

# 基于变分和稀疏表示的定量 MR 图像 快速重建模型和加速算法

王冬 (东南大学)

杨孝平教授 (南京大学)

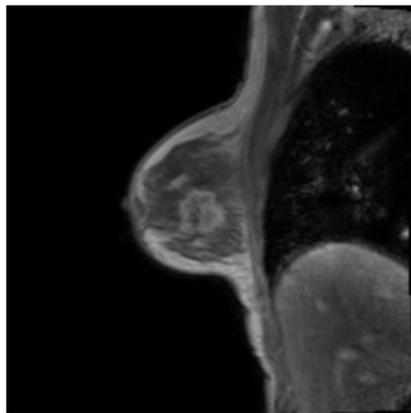
Prof. David Smith (Vanderbilt University)

2020 年 08 月 28 日

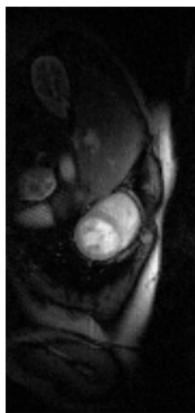
- ① 基于广义全变分和低秩分解的动态 MR 图像重建
- ② 基于图形处理单元的实时 MRF 字典生成与匹配算法

## 核磁共振成像

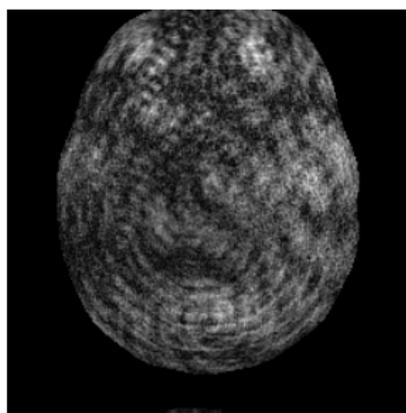
核磁共振成像 (MRI)，尤其是动态 MRI 和定量 MRI 是医学临床和研究中常用的成像方式。



磁共振动态造影增强



心脏灌注



活体人脑磁共振指纹

## 动态 MR 成像

记动态 MR 图像为  $X \in \mathbb{C}^{N_1 \times N_2 \times d}$ ，则动态 MR 成像的采样的过程相当于在噪声的干扰下，在 Fourier 域 (k-t space) 中进行采样

$$B = AX + \epsilon, \quad (1)$$

其中  $B$  为采集到的数据， $A = \mathcal{M} \cdot \mathcal{F}$  是采样算子， $\mathcal{F}$  是作用在每一帧图像上的二维 Fourier 变换， $\mathcal{M}$  是作用在每一帧图像上的二维采样模式， $\epsilon$  是加性高斯白噪声。

**问题：成像速度慢！**

## 基于压缩感知的动态 MR 图像重建模型

- ▶ 压缩感知理论是由 Candès 等提出的采样理论，是目前快速 MR 成像的主流方法。
- ▶ 假设图像在某个变换域**稀疏**的前提下，对图像进行**非相干下采样**，通过**非线性重建**算法可以高概率地将图像重建出来。

基于压缩感知的动态 MR 图像重建模型为：

$$\min_X \frac{1}{2} \|AX - B\|_F^2 + \alpha \|\Phi X\|_1, \quad (2)$$

其中  $\Phi$  是某个稀疏变换， $\alpha > 0$  是平衡数据项和稀疏正则项的参数， $F$  为 Frobenius 范数。这里  $A$  为下采样算子。

## 基于压缩感知的动态 MR 图像重建模型

基于压缩感知模型的动态 MR 图像重建有以下三个策略：

▶ 1. 仅利用动态 MR 图像的稀疏性

**kt-SPARSE (2006):**

$$\min_X \frac{1}{2} \|AX - B\|_F^2 + \alpha \|WX\|_1 + \beta \|\mathcal{F}_t X\|_1, \quad (3)$$

其中  $W$  是空间方向上的二维小波变换，而  $\mathcal{F}_t$  是时间上的一维 Fourier 变换。

**问题：适用于动态心脏成像，在胸部 DCE-MRI 中表现有限**

## 基于压缩感知的动态 MR 图像重建模型

### ► 2. 考虑动态 MR 图像的低秩性，找到一个既稀疏又低秩的解

这时通常将动态 MR 图像转化为一个时空矩阵，矩阵的每一列相当于一帧，矩阵的每一行相当于一个像素点。

$$X = \begin{bmatrix} X(p_1, t_1) & \cdots & X(p_1, t_d) \\ \vdots & \ddots & \vdots \\ X(p_{N_1 \times N_2}, t_1) & \cdots & X(p_{N_1 \times N_2}, t_d) \end{bmatrix},$$

其中  $p_i, i = \{1, \dots, N_1 \times N_2\}$  为空间坐标， $t_i, i = \{1, \dots, d\}$  为时间坐标。

## 基于压缩感知的动态 MR 图像重建模型

**kt-SLR (2011):**

$$\min_X \frac{1}{2} \|AX - B\|_F^2 + \alpha \text{TV}(x) + \beta \|X\|_*. \quad (4)$$

这里

$$\text{TV}(\cdot) = \|\nabla_{\mu} \cdot\|_1, \quad |\nabla_{\mu} \cdot| = \sqrt{(\nabla_x \cdot)^2 + (\nabla_y \cdot)^2 + \mu(\nabla_t \cdot)^2},$$

其中  $\nabla_x$ ,  $\nabla_y$  和  $\nabla_t$  分别为  $x$ ,  $y$  和  $t$  方向上的梯度算子,  $\mu$  是平衡空间稀疏性与时间稀疏性的参数,  $\|\cdot\|_*$  为核范数, 是矩阵奇异值 ( $\sigma_i$ ) 的和, 即  $\|X\|_* = \sum_i \sigma_i$ .

**问题: 重建图像中易出现阶梯效应, 重建时间长**

## 基于压缩感知的动态 MR 图像重建模型

- ▶ 3. 利用图像分解的思想，将动态 MR 图像分解为稀疏部分和低秩部分的加和，并且分别用不同的正则项来约束

Otazo 将图像分解的思想应用到动态 MR 图像重建中，提出模型 **L+S (2013)**：

$$\min_{L,S} \frac{1}{2} \|A(L+S) - B\|_F^2 + \alpha \|\nabla_t S\|_1 + \beta \|L\|_* \quad (5)$$

Trémouhéc 提出类似的模型 **kt-RPCA (2014)**：

$$\min_{L,S} \frac{1}{2} \|A(L+S) - B\|_F^2 + \alpha \|\mathcal{F}_t S\|_1 + \beta \|L\|_* \quad (6)$$

**问题：**仅利用了时间方向的稀疏性

## 基于压缩感知的动态 MR 图像重建模型

Schloegl 等提出了卷积下确界 TGV 泛函 (ICTGV):

$$\text{ICTGV}_{\alpha, \beta}^2(X) = \inf_{X=X_1+X_2} \text{TGV}_{\alpha_1}^2(X_1) + \beta \text{TGV}_{\alpha_2}^2(X_2).$$

这里二阶 TGV 泛函的定义为:

$$\text{TGV}_{\alpha}^2(X) = \min_w \alpha_1 \|\nabla_{\mu} X - w\|_1 + \alpha_0 \|\mathcal{E}_{\mu}(w)\|_1,$$

其中  $\mathcal{E}_{\mu}(w) = (\nabla_{\mu} w + \nabla_{\mu} w^T)/2$  是对称梯度算子。相应的重建模型为 **ICTGV (2017)**:

$$\min_X \frac{1}{2} \|AX - B\|_F^2 + \text{ICTGV}_{\alpha, \beta}^2(X). \quad (7)$$

**问题:** 在胸部 DCE-MRI 中的表现未知, 重建时间长

## 基于 TGV 和核范数的动态 MR 图像重建模型

总之，上述模型存在着**阶梯效应**、**仅利用时间方向的稀疏性**、**重建时间长**等不足，且在**胸部 DCE-MRI**中的表现未知。

针对动态 MR 图像，基于图像分解的思想，利用 TGV 泛函和核范数，提出如下模型：

$$\min_{L,S} \frac{1}{2} \|A(L+S) - B\|_F^2 + \text{TGV}_\alpha^2(S) + \beta \|L\|_* \quad (8)$$

- ▶ 核范数 → 建模时间上高度相关的背景
- ▶ TGV 泛函 → 建模背景之上的动态信息

## 模型的求解

Primal-Dual 算法被广泛地应用到寻找凸-凹鞍点问题的极大极小问题中：

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle \mathcal{K} x, y \rangle + f(x) - g(y). \quad (9)$$

这里算子  $\mathcal{K} : \mathcal{X} \rightarrow \mathcal{Y}$  是线性连续映射，泛函  $f : \mathcal{X} \rightarrow (-\infty, \infty]$  和  $g : \mathcal{Y} \rightarrow (-\infty, \infty]$  是适定的、凸的和下半连续的。

Chambolle-Pock 迭代格式：

$$\begin{cases} y^{n+1} = (I + \sigma \partial g)^{-1} (y^n + \sigma \mathcal{K} \bar{x}^n), \\ x^{n+1} = (I + \tau \partial f)^{-1} (x^n - \tau \mathcal{K}^* y^{n+1}), \\ \bar{x}^{n+1} = 2x^{n+1} - x^n. \end{cases}$$

## 模型的求解

将模型 (8) 离散化:

$$\begin{aligned} \min_{(S,w,L) \in U \times V \times U} \quad & \alpha_1 \|\nabla_{\mu} S - w\|_1 + \alpha_0 \|\mathcal{E}_{\mu}(w)\|_1 + \beta \|L\|_* \\ & + \frac{1}{2} \|A(L + S) - B\|_{\mathbb{F}}^2. \end{aligned} \quad (10)$$

相应的鞍点问题为:

$$\begin{aligned} \min_{(S,w,L) \in U \times V \times U} \quad & \max_{(p,q,\lambda) \in V \times W \times U} \langle \nabla_{\mu} S - w, p \rangle + \langle \mathcal{E}_{\mu}(w), q \rangle + \beta \|L\|_* \\ & + \langle A(L + S) - B, \lambda \rangle - \frac{1}{2} \|\lambda\|_{\mathbb{F}}^2 \\ & - \mathcal{J}_{\|\cdot\|_{\infty} \leq \alpha_1}(p) - \mathcal{J}_{\|\cdot\|_{\infty} \leq \alpha_0}(q). \end{aligned} \quad (11)$$

这里  $p$ ,  $q$  和  $\lambda$  是对偶变量。

## 模型的求解

将鞍点问题 (11) 转换为结构 (9), 令

$$\mathcal{X} = U \times V \times U, \quad \mathcal{Y} = V \times W \times U, \quad \mathcal{K} = \begin{pmatrix} \nabla_{\mu} & -I & 0 \\ 0 & \mathcal{E}_{\mu} & 0 \\ A & 0 & A \end{pmatrix},$$

并且

$$f(x) = \beta \|L\|_*,$$

$$g(y) = \langle B, \lambda \rangle + \frac{\|\lambda\|_F^2}{2} + \mathcal{J}_{\|\cdot\|_{\infty} \leq \alpha_1}(p) + \mathcal{J}_{\|\cdot\|_{\infty} \leq \alpha_0}(q).$$

于是模型 (11) 的求解过程如算法1的所示。

## 模型的求解

### 算法 1 二阶时空 TGV 和低秩分解模型的 Primal-Dual 算法

初始化:  $\sigma, \tau, S_0, L_0$ , 令  $L_0 = A^* B$ ,  $S_0 = 0$ ;

迭代: 根据以下步骤更新参数:

1.  $p^{n+1} = \mathcal{P}_{\alpha_1}(p^n + \sigma(\nabla_{\mu} \bar{S}^n - \bar{w}^n));$
2.  $q^{n+1} = \mathcal{P}_{\alpha_0}(q^n + \sigma \mathcal{E}_{\mu}(\bar{w}^n));$
3.  $\lambda^{n+1} = (\lambda^n + \sigma(A(\bar{L}^n + \bar{S}^n) - B))/(1 + \sigma);$
4.  $S^{n+1} = S^n - \tau(A^* \lambda^{n+1} - \text{div}_1 p^{n+1});$
5.  $w^{n+1} = w^n + \tau(\text{div}_2 q^{n+1} + p^{n+1});$
6.  $L^{n+1} = \mathcal{S}_{\beta}(L^n - \tau A^* r^{n+1});$
7.  $\bar{S}^{n+1} = 2S^{n+1} - S^n;$
8.  $\bar{w}^{n+1} = 2w^{n+1} - w^n;$
9.  $\bar{L}^{n+1} = 2L^{n+1} - L^n;$

直到收敛, 返回  $x^{n+1}$

## 模型的求解

当迭代步长满足  $\tau\sigma\|\mathcal{K}\|^2 < 1$  时，算法收敛，因此我们需要估计算子  $\mathcal{K}$  的模。经计算可得：

$$\|\mathcal{K}\|^2 \leq \frac{2\|A\|^2 + 25 + \sqrt{(2\|A\|^2 - 1)^2 + 48}}{2}.$$

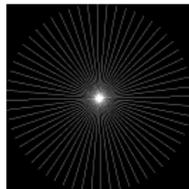
## 数值试验设置

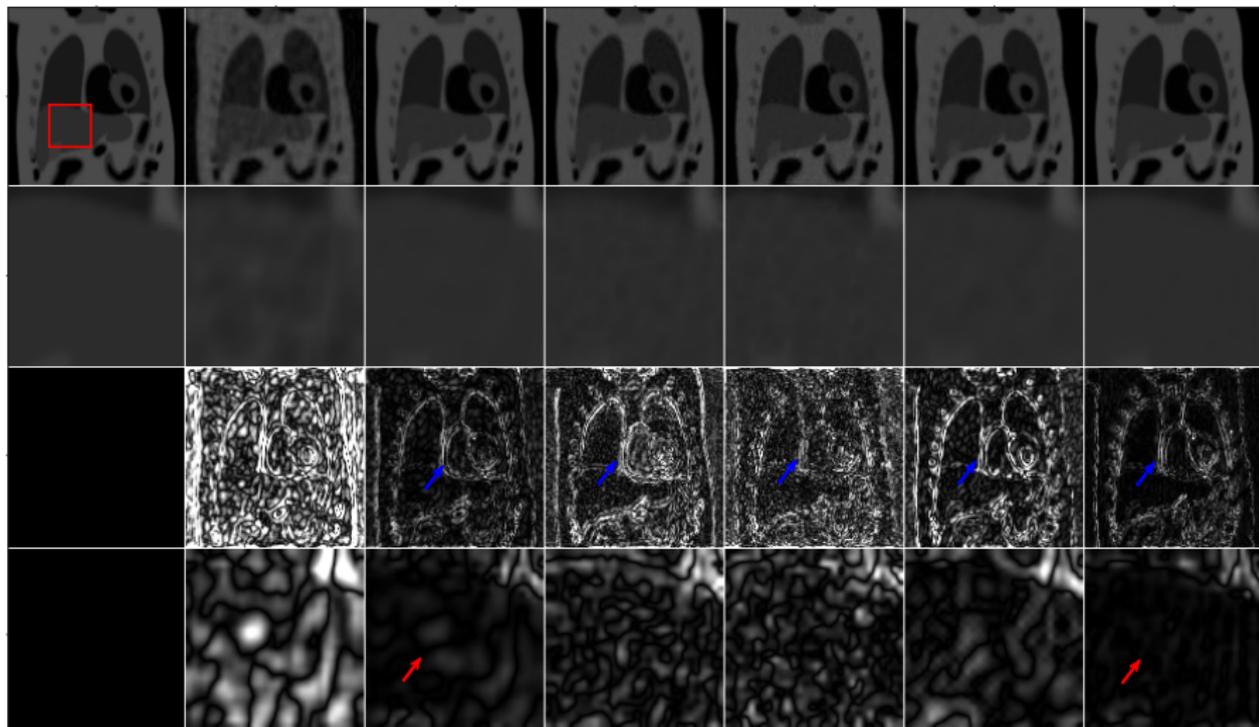
- ▶ 实验数据：躯干体模数据、心脏灌注、三组胸部 DCE-MRI
- ▶ 对比模型：kt-SLR, kt-RPCA, L+S, ICTGV
- ▶ 采样模式：Cartesian 采样、伪径向采样
- ▶ 评价指标：SER, SSIM, 像素时间曲线
- ▶ 实验程序：MATLAB 和 CUDA C 程序均已开源，程序链接<https://github.com/chixindebaoyu/tgvnn>

## 数值实验结果

Table: 各个模型在不同数据上的重建结果（伪径向采样）

数据集		躯干体模	心脏灌注	胸部 1	胸部 2	胸部 3
Zerofilled	SER	20.54	14.80	11.32	11.57	14.85
	SSIM	0.8321	0.8855	0.4820	0.5897	0.7086
kt-SLR	SER	<b>33.35</b>	17.58	17.42	13.61	18.38
	SSIM	0.9880	0.9412	0.7712	0.6956	0.8461
kt-RPCA	SER	29.27	18.33	19.31	14.82	19.81
	SSIM	0.9700	0.9447	0.8857	0.7920	0.9068
L+S	SER	27.99	19.12	17.60	14.74	19.91
	SSIM	0.9514	0.9490	0.7675	0.7319	0.8791
ICTGV	SER	26.88	17.87	16.31	12.18	16.84
	SSIM	0.9435	0.9405	0.6735	0.6080	0.7693
Proposed	SER	32.74	<b>19.57</b>	<b>20.56</b>	<b>16.24</b>	<b>21.08</b>
	SSIM	<b>0.9917</b>	<b>0.9514</b>	<b>0.9402</b>	<b>0.9091</b>	<b>0.9356</b>





Original

Zerofilled

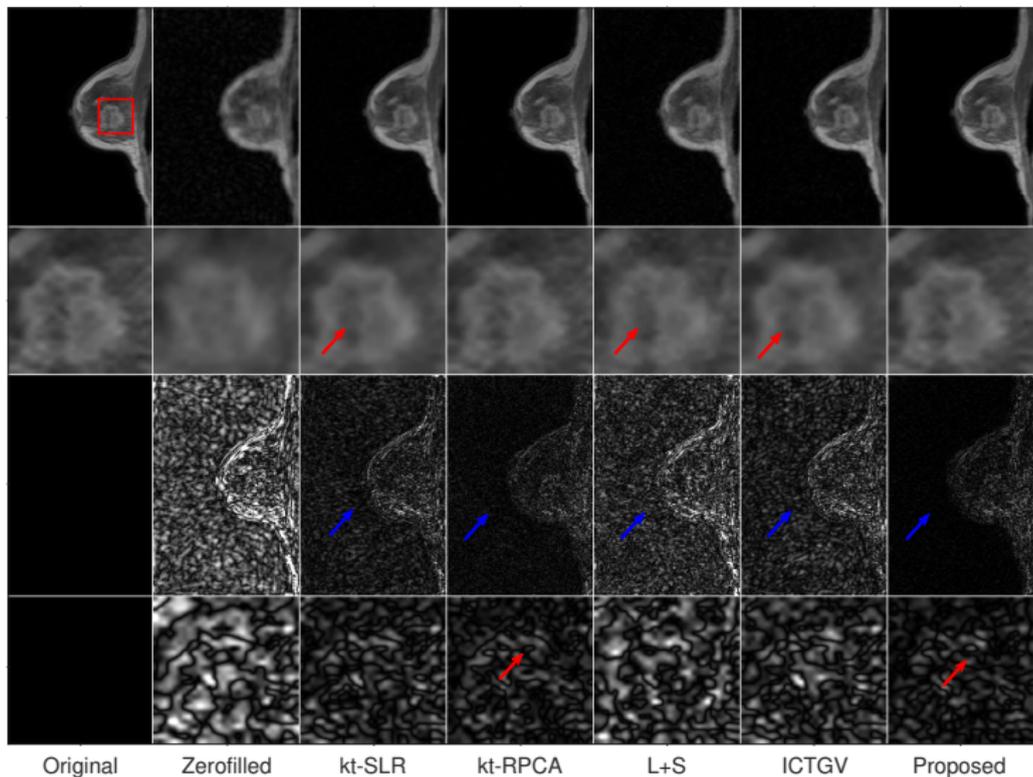
kt-SLR

kt-RPCA

L+S

ICTGV

Proposed



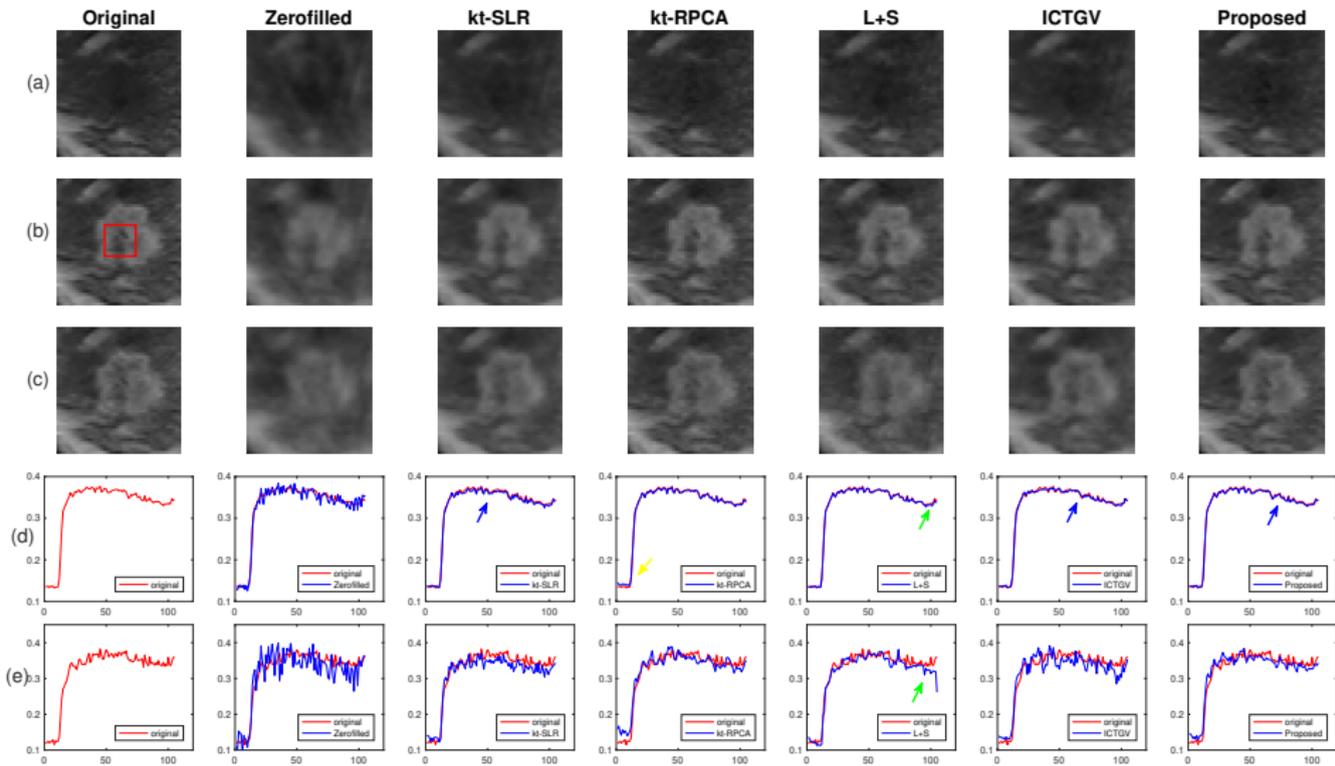


Table: 各个模型在不同下采样率下在胸部 1 数据上的重建结果

模型 \ 采样线		12	22	32	42	52
Zerofilled	SER	5.98	9.06	11.32	12.93	14.33
	SSIM	0.3218	0.4108	0.4820	0.5350	0.5818
kt-SLR	SER	15.16	16.10	17.42	19.07	19.11
	SSIM	0.7323	0.7312	0.7712	0.8087	0.7989
kt-RPCA	SER	13.10	17.74	19.31	20.26	21.13
	SSIM	0.6241	0.8356	0.8857	0.9061	0.9245
L+S	SER	12.75	16.00	17.60	18.73	19.64
	SSIM	0.5885	0.7119	0.7675	0.8018	0.8275
ICTGV	SER	13.56	15.30	16.31	17.03	17.56
	SSIM	0.6001	0.6485	0.6735	0.6895	0.7007
Proposed	SER	<b>16.49</b>	<b>19.06</b>	<b>20.56</b>	<b>21.82</b>	<b>22.96</b>
	SSIM	<b>0.8620</b>	<b>0.9119</b>	<b>0.9402</b>	<b>0.9535</b>	<b>0.9632</b>

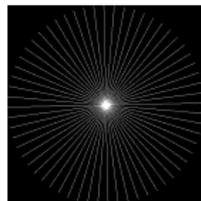


Table: 各个模型在胸部 1 数据上的重建结果 (Cartesian 采样)

模型	Zerofilled	kt-SLR	kt-RPCA	L+S	ICTGV	Proposed
SER	11.29	16.60	16.86	14.55	14.33	<b>17.79</b>
SSIM	0.7370	0.8942	0.9126	0.8379	0.7681	<b>0.9169</b>

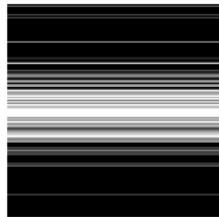


Table: 各个模型在胸部 1 数据上运行时间比较

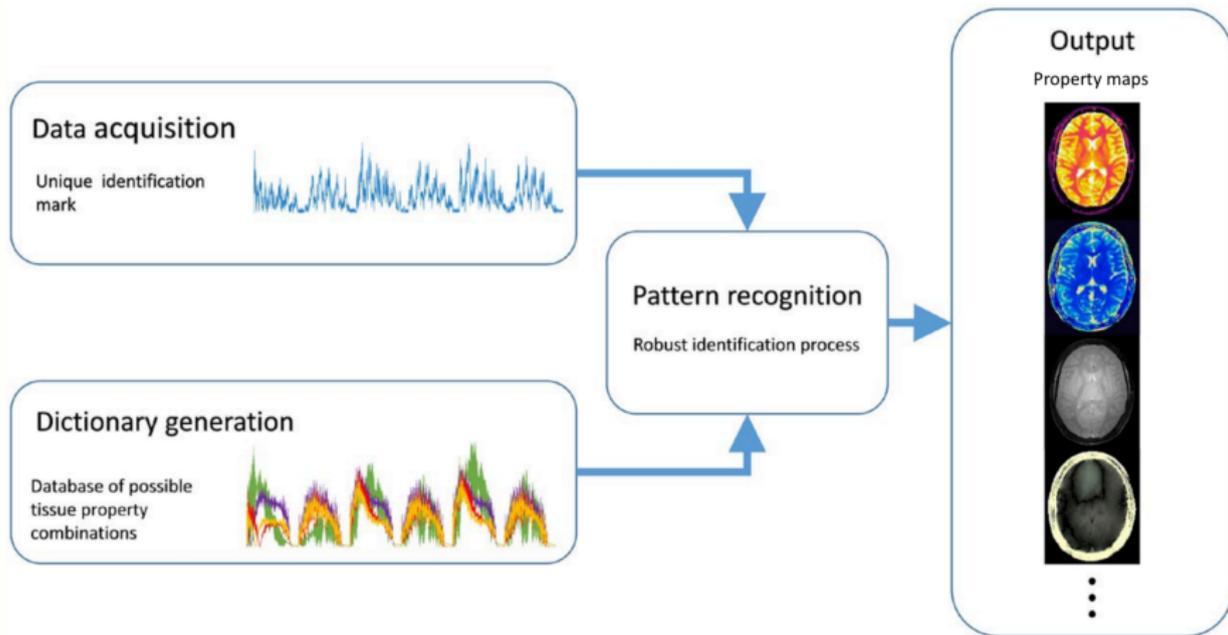
模型	kt-SLR (CPU)	kt-RPCA (CPU)	L+S (CPU)	ICTGV (CPU)	Proposed (CPU)	Proposed (GPU)
时间 (s)	5794.88	641.71	493.80	2572.04	2812.90	<b>28.35</b>

## 总结

- ▶ 利用图像分解思想，提出了针对动态 MR 图像的基于 **TGV 与核范数** 的重建模型
- ▶ 使用 **CUDA C** 在 GPU 上对 MATLAB 程序加速，**提速 100 倍**
- ▶ 数值试验表明在**不同的采样模式和采样率下**，提出的模型都比最前沿的模型在**消除空间伪影、保持边缘**上的表现更好，尤其是在**胸部 DCE-MRI 图像**上
- ▶ **Dong Wang, David S. Smith, Xiaoping Yang. Dynamic MR Image Reconstruction using TGV and Low-rank Decomposition. *Magnetic Resonance in Medicine*. 2020; 83(6); 2064-2076.**

- ① 基于广义全变分和低秩分解的动态 MR 图像重建
- ② 基于图形处理单元的实时 MRF 字典生成与匹配算法

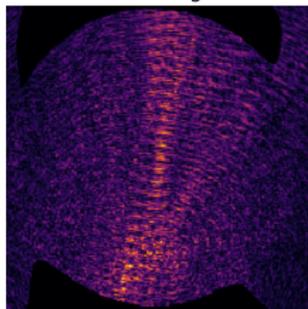
磁共振指纹 (MRF) 是一种新的定量 MRI 方法，可以在单次数据采集同时获取多个组织参数，如  $T_1$ ， $T_2$  和质子密度



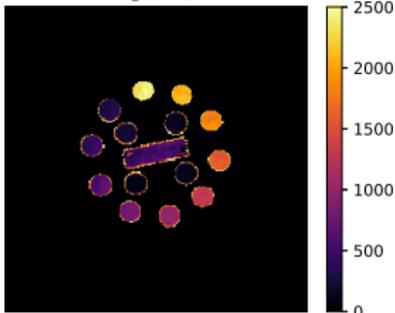
MRF 重建参数图的过程中涉及到三个步骤，分别为信号采集、预定义字典生成和模板匹配。

- ▶ **信号采集：**选取合适的 MR 序列对信号进行采样，并且 MR 序列的参数，如偏转角 ( $\alpha$ )、回波时间 ( $T_E$ )、重复时间 ( $T_R$ ) 等，需要随着时间随机变化，使得不同参数 (如  $T_1$ 、 $T_2$ ) 的组织在 MR 序列中产生独特的信号演化 (指纹)。
- ▶ **字典生成：**字典中的元素包含着不同参数的组织在该 MR 序列中的模拟演化。
- ▶ **模式识别：**将采集到的信号与字典中的元素进行匹配，重建参数图。

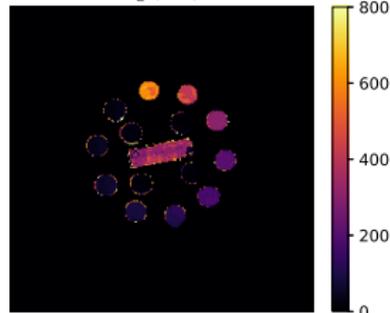
Raw Image



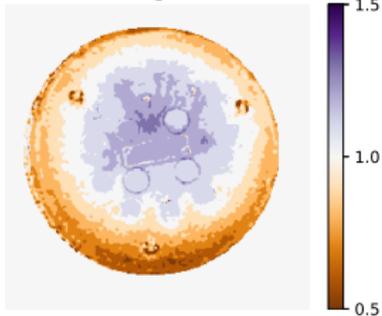
$T_1$  (ms)



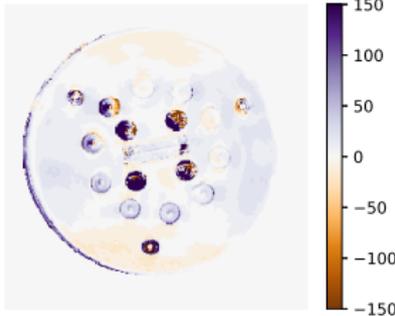
$T_2$  (ms)



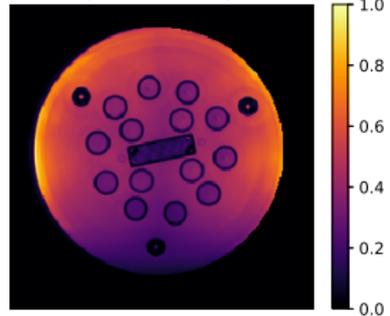
$B_1^+$

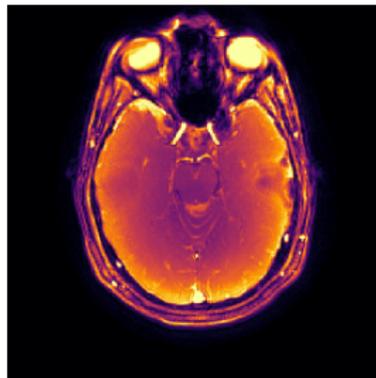
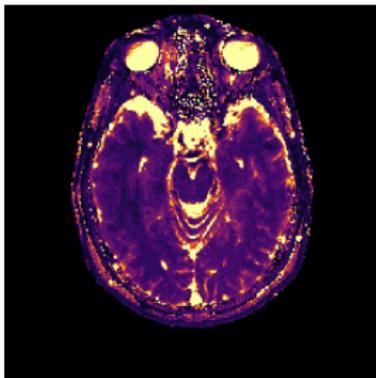
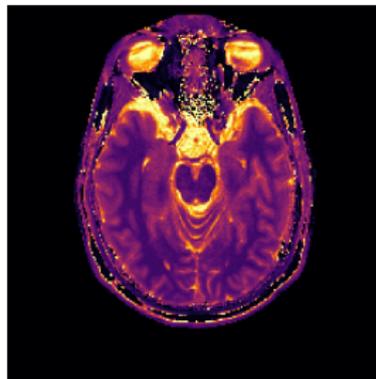
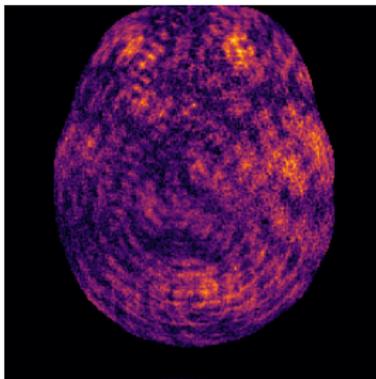


off-resonance (Hz)



proton density





## MRF 的主要问题:

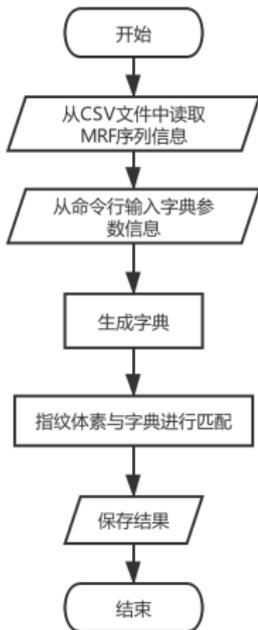
- ▶ 字典生成: 扩展相图 (EPG) 模型**十分耗时且复杂!**  $\Rightarrow$  梯度场、射频场、驰豫等
- ▶ 匹配算法: 模板匹配算法**十分耗时!**  $\Rightarrow O(NL^2K)$   
 $X = \{x_n \in \mathbb{C}^L, n = 1, \dots, N\}$  为指纹数据,  $D = \{d_k \in \mathbb{C}^L, k = 1, \dots, K\}$  为生成的字典,  $L$  为时间点的个数。模板匹配算法为:

$$\hat{k}_n = \underset{k}{\operatorname{argmax}} |\langle d_k, x_n \rangle|, \quad \hat{\rho}_n = \left| \langle d_{\hat{k}_n}, x_n \rangle \right|$$

## 图形处理单元

- ▶ 图形处理单元 (GPU) 逐渐成为并行加速科学计算的主流方法, 在医学成像领域有着广泛的应用
- ▶ CUDA 是一种通用并行计算架构, 可以使 GPU 能够高效地解决复杂的科学计算问题
- ▶ **首次**在 CUDA 的架构下, 提出了基于 GPU 的 MRF 字典生成与模板匹配的并行加速算法 snapMRF
- ▶ snapMRF 程序链接  
<https://github.com/chixindebaoyu/snapMRF>

## snapMRF 算法流程



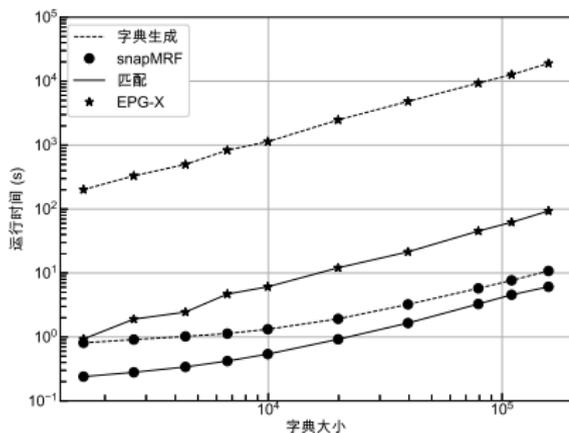
### 算法2 snapMRF生成字典与模板匹配算法流程

输入: \*d\_mrf, \*d\_params, \*d\_img

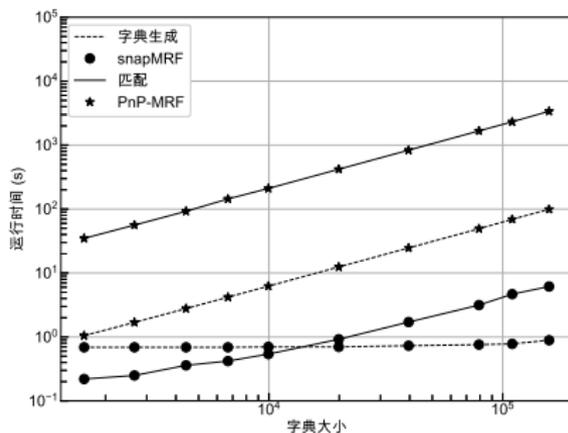
输出: \*d\_atoms, \*d\_maps

- 01: 从CSV文件中读取MRF序列信息, 存入 \*d\_mrf;
- 02: 从命令行输入字典参数信息, 存入 \*d\_params;
- 03: 初始化状态矩阵 \*d\_w;
- 04: **迭代:** 从第1个时刻到第L个时刻, 并行计算字典中所有元素
- 05: 使用函数 fill\_transition\_matrix() 构造转移矩阵;
- 06: 使用函数 apply\_rf\_pulse() 将射频场作用在 \*d\_w上;
- 07: 使用函数 decay\_signal() 将  $T_1$  和  $T_2$  衰减作用在 \*d\_w上;
- 08: 使用函数 save\_atoms() 将元素的信号保存在 \*d\_atoms中;
- 09: 使用函数 dephase\_gradients() 将梯度场作用在 \*d\_w上;
- 10: 使用函数 decay\_signal() 将  $T_1$  和  $T_2$  衰减作用在 \*d\_w上;
- 11: **终止迭代;**
- 12: 释放 \*d\_w;
- 13: 从 RawArray 文件中读取指纹数据, 存入 \*d\_img;
- 14: 计算剩余显存大小, 并根据剩余显存, 将 \*d\_img分为G组;
- 15: **迭代:** 从第1组到第G组, 在每一组内并行计算所有体素的参数
- 16: 使用函数 MRF\_minimatch() 进行匹配;
- 17: 使用函数 generate\_maps() 生成参数图;
- 18: **终止迭代;**
- 19: 将 \*d\_atoms 和 \*d\_maps 保存为 RawArray 文件;
- 20: 释放所有显存和内存。

## 数值实验结果



snapMRF vs EPG-X



snapMRF vs PnP-MRF

Table: snapMRF 与 EPG-X 在字典生成和模板匹配上运行时间的比较

运行时间 (s)	EPG-X 固定 $T_R$	snapMRF 固定 $T_R$	snapMRF 变化 $T_R$	snapMRF 变化 $T_R+B_1^+$
体模/字典生成	17797.05	11.00	7.42	9.39
体模/模板匹配	137.13	5.97	4.14	4.88
脑部/字典生成	18629.82	11.29	7.63	8.72
脑部/模板匹配	143.55	6.13	4.23	4.63

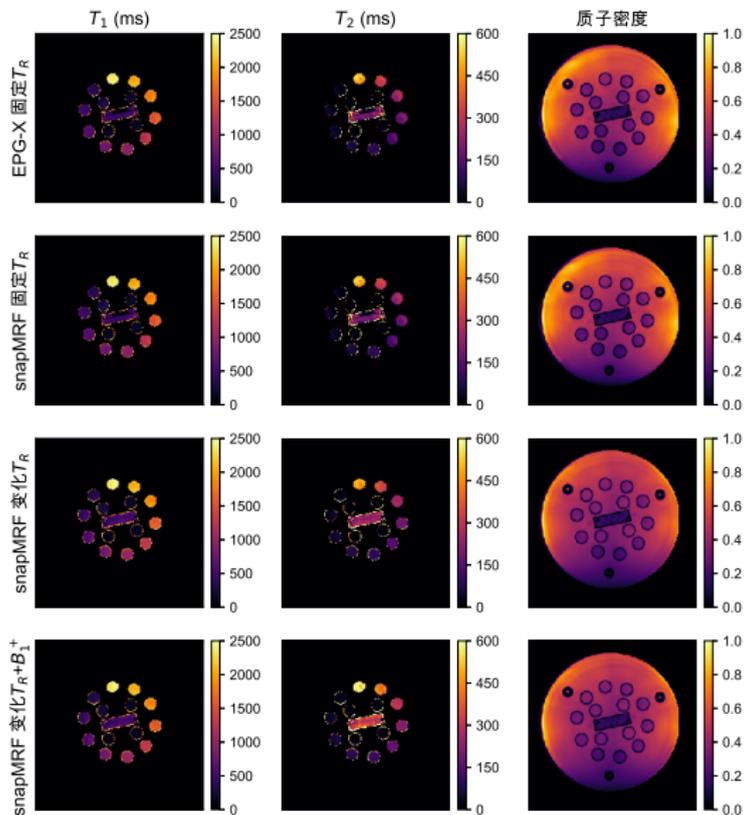
- ▶ 字典大小  $1,000 \times 100,000$
- ▶ 指纹大小  $1,000 \times 240 \times 240$

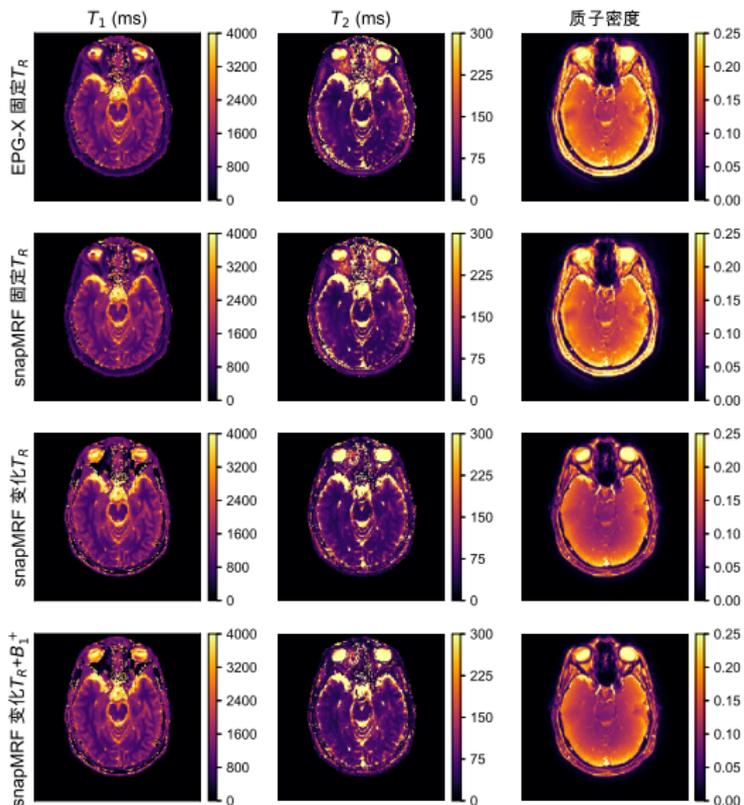
Table:  $T_1$  参数图的准确性比较

真实 $T_1$ (ms)	EPG-X 固定 $T_R$	snapMRF 固定 $T_R$	snapMRF 变化 $T_R$	snapMRF 变化 $T_R+B_1^+$
90.9	128.5	127.7	111.5	94.2
126.9	155.0	155.0	146.5	127.9
176.6	173.1	173.1	172.3	153.8
244.2	280.4	280.4	265.0	225.0
336.5	342.7	342.7	326.5	319.2
458.4	471.2	471.2	471.9	468.3
608.6	602.7	601.2	625.0	622.1
801.7	771.5	770.4	818.5	813.5
1044.0	945.0	943.1	1032.3	1026.0
1332.0	1262.7	1263.1	1310.8	1306.7
1604.0	1568.8	1568.1	1607.3	1593.3
1907.0	1861.2	1861.9	1854.2	1828.8
2173.0	2043.1	2043.1	2091.9	2094.2
2480.0	2366.5	2366.2	2434.6	2416.3
err (%)	4.9	5.0	2.6	3.0

Table:  $T_2$  参数图的准确性比较

真实 $T_2$ (ms)	EPG-X 固定 $T_R$	snapMRF 固定 $T_R$	snapMRF 变化 $T_R$	snapMRF 变化 $T_R+B_1^+$
5.6	6.9	6.9	9.4	12.3
7.9	11.5	11.2	10.0	13.5
11.2	13.3	13.3	11.2	13.5
15.8	13.7	13.5	11.3	20.4
22.6	21.2	21.2	23.3	30.0
32.0	32.3	32.3	38.3	47.3
46.4	45.0	44.8	50.4	60.0
64.1	64.2	64.2	70.2	83.8
96.9	84.6	84.4	90.8	104.6
133.3	144.0	143.8	146.9	170.8
190.9	175.4	175.4	185.2	213.8
278.1	266.5	266.5	255.4	290.0
403.5	323.3	323.5	343.7	407.7
581.3	474.0	474.2	453.5	531.5
err (%)	16.9	16.9	17.9	9.3





## 总结

- ▶ 针对 MRF 中字典生成与模板匹配速度慢的问题，提出了基于 GPU 的算法 snapMRF
- ▶ 字典生成的速度提高了 100-10000 倍，模板匹配的速度提高了 10-100 倍
- ▶ snapMRF 可以处理多种 MR 序列，生成的参数图精度更高
- ▶ **Dong Wang, Jason Ostenson, David S. Smith. snapMRF: GPU-accelerated magnetic resonance fingerprinting dictionary generation and matching using extended phase graphs. *Magnetic Resonance Imaging*. 2020; 66; 248-256.**

谢谢大家!