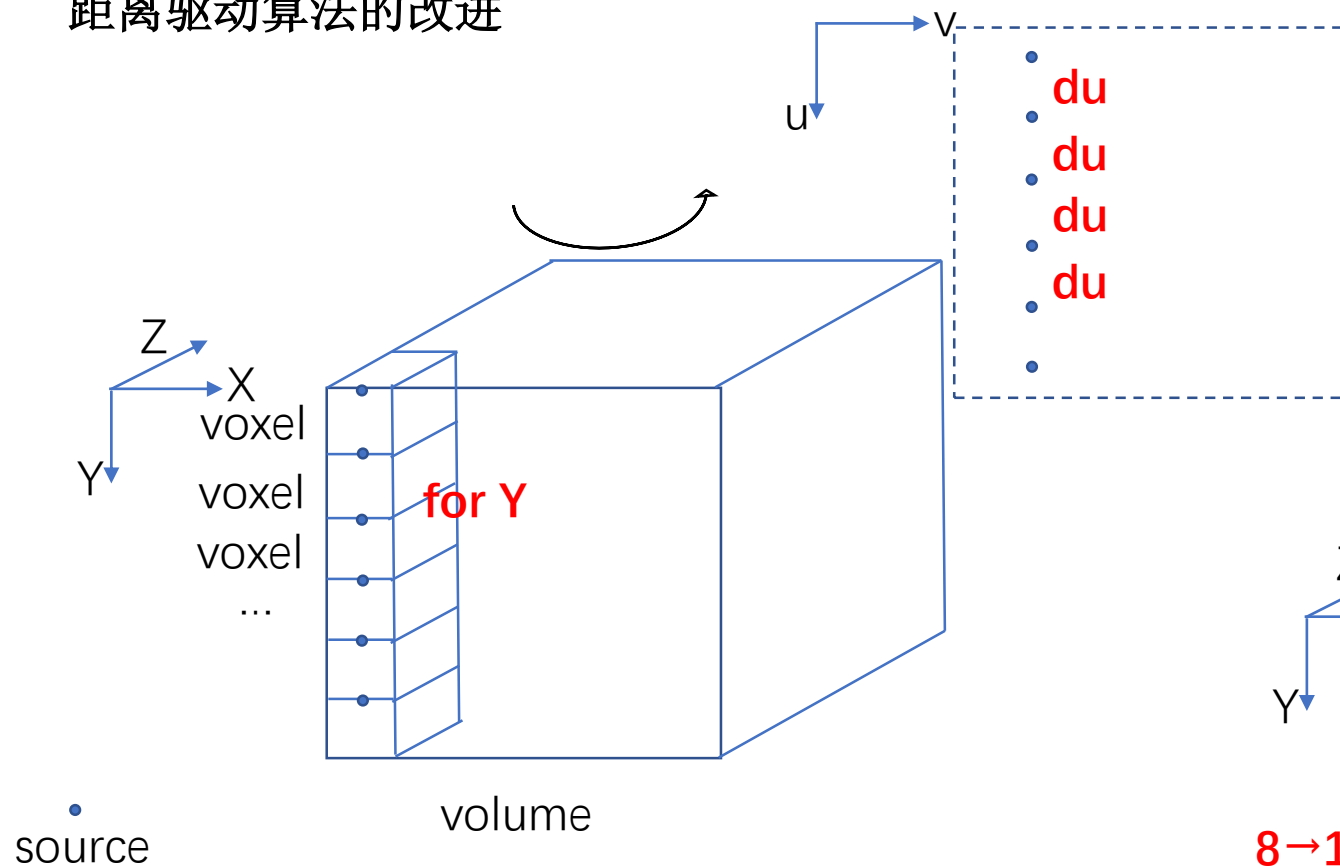


距离驱动算法的实现

$$V = \sum_{projs=1}^n \left[\left(\frac{1}{w^2 \sum S_{R_i}} \right) \sum_{i=1}^m P_{R_i} \times S_{R_i} \right]$$

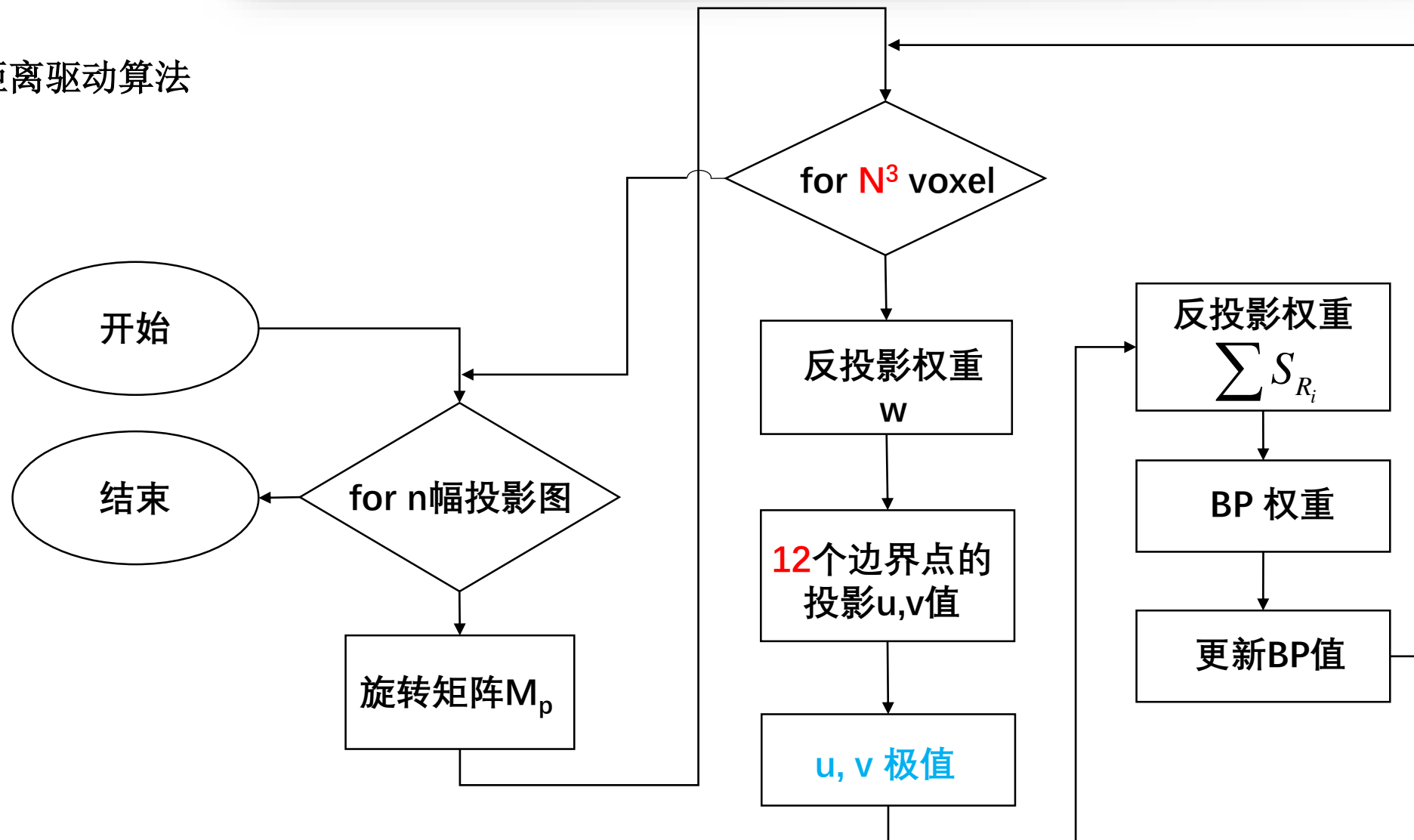


距离驱动算法的改进



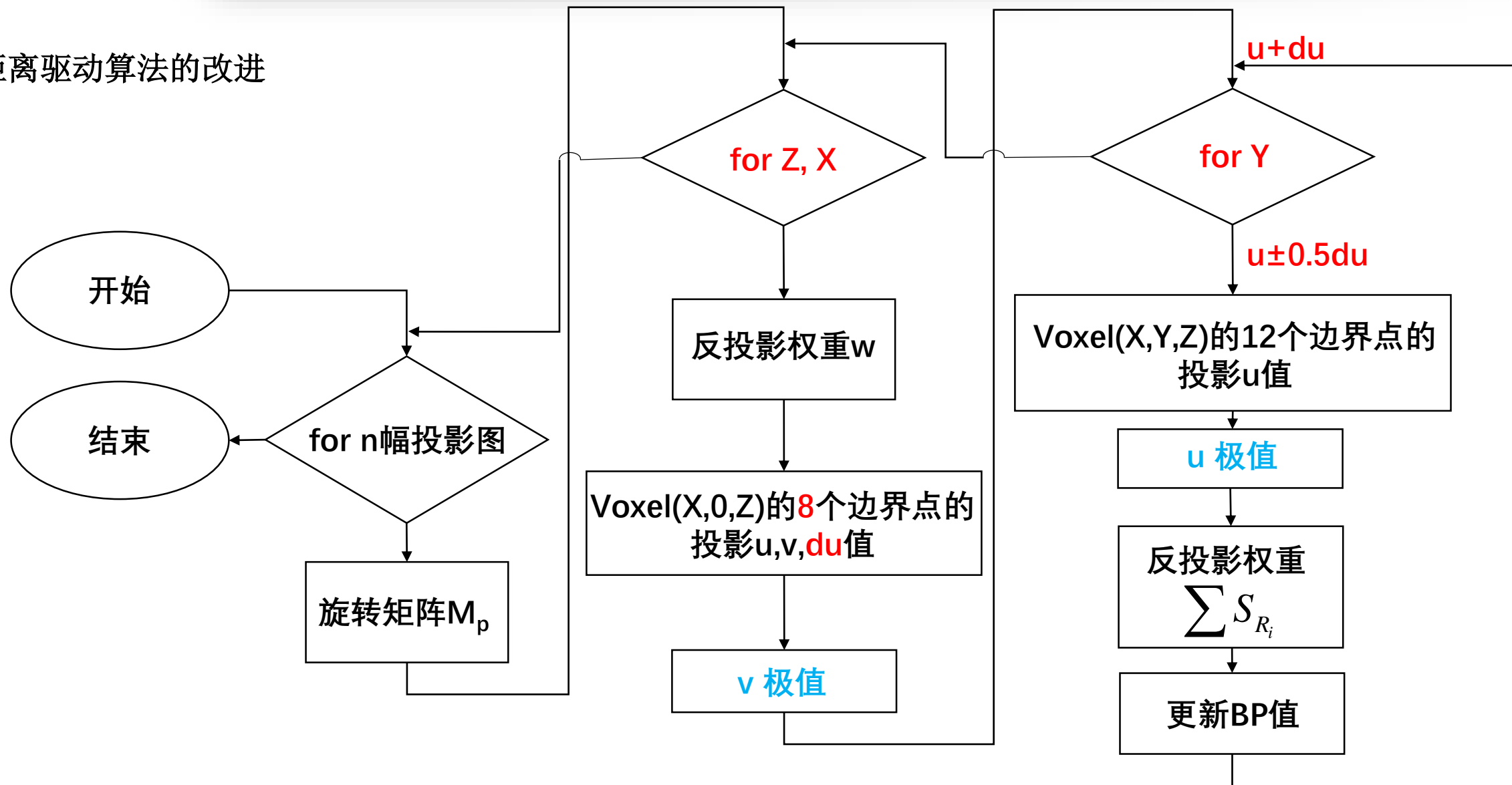


距离驱动算法





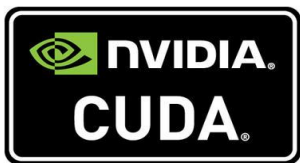
距离驱动算法的改进





重建参数	值
原始模体	Shepp-Logan
生成仿真投影图的类	RayEllipsoidIntersectionImageFilter
投影图参数	尺寸256*256 间距2 全扫描方式得180幅投影图
反投影参数	尺寸256*256*256 间距2
射影几何参数	SDD=1536mm, SID=1000mm

方法	驱动方式	硬件环境	线程数	重建一幅投影图的耗时	PSNR/dB
Distance-driven	距离驱动	CPU	8	15s	31.01
Distance-driven Opti.	距离驱动	CPU	8	1.5s	31.01



距离驱动算法的GPU加速

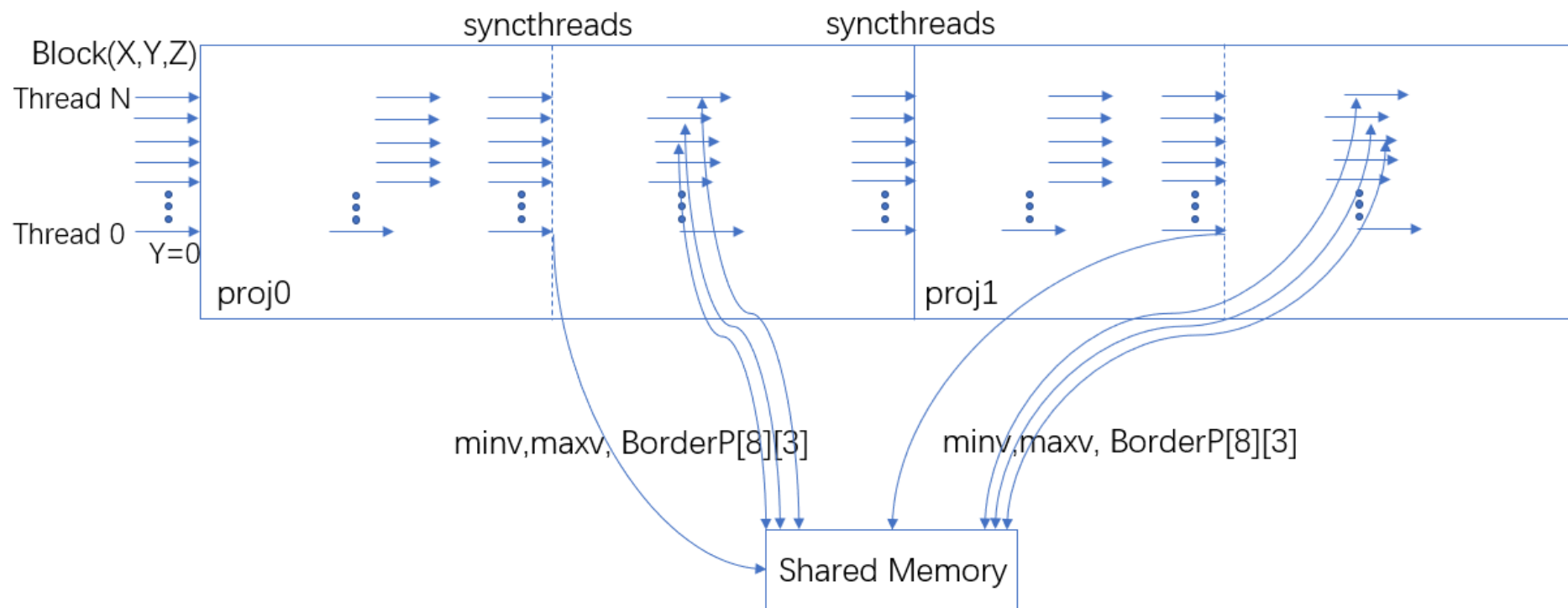
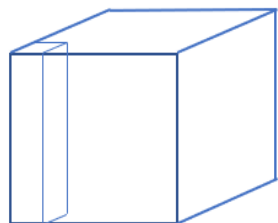
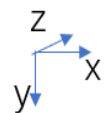
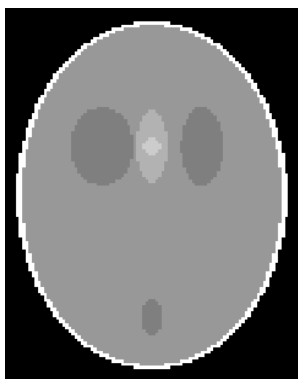


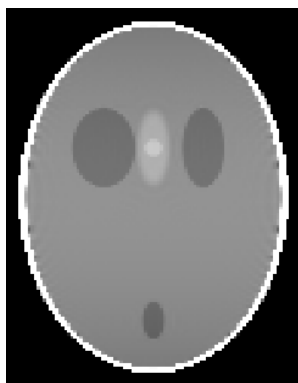


表 5.1 重建时间比较

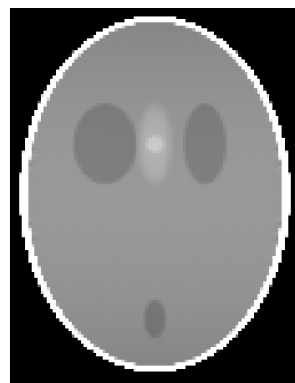
方法	驱动方式	硬件环境	重建一幅投影图的耗时
Voxel-based	体素驱动	CPU	16.7ms
CudaBackProjection	体素驱动	GPU	6.5ms
Distance-driven Opti.	距离驱动	CPU	1.5s
Cuda Distance-driven Opti.	距离驱动	GPU	97ms
CudaRayCast	像素驱动	GPU	129ms



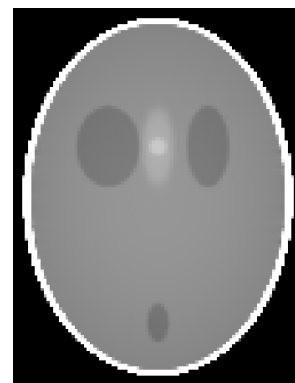
原模体图



体素驱动



距离驱动

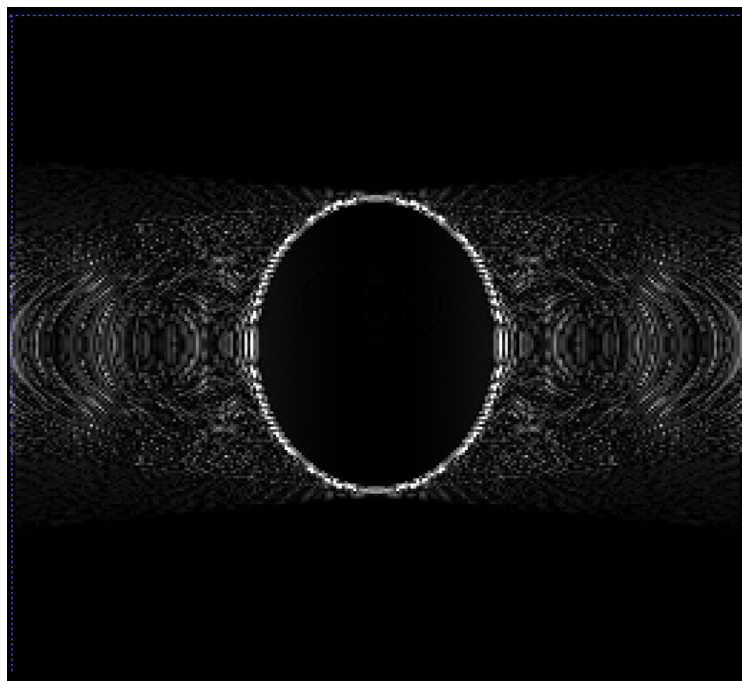
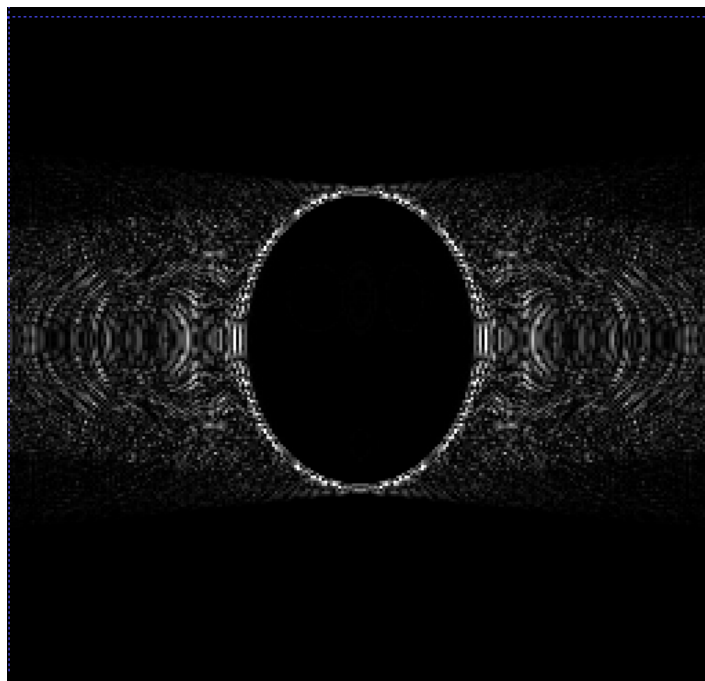
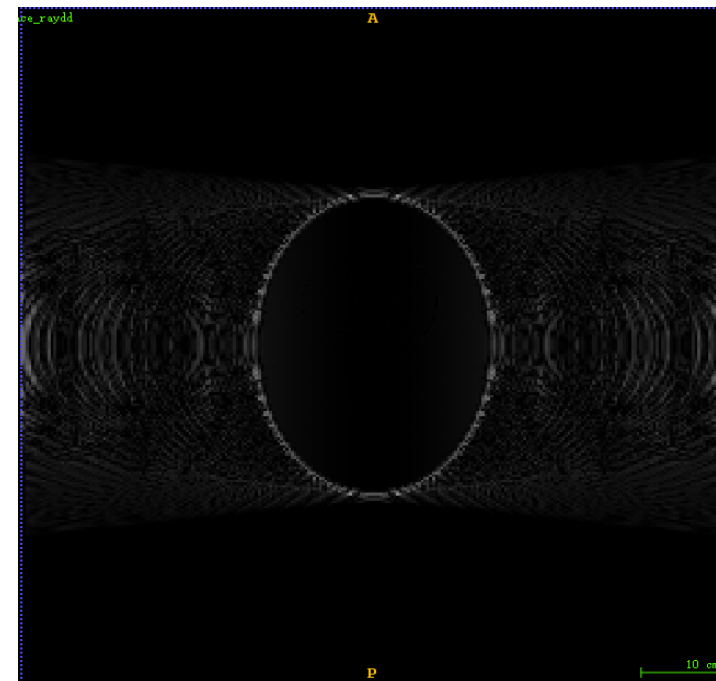


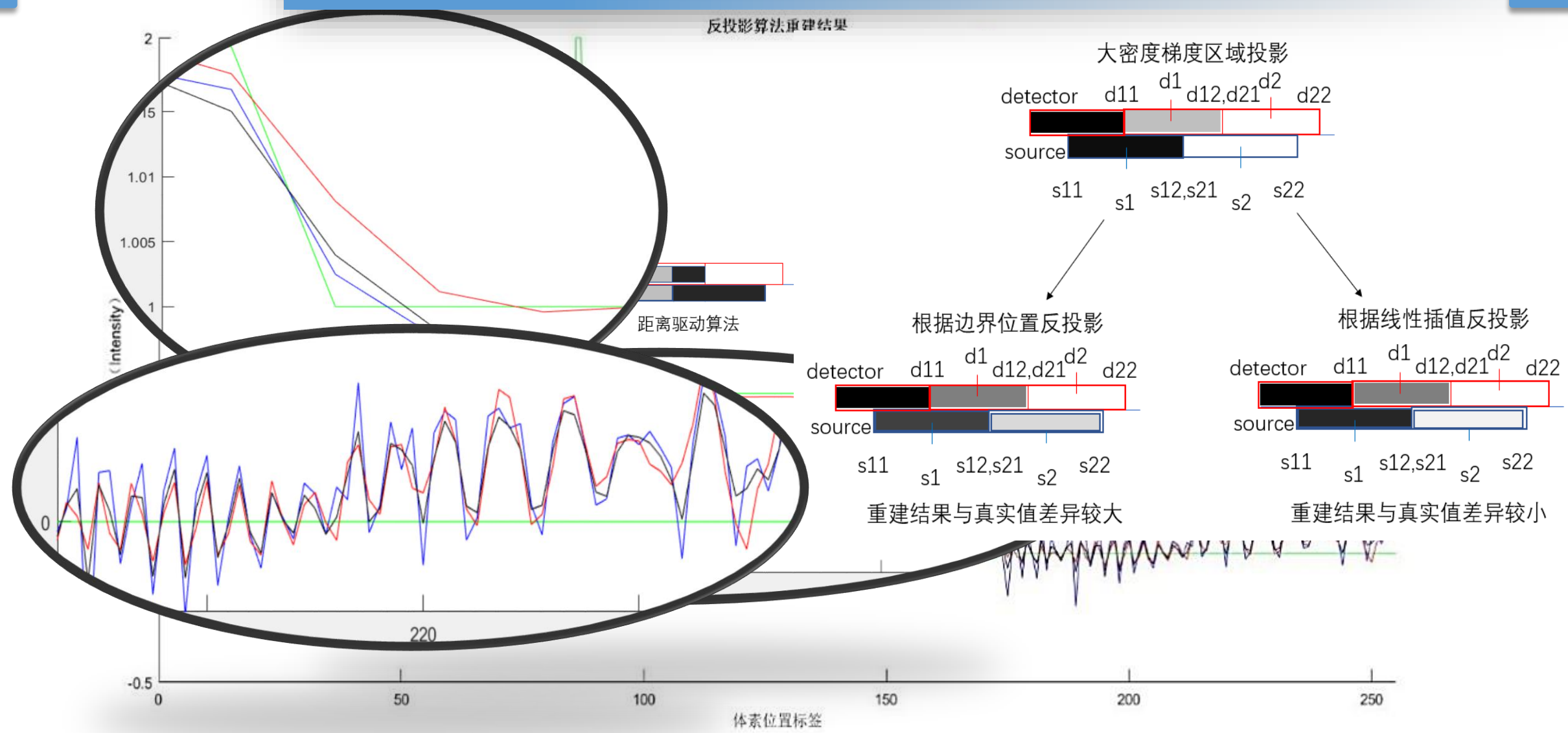
像素驱动



表 5.2 重建精度比较

反投影方式	体素驱动		像素驱动	距离驱动	
	Voxel-based	CudaVoxel-based	CudaRayCast	Distance-driven Opti.	CudaDistance-driven Opti.
PSNR/dB	30.51	30.54	31.46	31.01	31.27

体素驱动误差与像素驱动误差的
做差结果体素驱动误差与距离驱动误差的
做差结果像素驱动误差与距离驱动误差的
做差结果



取原模体与三种驱动方式所得重建图像
在(1, 122, 140)至(256, 122, 140)体素坐标下的密度值进行分析

表 5.3 鲁棒性比较 噪声条件：高斯白噪声， $\mu=0$ ， $\sigma=0.5$

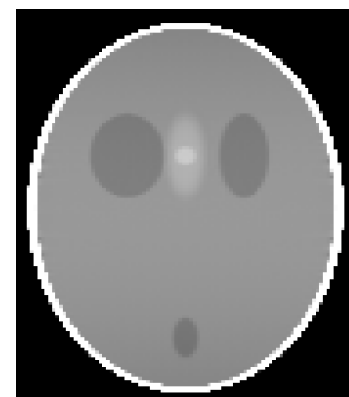
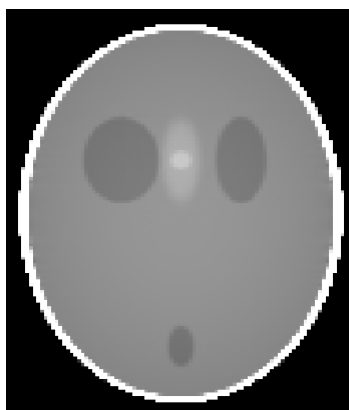
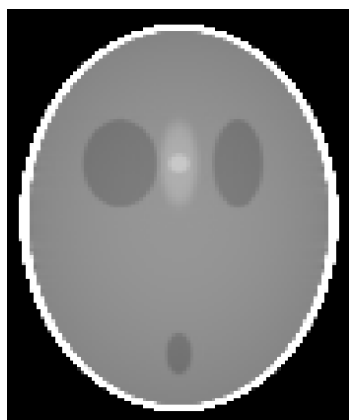
方法	驱动方式	PSNR/dB	加入噪声后 PSNR/dB	PSNR 变换情况
CudaBackProjection	体素驱动	30.54	30.072	0.468
CudaDistance-driven Opti.	距离驱动	31.27	31.16	0.11
CudaRayCast	像素驱动	31.46	31.305	0.155

体素驱动

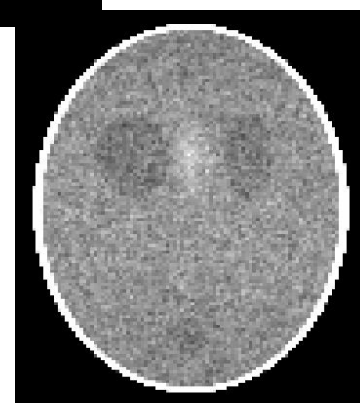
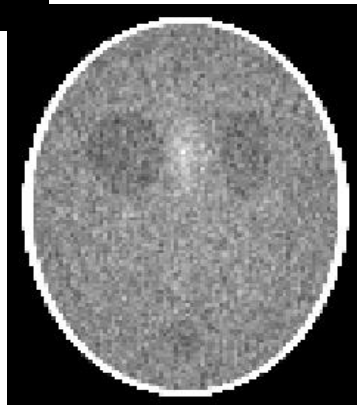
像素驱动

距离驱动

未加噪声



加噪声





总结

锥形束CT重建的原理及FDK算法

体素驱动、像素驱动、距离驱动

编程实现距离驱动的FDK算法

距离驱动边界点选取方案进行改进

距离驱动算法的GPU加速

展望

不能达到旋转轴与X或Y坐标轴的绝对平行，寻找补偿方式的数学模型

自适应得到GPU配置参数

边界点的选取对重建速度精度的影响及背后的数学原理



感谢

答辩人：马煜华 指导老师：周付根

THANKS

