Dynamic MR Image Reconstruction with BART

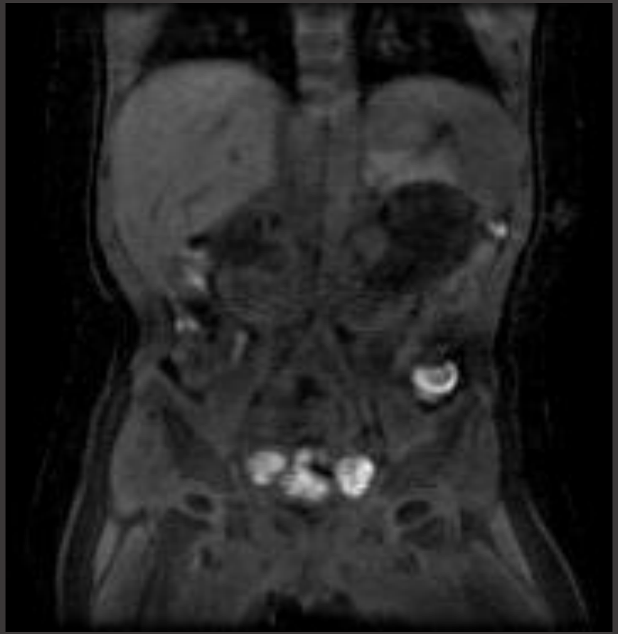
Efrat Shimron[1], Jon Tamir[2]

[1]UC Berkeley, [2]UT Austin

Introduction – Dynamic MRI

- Dynamic MRI – k-space data is acquired over time.
- Examples:
  - Dynamic Contrast Enhanced (DCE) MRI
  - Cardiac MRI (CINE)

Tutorial's focus:
Reconstruction of Dynamic 2D+Time data using BART



Pediatric DCE scan, Stanford

Outline

1. BART Data Format & Dimensionality
2. Pipeline Overview
3. Pre-processing:
   - Sensitivity maps computation
4. Reconstruction of 2D+Time data using Compressed Sensing with Low Rank Regularization

Additional slides

1. Reconstruction of 2D+Time data using different Regularizations:
   - L1-wavelet
   - Total Variation (TV)
   - Global Low Rank (LR)
2. Intro to 3D+Time data reconstruction
3. K-space normalization

BART's Data Format

Each dataset contains two files: ".cfl" and ".hdr" files

```
ls
```

```
ksp.cfl  ksp.hdr  maps.cfl  maps.hdr
```

K-space data   Sensitivity maps

BART's commands operate on pairs of .cfl & .hdr files

```
bart show -m ksp
```

Data Dimensionality – DCE Example

View the data dimensions:

Example: 2D+T multi-coil data

```
bart show -m ksp
```

```
Type: complex float
Dimensions: 16
AoD:    1      68      180     20      1       1       1       1       1       1       18      1       1       1
```

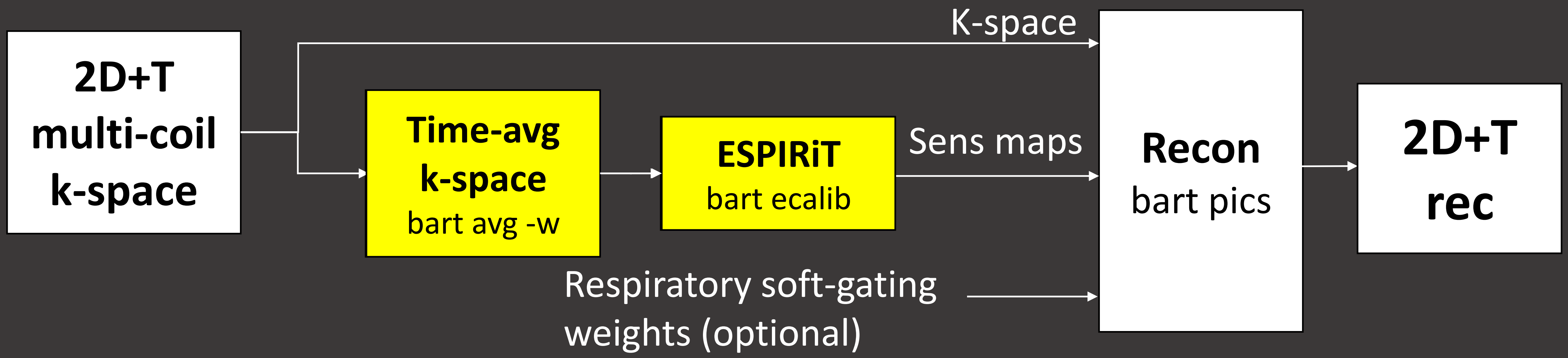
Single slice in x dim

y, z

20 coils

18 Time frames

Pipeline Overview



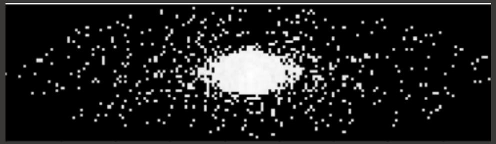
Cheng et al. 2013; Zhang et al., 2015; Uecker et al., 2015

Sensitivity Maps Computation

1. Compute the time-average of k-space data (bart avg):

```
bart avg -w $(bart bitmask 10) data/ksp  data/ksp_t_avg
```

“-w” flag: do not include unsampled pixels



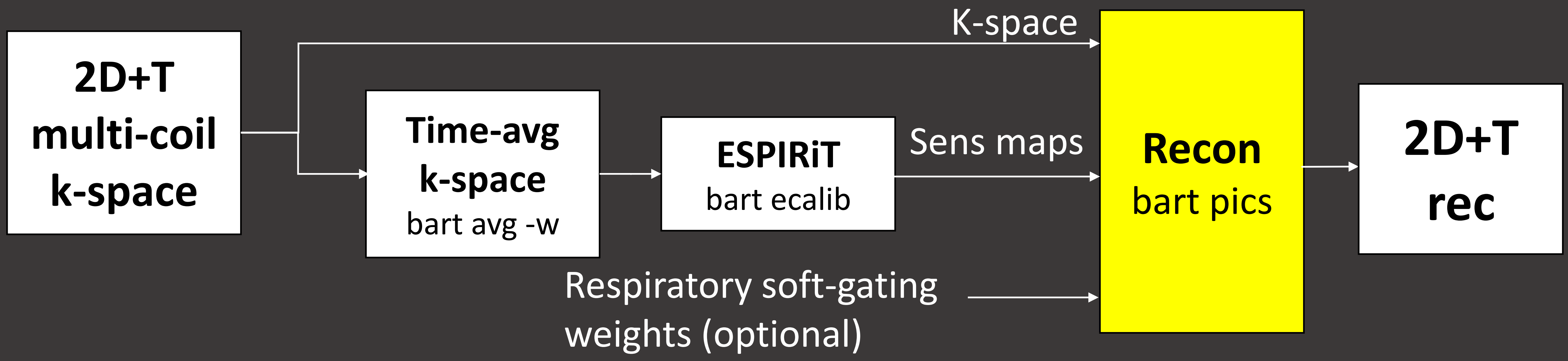
Specify the time dimension (zero indexing)

2. Compute ESPIRiT sensitivity maps from the time-averaged data:

```
bart ecalib -a data/ksp_t_avg  data/maps
```

Uecker et al., ESPIRiT, 2014; Iyer et al. MRM 2020

Pipeline Overview

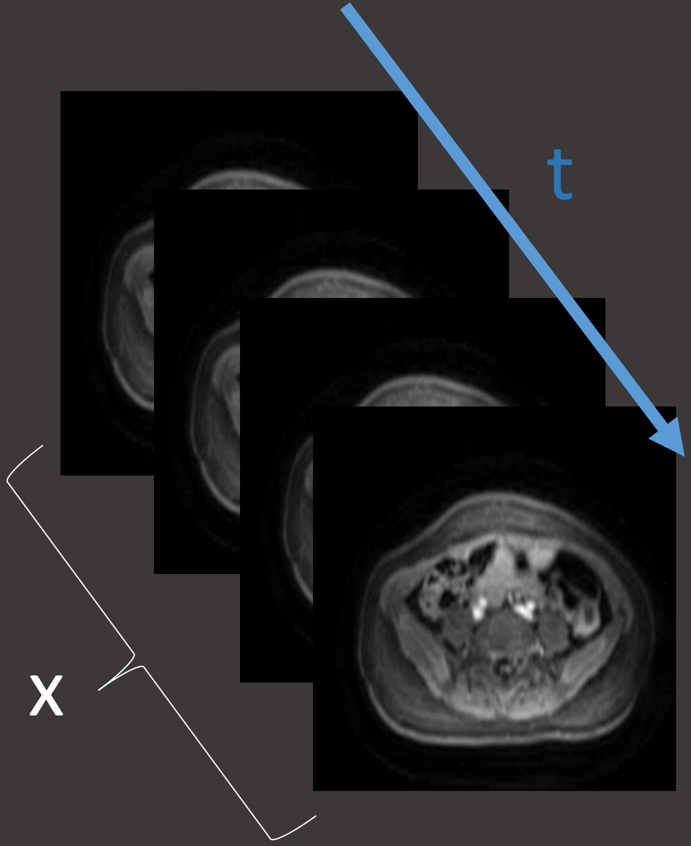


Cheng et al. 2013; Zhang et al., 2015; Uecker et al., 2015

Reconstruction

The Compressed Sensing problem:

$$x = \arg\min_{x} \underbrace{\left\| Ax - y \right\|_2^2}_{\text{data consistency}} + \underbrace{\sum_k \lambda_k R_k(x)}_{\text{regularization}}$$



In Dynamic MRI: x = a series of images

Lustig et al. (2006); Candés, Romberg, Tao (2006); Donoho (2006)

Reconstruction

In BART, Compressed Sensing reconstruction is implemented using the command “bart pics”

```
bart pics
```

How to get help:

```
bart pics -h
```

“bart pics” command – general structure

```
bart pics .....   <kspace data filename> <sens maps filename> <output filename>
```

Flags

Input 1

Kspace data

Input 2

Sensitivity maps

Output (recon)

.cfl files

“bart pics” command – general structure

```
bart pics .....   <kspace data filename>  <sens maps filename>  <output filename>
```

“-R” flag: determines the regularizer in the objective function

$$x = \arg\min_{x} \|Ax - y\|_2^2 + \sum_k \lambda_k R_k(x)$$

Get help:

```
bart pics -Rh
```

```
-R <T>:A:B:C     <T> is regularization type (single letter),
                 A is transform flags, B is joint threshold flags,
                 and C is regularization value. Specify any number
                 of regularization terms.

-R Q:C           l2-norm in image domain
-R I:B:C         l1-norm in image domain
-R W:A:B:C       l1-wavelet
-R T:A:B:C       total variation
-R T:7:0:.01     3D isotropic total variation with 0.01 regularization.
-R L:7:7:.02     Locally low rank with spatial decimation and 0.02 regularization.
-R M:7:7:.03     Multi-scale low rank with spatial decimation and 0.03 regularization.
```

See additional slides

Today’s example

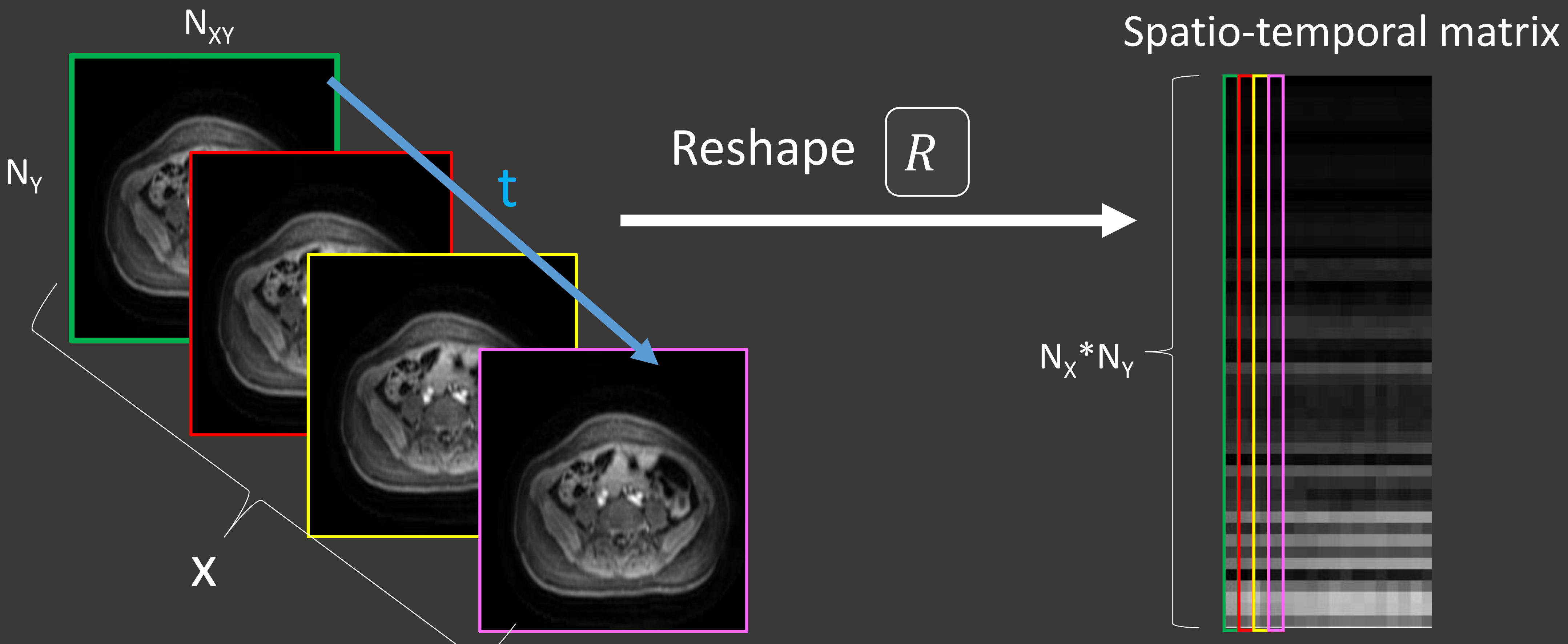
Regularization Example: Low Rank

- A dynamic series of images → Casorati matrix (spatio-temporal matrix)
- Each temporal frame → a column.

Haldar and Liang, 2011; Zhang et al., 2015; Ong and Lustig, 2016

Regularization Example: Low Rank

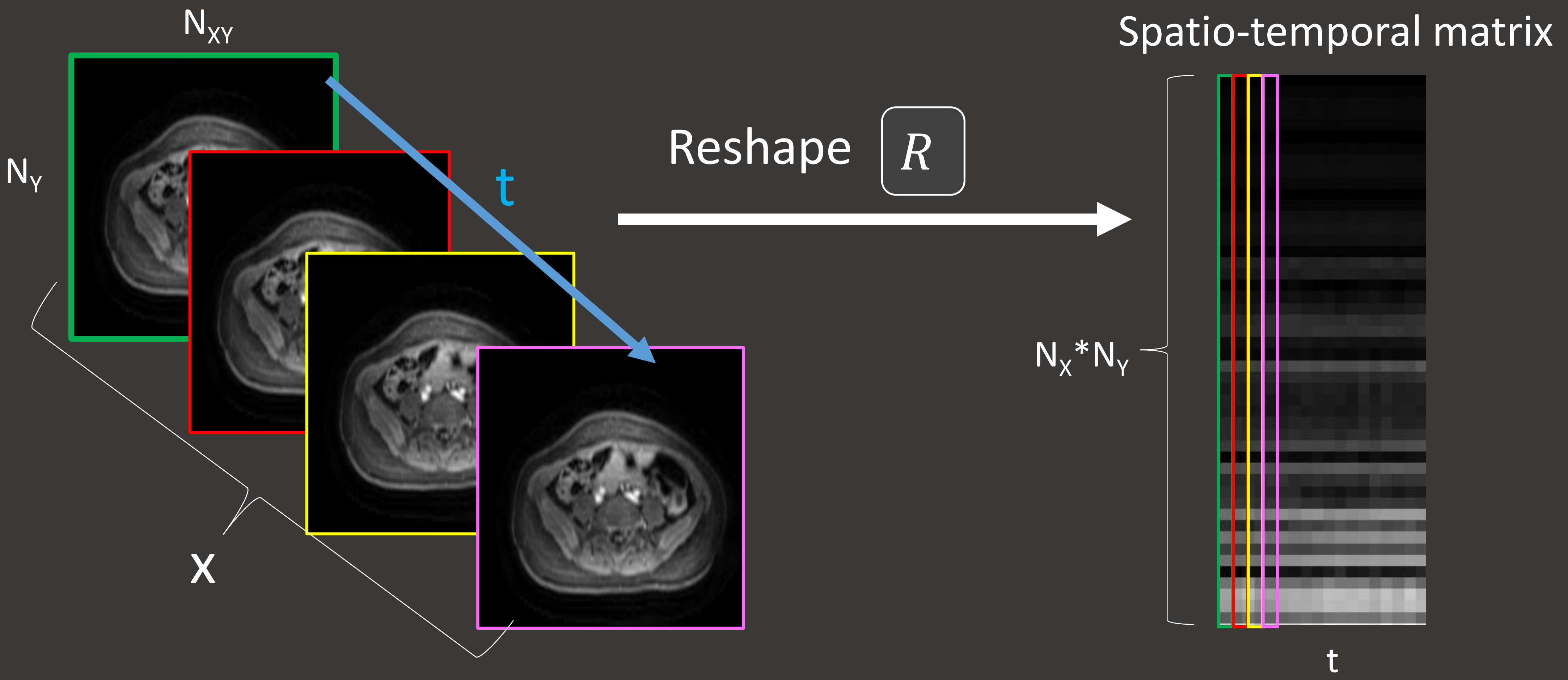
- A dynamic series of images → Casorati matrix (spatio-temporal matrix)
- Each temporal frame → a column.



Haldar and Liang, 2011; Zhang et al., 2015; Ong and Lustig, 2016

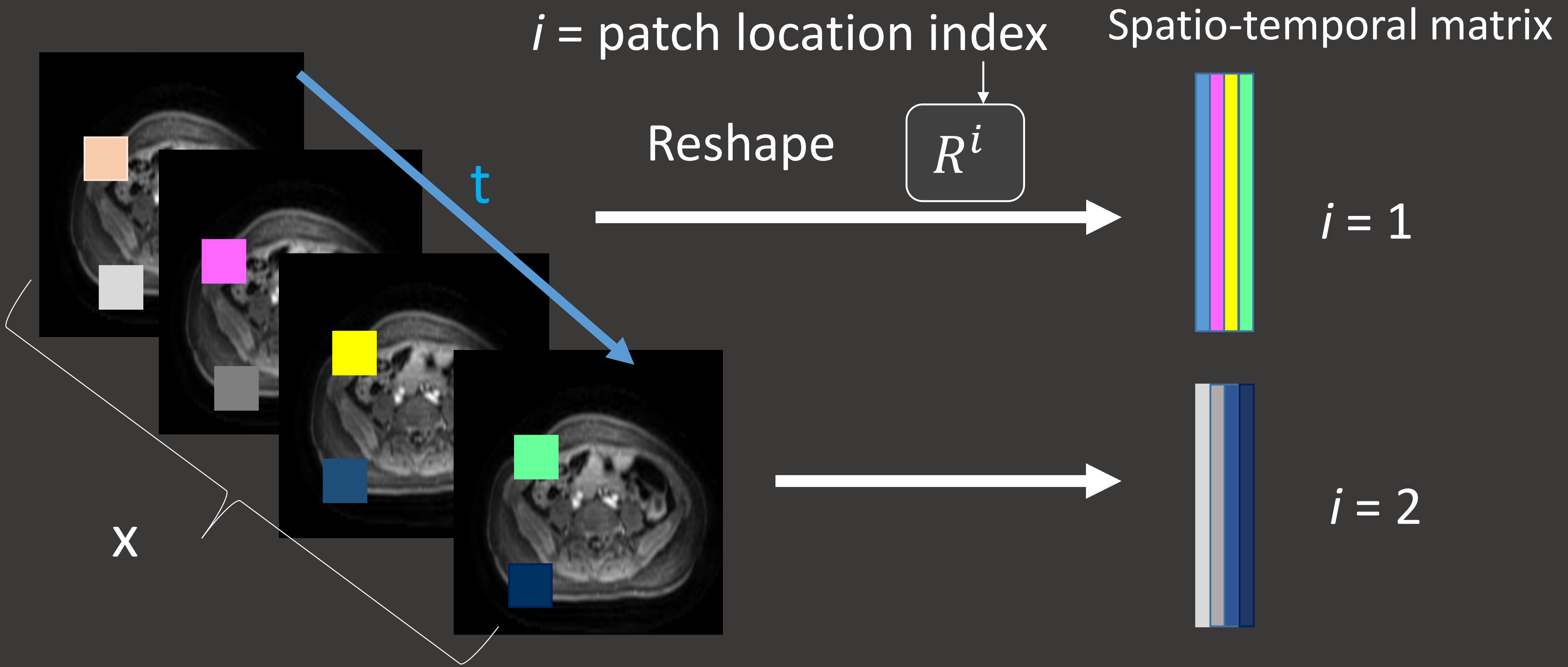
Regularization Example: Low Rank

Global Low Rank ⟷ Each column = a full time frame



Regularization Example: Low Rank

Locally Low Rank ⟷ Each column = reshaped **patch**



Regularization Example: Low Rank

The optimization problem:

$$x = \arg\min_x \|Ax - y\|_2^2 + \lambda \|Rx\|_*$$

Global Low-rank (GLR)

$$x = \arg\min_x \|Ax - y\|_2^2 + \lambda \sum_i \|R^i x\|_*$$

Locally Low Rank (LLR)

Rx

$\|x\|_*$ - Nuclear norm (sum of singular values of the matrix)

R - reshapes the data into the spatio-temporal matrix

R^i - extracts patch *i* and reshapes it into a small Casorati matrix

Haldar and Liang, 2011; Zhang et al., 2015; Ong and Lustig, 2016

Regularization Example: Low Rank

$$x = \arg\min_x \|Ax - y\|_2^2 + \lambda \underbrace{\sum_i \|R^i x\|_*}_{\text{LR regularizer}}$$

```
bart pics -R L:$(bart bitmask 0 1 2):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

R = regularization flag

L = LR

Dimension to be in included in the first dimension of Casorati matrix

GLR / LLR (next slide)

$\lambda = 0.01$

Regularization Example: Low Rank

Example #1 for the bitmasks:

```
bart pics -R L:$(bart bitmask 0 1 2):0:0.01 ksp maps recon_LLR
```

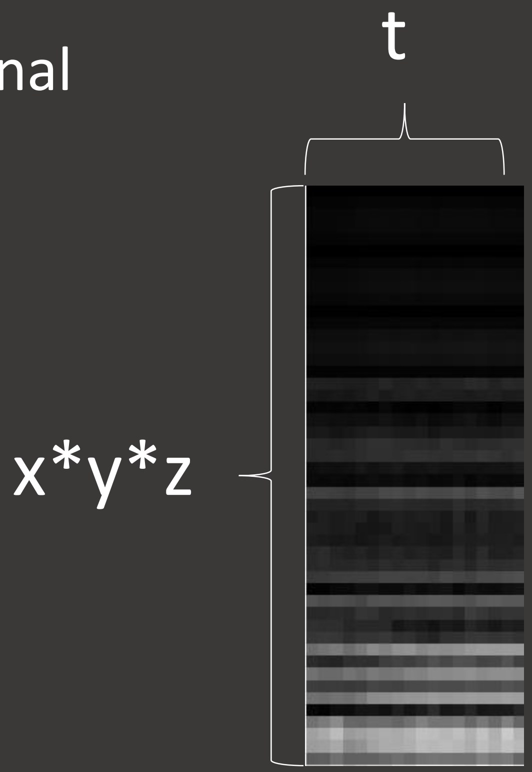
First dim of the Casorati matrix
x,y,z (0,1,2) in this case

All other dimensions will appear in the second dimension

t in this case

Which dimensions (of the original data!) will be split to patches.

0 → do not split to patches → Global LR



Regularization Example: Low Rank

Example #2 for the bitmasks:

```
bart pics -R L:$(bart bitmask 0 1 2):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

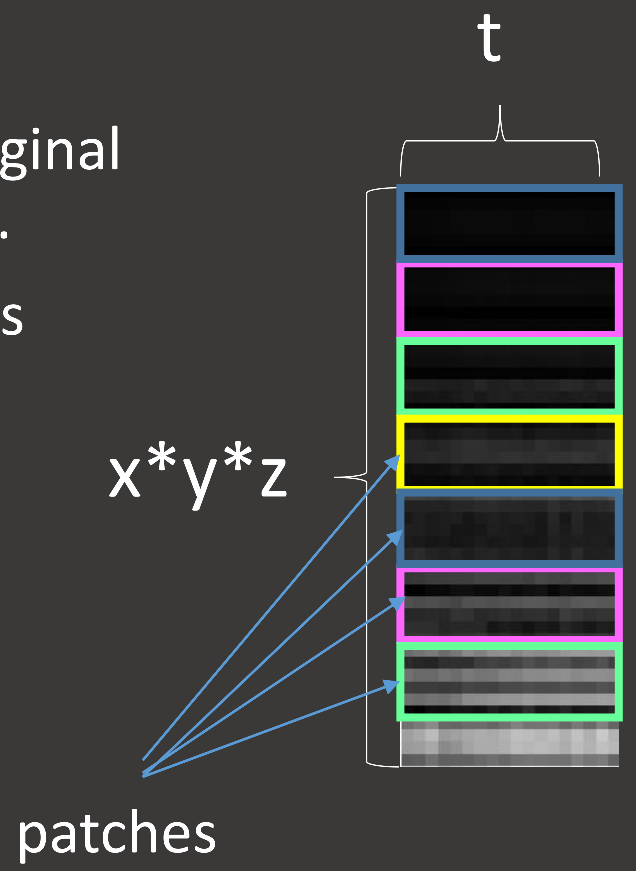
First dim of the Casorati matrix
x,y,z (0,1,2) in this case

All other dimensions will appear in the second dimension

t in this case

Which dimensions (of the original data!) will be split to patches.

x,y,z (0,1,2) → split to patches along these dims → LLR



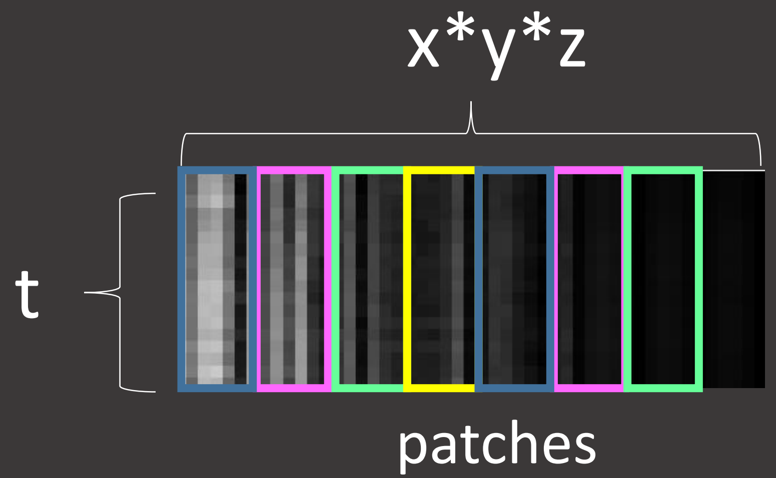
Regularization Example: Low Rank

Example #3 for the bitmasks:

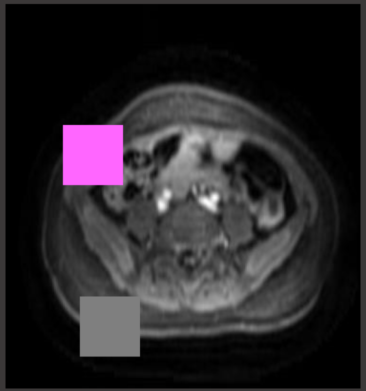
```
bart pics -R L:$(bart bitmask 10):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

Here the 1st bitmask specifies the *time* dimension

2nd bitmask is x,y,z (0,1,2) → split to patches along these dims → LLR



Regularization Example: Low Rank

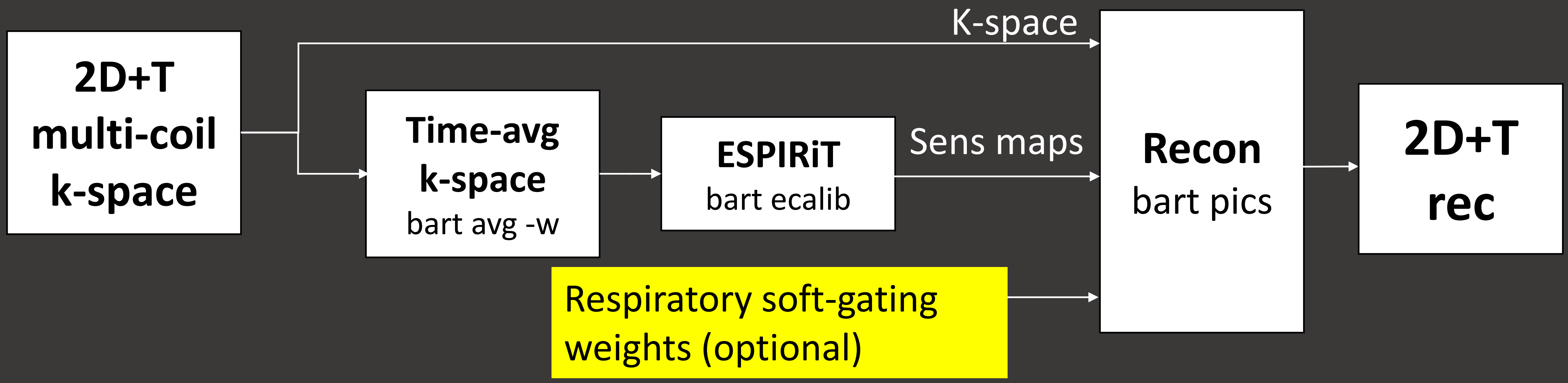


- By default, BART's blocks are of size 8x8
- The block (patch) size can be specified using "-b":

```
bart pics -b 20 -R L:$(bart bitmask 0 1 2):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

"-b" = block size
20x20 in this case

Pipeline Overview

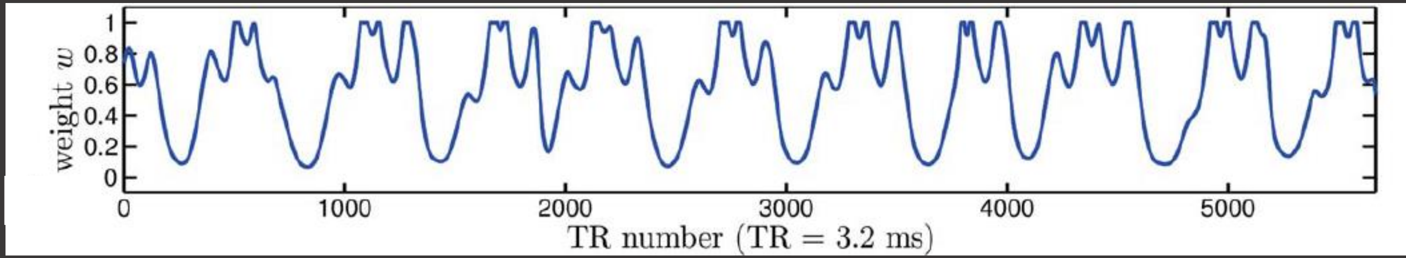


Cheng et al. 2013; Zhang et al., 2015; Uecker et al., 2015

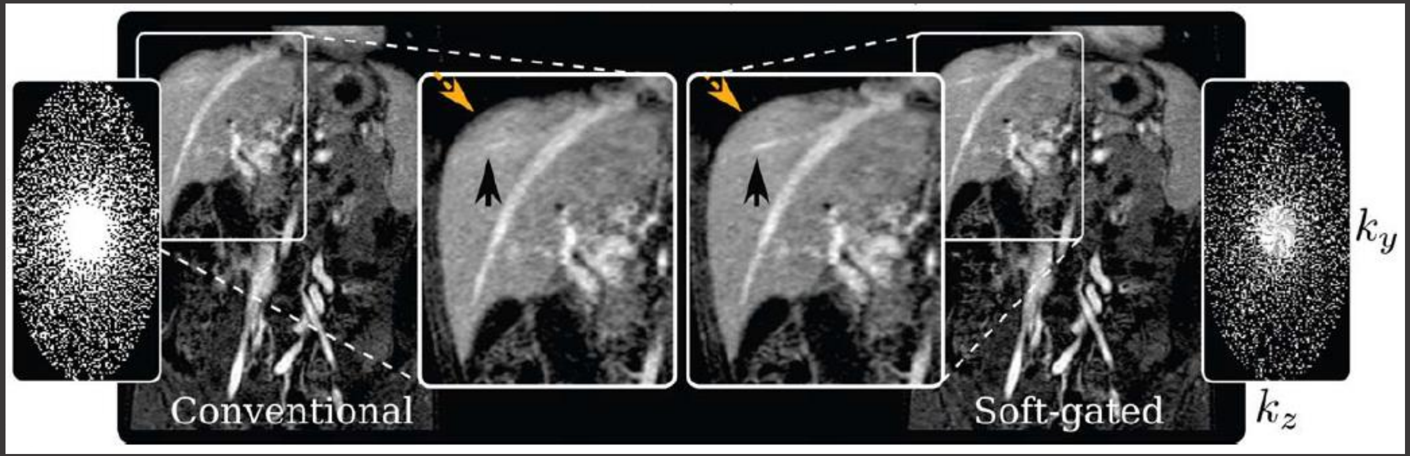
Soft-gating Weights - Mitigation of Motion Artifacts

```
bart pics -p weights_filename ....
```

$$x = \arg\min_x \left\| W(Ax - y) \right\|_2^2 + \sum_k \lambda_k R_k(x)$$



Weights



Johnson et al., MRM 2012; Cheng et al. 2013; Zhang et al., 2015

Summary

Take home message

BART pics is useful for Compressed Sensing reconstruction of 2D+Time data with:

- Global Low Rank (GLR) regularization:

```
bart pics -R L:$(bart bitmask 0 1 2):0:0.01 ksp maps recon_LLR
```

- Locally Low Rank (LLR) regularization:

```
bart pics -b 20 -R L:$(bart bitmask 0 1 2):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

Additional Slides

for independent reading

Outline

1. BART Data Format & Dimensionality
2. Pipeline Overview
3. Pre-processing:
    - Sensitivity maps computation
4. Reconstruction of 2D+Time data using Compressed Sensing with Low Rank Regularization

Additional slides

1. Reconstruction of 2D+Time data using different Regularizations:
    - L1-wavelet
    - Total Variation (TV)
    - Global Low Rank (LR)
2. Intro to 3D+Time data reconstruction
3. K-space normalization

Regularization Example 1: /1-wavelet in the spatial domain

$$x = \arg\min_{x} \underbrace{\|Ax - y\|_2^2}_{\text{data consistency}} + \underbrace{\lambda \|\Psi_{xyz} x\|_1}_{\text{L1-wavelet regularization}}$$

Ψ_{XYZ} - operator of a *spatial* wavelet transform.

This regularizer promotes sparsity in the wavelet domain

Lustig et al. (2006); Candés, Romberg, Tao (2006); Donoho (2006)

Regularization Example 1: /1-wavelet in the spatial domain

$$x = \arg\min_x \|Ax - y\|_2^2 + \underbrace{\lambda \|\Psi_{xyz} x\|_1}_{\text{L1-wavelet regularization}}$$

```
bart pics -R W:$(bart bitmask 0 1 2):0:0.001 ksp maps recon_l1wav
```

R = regularization flag

w = wavelet transform

1st bitmask: determines the dimensions along which the transform is applied. Here (0,1,2)=(x,y,z)

2nd bitmask promotes joint sparsity (see next slide)

λ=0.001

Lustig et al. (2006); Candés, Romberg, Tao (2006); Donoho (2006)

Regularization Example 1:
l1-wavelet in the spatial domain

Options for the 2nd bitmask:

Zero → Enforces sparsity of each wavelet image *separately.*

```
bart pics -R W:$(bart bitmask 0 1 2):0:0.001 ksp maps recon_l1wav
```

$$x = \arg\min_x \|Ax - y\|_2^2 + \lambda \|\Psi_{xyz} x\|_1$$

Non-zero → Enforces **joint sparsity** of the wavelet images along the specified dimension(s)

```
bart pics -R W:$(bart bitmask 0 1 2):$(bart bitmask 10):0.001 ksp maps recon_l1wav
```

$$x = \arg\min_x \|Ax - y\|_2^2 + \lambda \left\| \, \|\Psi_{xyz} x\|_{2,t} \, \right\|_1$$

Vasanawala et al., Radiology 2010; Murphy et al., IEEE TMI 2012

Regularization Example 2:
Total Variation (TV) along the temporal domain

$$x = \arg\min_x \underbrace{\|Ax - y\|_2^2}_{\text{data consistency}} + \underbrace{\lambda TV_t(x)}_{\text{TV regularization along the temporal domain}}$$

TV - finite differences

In dynamic MRI TV is applied along the *temporal domain.*

Lustig et al. (2006); Candés, Romberg, Tao (2006); Donoho (2006)

Regularization Example 2: Total Variation (TV) along the temporal domain

$$x = \arg\min_{x} \underbrace{\|Ax - y\|_2^2}_{\text{data consistency}} + \underbrace{\lambda TV_t(x)}_{\text{TV regularization along the temporal domain}}$$

```
bart pics -R T:$(bart bitmask 10):0:.04 ksp maps recon_tv
```

$\lambda = 0.04$

R = regularization flag

T = Total Variation Transform

Dimension(s) along which the transform is applied.
10 = temporal dimension in this example

Regularization Example 3: /1-wavelet in the spatial domain + TV in time

$$x = \arg\min_x \underbrace{\|Ax - y\|_2^2}_{\text{data consistency}} + \underbrace{\lambda_1 \|\Psi_{xyz} x\|_1}_{\text{L1-wavelet in space}} + \underbrace{\lambda_2 TV_t(x)}_{\text{TV in time}}$$

```
bart pics -R W:$(bart bitmask 0 1 2):0:0.001 -R T:$(bart bitmask 10):0:.04  ksp maps recon_l1wav_tv
```

Regularization Example 3:
*l*1-wavelet in the spatial domain + TV in time

$$x = \arg\min_x \underbrace{\|Ax - y\|_2^2}_{\text{data consistency}} + \lambda_1 \underbrace{\|\Psi_{xyz} x\|_1}_{\text{L1-wavelet in space}} + \lambda_2 \underbrace{TV_t(x)}_{\text{TV in time}}$$

```
bart pics -R W:$(bart bitmask 0 1 2):0:0.001 -R T:$(bart bitmask 10):0:.04  ksp maps recon_l1wav_tv
```

R = regularization flag

Needed before each regularization term

Regularization Example 3:
l1-wavelet in the spatial domain + TV in time

$$x = \underbrace{\arg\min_x \|Ax - y\|_2^2}_{\text{data consistency}} + \lambda_1 \underbrace{\|\Psi_{xyz} x\|_1}_{\text{L1-wavelet in space}} + \lambda_2 \underbrace{TV_t(x)}_{\text{TV in time}}$$

```
bart pics -R W:$(bart bitmask 0 1 2):0:0.001 -R T:$(bart bitmask 10):0:.04  ksp maps recon_l1wav_tv
```

λ_1 λ_2

Summary – BART pics Regularizers for Dynamic MRI

BART has several useful regularizers:

1. *l*1-wavelet in the spatial domain with joint sparsity of frames:

```
bart pics -R W:$(bart bitmask 0 1 2):$(bart bitmask 10):0.001 ksp maps recon_l1wav
```

2. Total Variation (TV) in time:

```
bart pics -R T:$(bart bitmask 10):0:.04 ksp maps recon_tv
```

3. Locally Low Rank in the spatio-temporal domain:

```
bart pics -R L:$(bart bitmask 0 1 2):$(bart bitmask 0 1 2):0.01 ksp maps recon_LLR
```

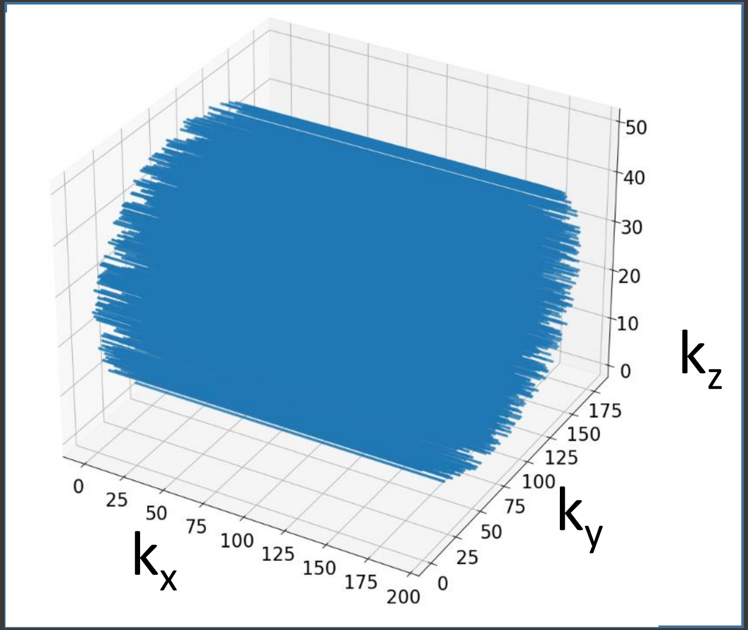
Intro to 3D+Time Reconstruction: Data Example

View the data dimensions: `bart show -m ksp3D`

```
Type: complex float
Dimensions: 16
AoD:    192    180    50    32    1    1    1    1    1    1    18    1    1
```

x, y, z   coils   Time frames

The data is usually fully sampled along one dimension (readout).



Outline

1. BART Data Format & Dimensionality
2. Pipeline Overview
3. Pre-processing:
   - Sensitivity maps computation
4. Reconstruction of 2D+Time data using Compressed Sensing with Low Rank Regularization

Additional slides

1. Reconstruction of 2D+Time data using different Regularizations:
   - L1-wavelet
   - Total Variation (TV)
   - Global Low Rank (LR)
2. Intro to 3D+Time data reconstruction
3. K-space normalization

Intro to 3D+Time Reconstruction: Data Example

A practical workflow:

1. Do IFFT along fully-sampled dimension.
2. Extract 2D+Time “slices”.
3. Reconstruct each 2D+T “slice” separately (as before..)
4. Build the reconsntructed 3D volume.

```
bart fft -u -i $(bart bitmask 0)   data/ksp3D   data/ksp_all_slices

bart slice 0 80  data/ksp_all_slices  data/ksp_slice_80
```

Intro to 3D+Time Reconstruction: Data Example

1. Do IFFT along fully-sampled dimension (the 0 dim in this example):

```
bart fft -u -i $(bart bitmask 0)   data/ksp3D   data/ksp_all_slices
```

Notice - the data dimensions didn't change – but its no longer a 3D k-space:

```
Type: complex float
Dimensions: 16
AoD:     192     180     50     32     1     1     1     1     1     1     18     1     1
```

2. Extract 2D+Time "slices" - example for slice at x=80:

```
bart slice 0 80  data/ksp_all_slices  data/ksp_slice_80
```

Slicing dimension

Slice position

Outline

1. BART Data Format & Dimensionality
2. Pipeline Overview
3. Pre-processing:
   - Sensitivity maps computation
4. Reconstruction of 2D+Time data using Compressed Sensing with Low Rank Regularization

Additional slides

1. Reconstruction of 2D+Time data using different Regularizations:
   - L1-wavelet
   - Total Variation (TV)
   - Global Low Rank (LR)
2. Intro to 3D+Time data reconstruction
3. K-space normalization

Intro to K-space Normalization

Motivation: normalize k-space to use similar λ values for different datasets.

Practical Method I:

- Ksp = ksp/max(abs(ksp))

Practical Method II:

1. Compute time-averaged k-space data for each coil
2. IFFT2 for each coil → images
3. Compute square Root Sum of Squares (RSS) over the coil images
4. Sort the RSS values
5. Scaling_factor = the value that is in location 0.95*len(RSS_vec)