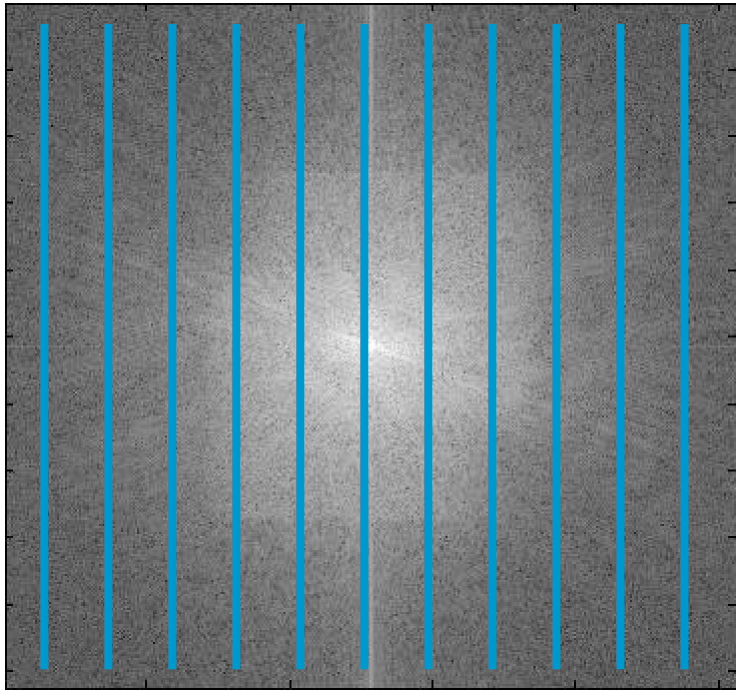
Repetition

Friedrich-Alexander-Universität
Medizinische Fakultät

Uniklinikum Erlangen

Partly k-space coverage: problems

Fourier transform

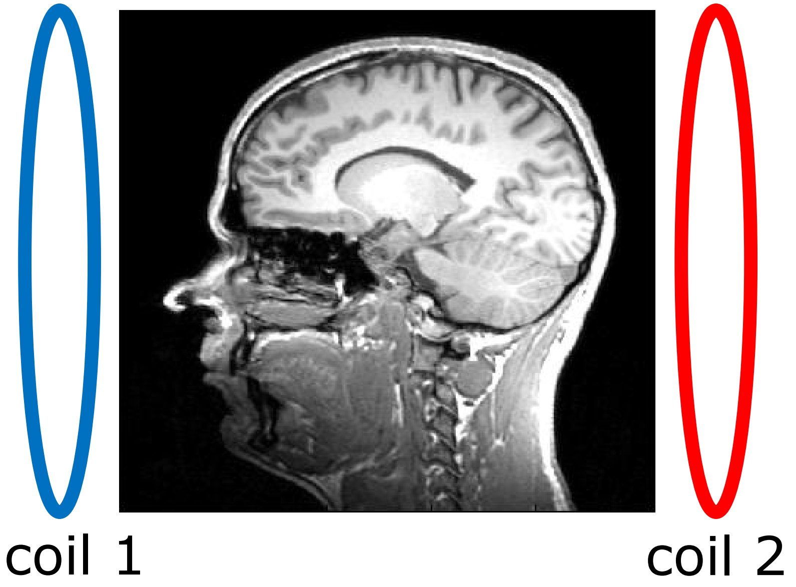


Leave out every other k-space line
Acceleration factor $R = 2$

E.g. acquisition time = 3 min
But: wrapping artifact

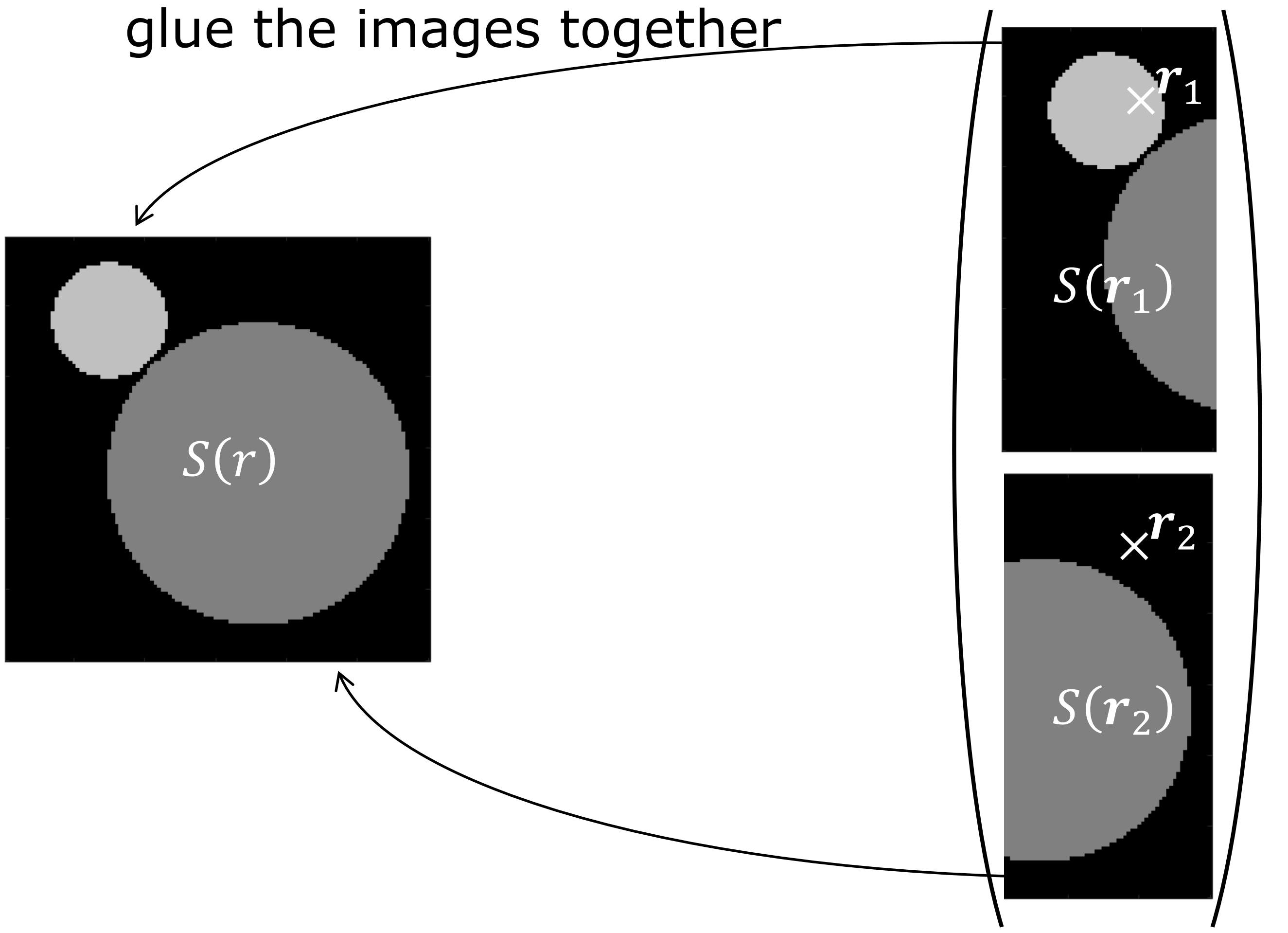
Parallel Imaging
Basic Principle

- Surface Coils:
  - Each coil receives signal from a limited part of the object
- Intrinsic spatial encoding



Numerical phantom: $N_{\text{coils}} = 2$ and $R = 2$

$$\begin{pmatrix} S_{\text{coil1}}(\boldsymbol{r}_1) \\ S_{\text{coil2}}(\boldsymbol{r}_1) \end{pmatrix} = \begin{pmatrix} \mathscr{s}_{\text{coil1}}(\boldsymbol{r}_1) & \mathscr{s}_{\text{coil1}}(\boldsymbol{r}_2) \\ \mathscr{s}_{\text{coil2}}(\boldsymbol{r}_1) & \mathscr{s}_{\text{coil2}}(\boldsymbol{r}_2) \end{pmatrix} \cdot \begin{pmatrix} S(\boldsymbol{r}_1) \\ S(\boldsymbol{r}_2) \end{pmatrix}$$

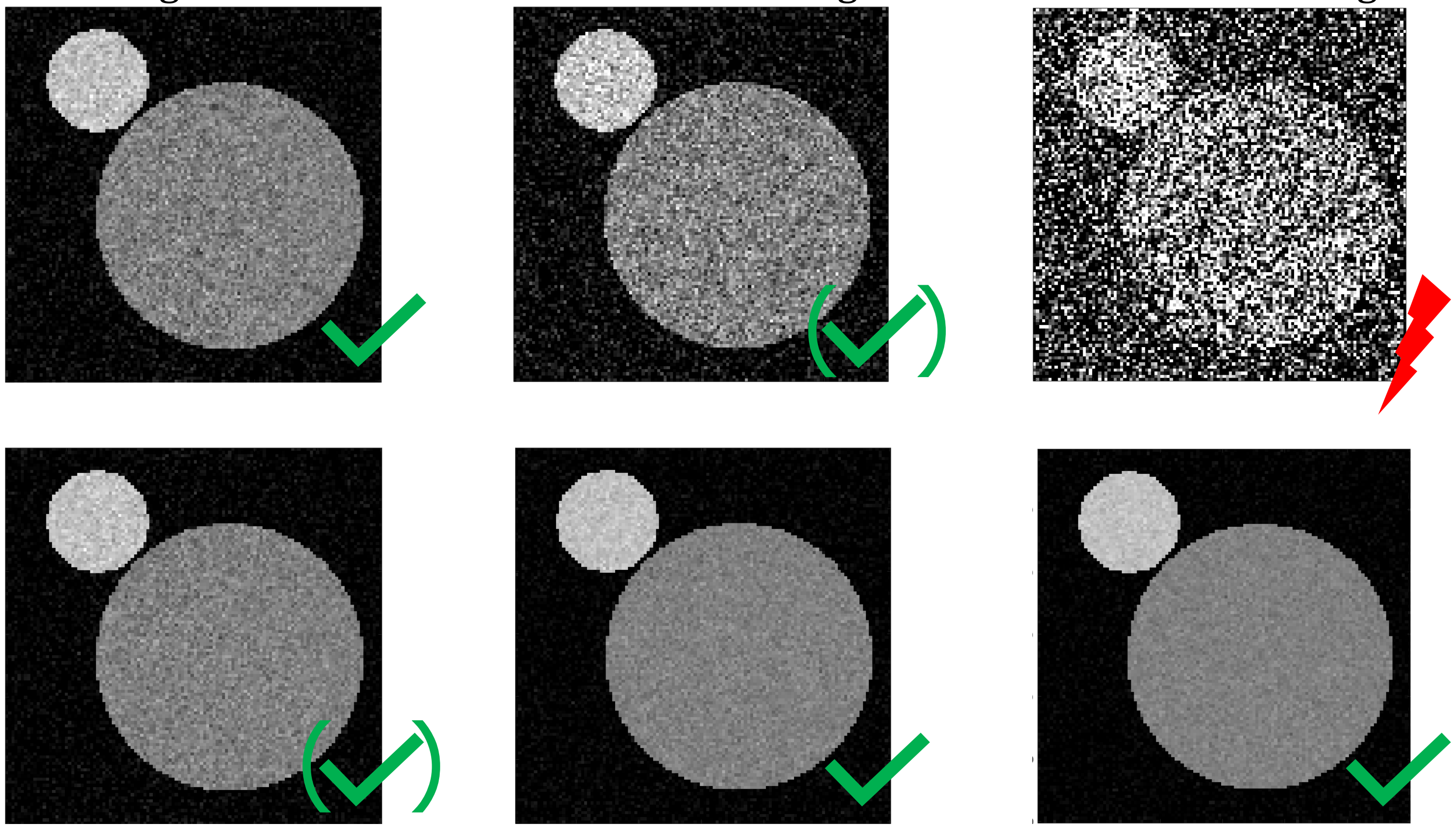


SENSE reconstruction with noise

s_{good} | $0.5 s_{\text{bad}} + 0.5 s_{\text{good}}$ | $0.9 s_{\text{bad}} + 0.1 s_{\text{good}}$

SENSE reconstruction

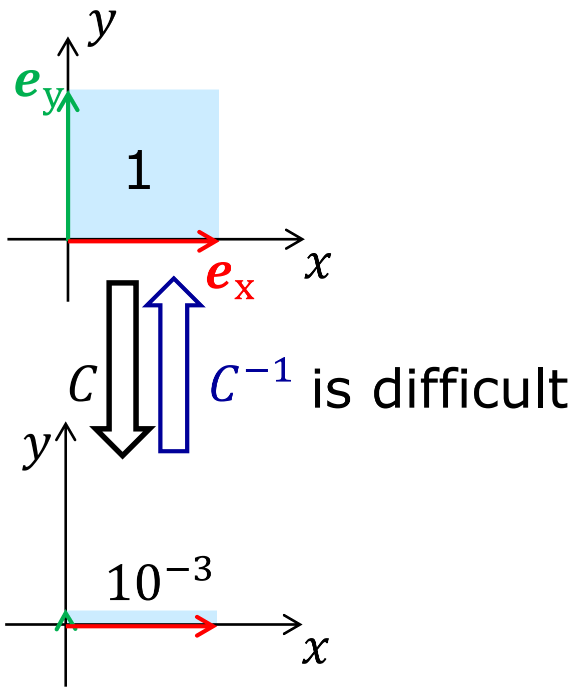
reconstruction with full sampling



Summary

Noise propagation depends on C

$$C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.001 \end{pmatrix}$$



- Ill-conditioned matrices:
  - Small determinants
  - Large matrix components C^{-1}

$$|C_1| = 10^{-3} \qquad C_1^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1{,}000 \end{pmatrix}$$

- Condition number:
  measure for noise propagation

$$\kappa(C) = \|C\| \cdot \|C^{-1}\|$$
$$\kappa(C_1) = 1{,}000$$

Summary

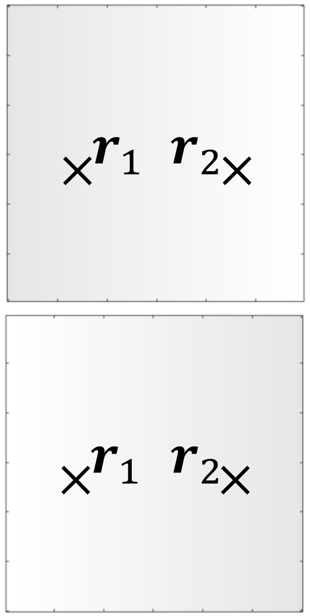
Explicit formula for noise propagation derived $SNR(\boldsymbol{r}_n) = \dfrac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$

Formula is quite abstract.

We learned how to apply it

coil 1

coil 2



$0.95 s_{\text{bad}} + 0.05 s_{\text{good}}$

$$C(\boldsymbol{r}_1) = \begin{pmatrix} 0.925 & 0.975 \\ 0.975 & 0.925 \end{pmatrix}$$

$$\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1} \approx \begin{pmatrix} \boxed{200} & -200 \\ -200 & \boxed{200} \end{pmatrix}$$

$$SNR(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma \cdot \sqrt{200}}$$

$$SNR(\boldsymbol{r}_2) = \frac{S(\boldsymbol{r}_2)}{\sigma \cdot \sqrt{200}}$$

g-factor

g-Factor: definition

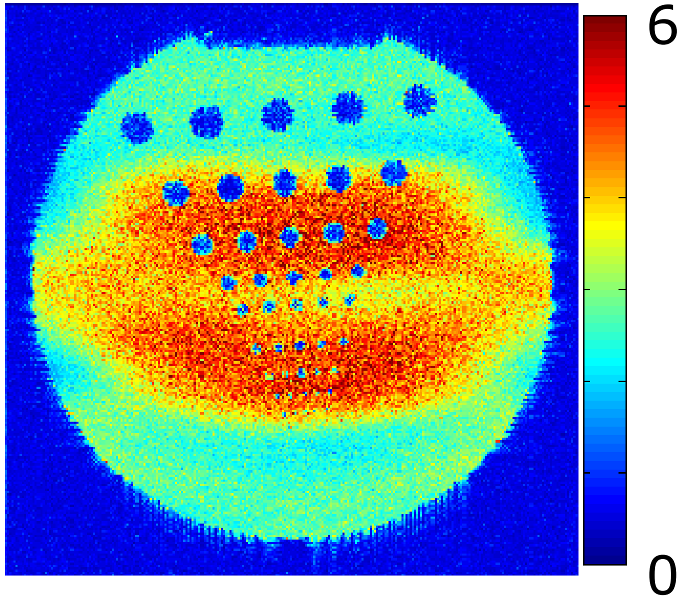
- g-factor = „grief factor“
- actually „geometry factor“

$$\text{SNR with } R\text{-fold accelerated SENSE} = \frac{\text{SNR without SENSE}}{\text{g-factor}}$$

- $g(\boldsymbol{r}) = \frac{SNR_{\mathrm{R=1}}(\boldsymbol{r})}{SNR_{\mathrm{R}}(\boldsymbol{r})}$

Phantom experiment
(12 channel coil, R=4)

g-factor



Where does this strange spatial dependence come from?

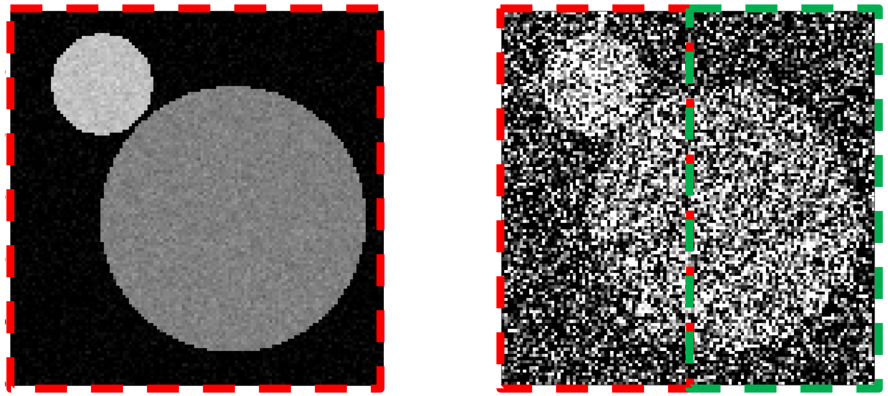
Recipe to compute $g(\boldsymbol{r})$:

Use definition $g(\boldsymbol{r}) = \frac{SNR_{\mathrm{R=1}}(\boldsymbol{r})}{SNR_{\mathrm{R}}(\boldsymbol{r})}$

Use our SNR formula $SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$

Simple in principle

Difficulty: different definition regimes of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$



SNR for $R = 1$

$$SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

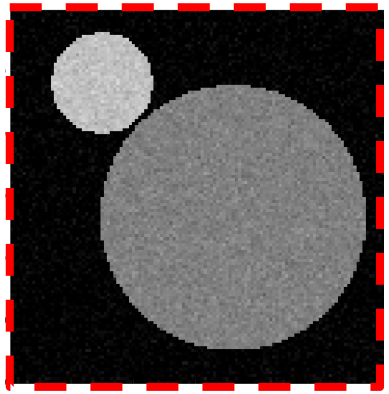
$$C_{\mathrm{R=1}}(\boldsymbol{r}_1) = \begin{pmatrix} s_{\mathrm{coil1}}(\boldsymbol{r}_1) \\ s_{\mathrm{coil2}}(\boldsymbol{r}_1) \end{pmatrix}$$

$$C^T_{\mathrm{R=1}}(\boldsymbol{r}_1) = \begin{pmatrix} s_{\mathrm{coil1}}(\boldsymbol{r}_1) & s_{\mathrm{coil2}}(\boldsymbol{r}_1) \end{pmatrix}$$

$$C^T_{\mathrm{R=1}}(\boldsymbol{r}_1)C_{\mathrm{R=1}}(\boldsymbol{r}_1) = \begin{pmatrix} s_{\mathrm{coil1}}(\boldsymbol{r}_1) & s_{\mathrm{coil2}}(\boldsymbol{r}_1) \end{pmatrix} \cdot \begin{pmatrix} s_{\mathrm{coil1}}(\boldsymbol{r}_1) \\ s_{\mathrm{coil2}}(\boldsymbol{r}_1) \end{pmatrix}$$
$$= s^2_{\mathrm{coil1}}(\boldsymbol{r}_1) + s^2_{\mathrm{coil2}}(\boldsymbol{r}_1)$$

$$\left(C^T_{\mathrm{R=1}}(\boldsymbol{r}_1)C_{\mathrm{R=1}}(\boldsymbol{r}_1)\right)^{-1} = \frac{1}{s^2_{\mathrm{coil1}}(\boldsymbol{r}_1) + s^2_{\mathrm{coil2}}(\boldsymbol{r}_1)}$$

$$SNR_{\mathrm{R=1}}(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma} \cdot \sqrt{s^2_{\mathrm{coil1}}(\boldsymbol{r}_1) + s^2_{\mathrm{coil2}}(\boldsymbol{r}_1)}$$

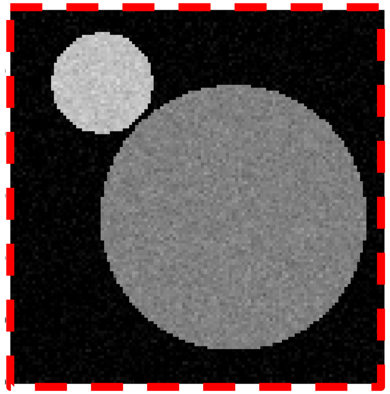


$S_{\mathrm{noise}}(\boldsymbol{r}_1)$

SNR for $R = 1$ and $R > 1$

$$g(\boldsymbol{r}) = \frac{SNR_{R=1}(\boldsymbol{r})}{SNR_R(\boldsymbol{r})} \qquad SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

$$SNR_{\mathrm{R=1}}(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_1) + s_2^2(\boldsymbol{r}_1)}$$

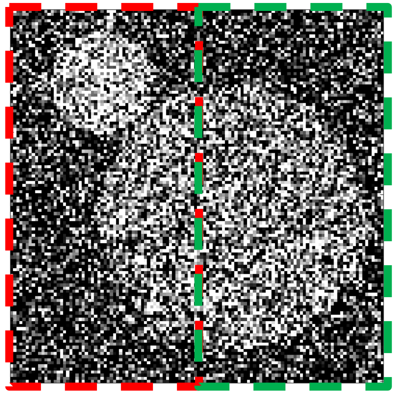


$S_{\mathrm{noise}}(\boldsymbol{r}_1)$

How to properly related $SNR_{\mathrm{R=1}}$ and $SNR_{\mathrm{R=2}}$?

$$SNR_{R=2}(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma \cdot \sqrt{\left[\left(C_{R=2}^T(\boldsymbol{r}_1)C_{\mathrm{R=2}}(\boldsymbol{r}_1)\right)^{-1}\right]_{1,1}}}$$

$$SNR_{R=2}(\boldsymbol{r}_2) = \frac{S(\boldsymbol{r}_2)}{\sigma \cdot \sqrt{\left[\left(C_{\mathrm{R=2}}^T(\boldsymbol{r}_1)C_{\mathrm{R=2}}(\boldsymbol{r}_1)\right)^{-1}\right]_{2,2}}}$$



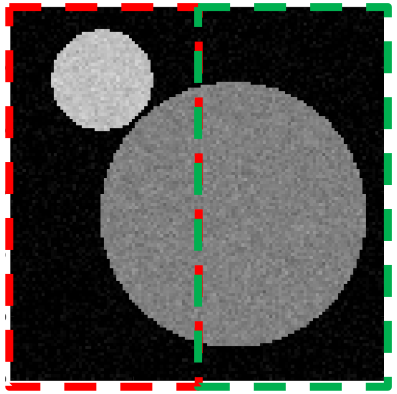
$S_{\mathrm{noise}}(\boldsymbol{r}_1)$ $S_{\mathrm{noise}}(\boldsymbol{r}_2)$

SNR for $R = 1$ and $R > 1$

$$g(\boldsymbol{r}) = \frac{SNR_{R=1}(\boldsymbol{r})}{SNR_R(\boldsymbol{r})} \qquad SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

$$SNR_{\mathrm{R=1}}(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_1) + s_2^2(\boldsymbol{r}_1)}$$

$$SNR_{\mathrm{R=1}}(\boldsymbol{r}_2) = \frac{S(\boldsymbol{r}_2)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_2) + s_2^2(\boldsymbol{r}_2)}$$

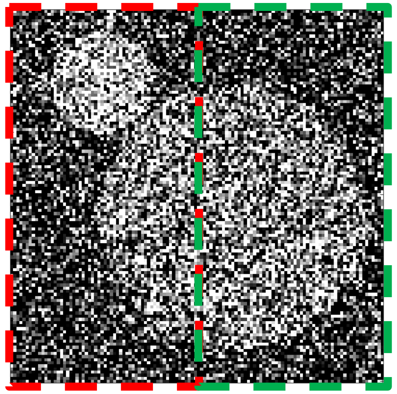


$S_{\mathrm{noise}}(\boldsymbol{r}_1)$ $S_{\mathrm{noise}}(\boldsymbol{r}_2)$

How to properly related $SNR_{\mathrm{R=1}}$ and $SNR_{\mathrm{R=2}}$?

$$SNR_{R=2}(\boldsymbol{r}_1) = \frac{S(\boldsymbol{r}_1)}{\sigma \cdot \sqrt{\left[\left(C_{R=2}^T(\boldsymbol{r}_1)C_{\mathrm{R=2}}(\boldsymbol{r}_1)\right)^{-1}\right]_{1,1}}}$$

$$SNR_{R=2}(\boldsymbol{r}_2) = \frac{S(\boldsymbol{r}_2)}{\sigma \cdot \sqrt{\left[\left(C_{\mathrm{R=2}}^T(\boldsymbol{r}_1)C_{\mathrm{R=2}}(\boldsymbol{r}_1)\right)^{-1}\right]_{2,2}}}$$



$S_{\mathrm{noise}}(\boldsymbol{r}_1)$ $S_{\mathrm{noise}}(\boldsymbol{r}_2)$

SNR for $R = 1$ and $R > 1$

$$g(\boldsymbol{r}) = \frac{SNR_{R=1}(\boldsymbol{r})}{SNR_R(\boldsymbol{r})} \qquad SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

$$\left.\begin{aligned} SNR_{\mathrm{R}=1}(\boldsymbol{r}_1) &= \frac{S(\boldsymbol{r}_1)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_1) + s_2^2(\boldsymbol{r}_1)} \\ SNR_{\mathrm{R}=1}(\boldsymbol{r}_2) &= \frac{S(\boldsymbol{r}_2)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_2) + s_2^2(\boldsymbol{r}_2)} \end{aligned}\right\} SNR_{\mathrm{R}=1}(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_n) + s_2^2(\boldsymbol{r}_n)}$$

$$\left.\begin{aligned} SNR_{R=2}(\boldsymbol{r}_1) &= \frac{S(\boldsymbol{r}_1)}{\sigma \cdot \sqrt{\left[\left(C_{\mathrm{R}=2}^T(\boldsymbol{r}_1)C_{\mathrm{R}=2}(\boldsymbol{r}_1)\right)^{-1}\right]_{1,1}}} \\ SNR_{R=2}(\boldsymbol{r}_2) &= \frac{S(\boldsymbol{r}_2)}{\sigma \cdot \sqrt{\left[\left(C_{\mathrm{R}=2}^T(\boldsymbol{r}_1)C_{\mathrm{R}=2}(\boldsymbol{r}_1)\right)^{-1}\right]_{2,2}}} \end{aligned}\right\} SNR_{\mathrm{R}=2}(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C^T(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

A formula for the g-factor

$$g(\boldsymbol{r}) = \frac{SNR_{R=1}(\boldsymbol{r})}{SNR_R(\boldsymbol{r})} \qquad SNR(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

$$SNR_{\mathrm{R=1}}(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma} \cdot \sqrt{s_1^2(\boldsymbol{r}_n) + s_2^2(\boldsymbol{r}_n)}$$

$$\rightarrow g(\boldsymbol{r}_n) = \frac{SNR_{R=1}(\boldsymbol{r}_n)}{SNR_R(\boldsymbol{r}_n)} = \sqrt{\left(s_1^2(\boldsymbol{r}_n) + s_2^2(\boldsymbol{r}_n)\right)\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

$$SNR_{\mathrm{R=2}}(\boldsymbol{r}_n) = \frac{S(\boldsymbol{r}_n)}{\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}}$$

Relating the g-factor to C

$$g(\boldsymbol{r}) = \sqrt{\left(s_1^2(\boldsymbol{r}_n) + s_2^2(\boldsymbol{r}_n)\right)\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

$$C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1) = \begin{pmatrix} s_{\text{coil1}}(\boldsymbol{r}_1) & s_{\text{coil2}}(\boldsymbol{r}_1) \\ s_{\text{coil1}}(\boldsymbol{r}_2) & s_{\text{coil2}}(\boldsymbol{r}_2) \end{pmatrix} \cdot \begin{pmatrix} s_{\text{coil1}}(\boldsymbol{r}_1) & s_{\text{coil1}}(\boldsymbol{r}_2) \\ s_{\text{coil2}}(\boldsymbol{r}_1) & s_{\text{coil2}}(\boldsymbol{r}_2) \end{pmatrix}$$

$$= \begin{pmatrix} s_{\text{coil1}}^2(\boldsymbol{r}_1) + s_{\text{coil2}}^2(\boldsymbol{r}_1) & \dots \\ \dots & s_{\text{coil1}}^2(\boldsymbol{r}_2) + s_{\text{coil2}}^2(\boldsymbol{r}_2) \end{pmatrix}$$

$$s_1^2(\boldsymbol{r}_1) + s_2^2(\boldsymbol{r}_1) = [C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{1,1}$$

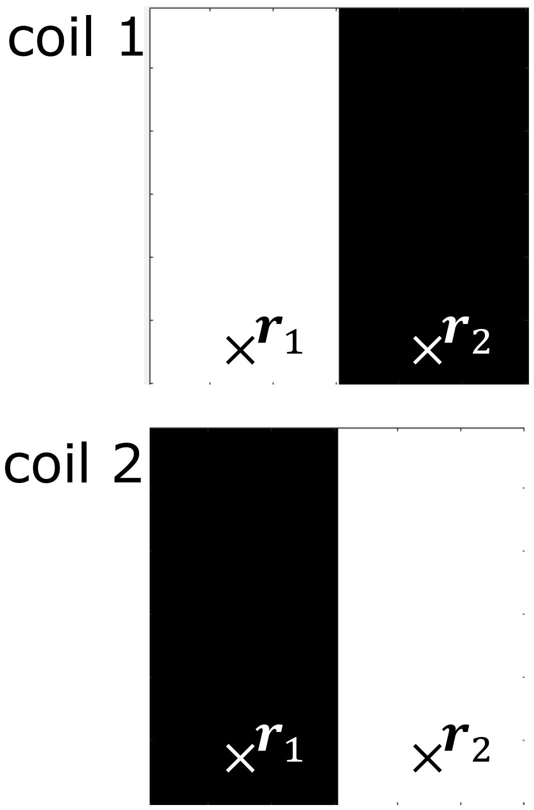
$$s_1^2(\boldsymbol{r}_2) + s_2^2(\boldsymbol{r}_2) = [C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{2,2}$$

$$s_1^2(\boldsymbol{r}_n) + s_2^2(\boldsymbol{r}_n) = [C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}$$

$$\rightarrow g(\boldsymbol{r}_n) = \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

SNR for the perfect sensitivity profile

$$g(\boldsymbol{r}_n) = \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$



$$C(\boldsymbol{r}_1) = \begin{pmatrix} s_{\text{coil1}}(\boldsymbol{r}_1) & s_{\text{coil1}}(\boldsymbol{r}_2) \\ s_{\text{coil2}}(\boldsymbol{r}_1) & s_{\text{coil2}}(\boldsymbol{r}_2) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

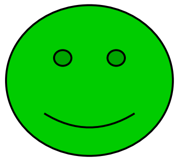
$$C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$g(\boldsymbol{r}_1) = \sqrt{\underbrace{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{1,1}}_{1}\underbrace{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{1,1}}_{1}} = 1$$

$$g(\boldsymbol{r}_2) = \sqrt{\underbrace{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{2,2}}_{1}\underbrace{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{2,2}}_{1}} = 1$$

→ No SNR reduction

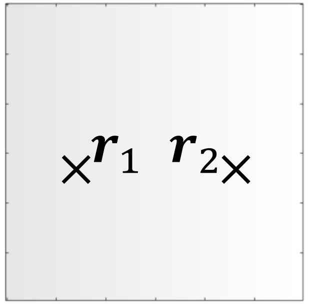


SNR for the bad sensitivity profile

$$g(\boldsymbol{r}_n) = \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

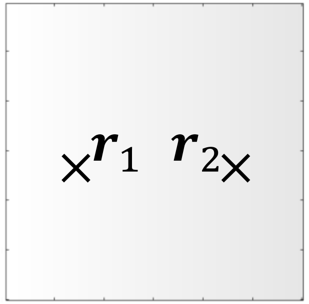
$0.9\mathcal{s}_{\text{bad}} + 0.1\mathcal{s}_{\text{good}}$

coil 1



$$\mathcal{s}_{\text{coil1}}(\boldsymbol{r}) = 0.9 + 0.1 \cdot \frac{x}{FoV}$$

coil 2



$$\mathcal{s}_{\text{coil1}}(\boldsymbol{r}) = 1 - 0.1 \cdot \frac{x}{FoV}$$

$$C(\boldsymbol{r}_1) = \begin{pmatrix} 0.925 & 0.975 \\ 0.975 & 0.925 \end{pmatrix}$$

$$C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1) \approx \begin{pmatrix} 1.8 & 1.8 \\ 1.8 & 1.8 \end{pmatrix}$$

$$\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1} \approx \begin{pmatrix} 200 & -200 \\ -200 & 200 \end{pmatrix}$$

$$g(\boldsymbol{r}_1) = \sqrt{\underbrace{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{1,1}}_{1.8}\underbrace{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{1,1}}_{200}} \approx 19$$

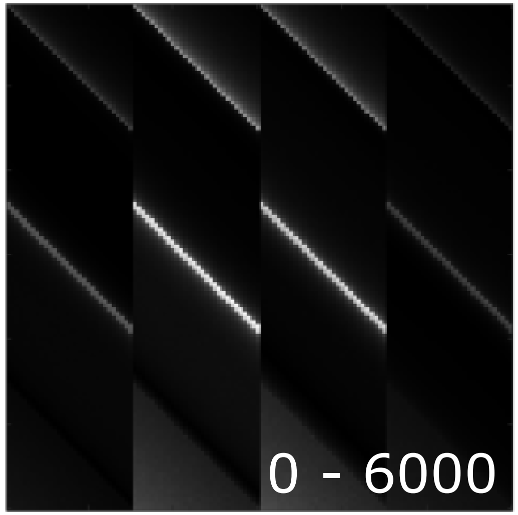
$$g(\boldsymbol{r}_2) = \sqrt{\underbrace{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{2,2}}_{1.8}\underbrace{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{2,2}}_{200}} \approx 19$$

→ SNR reduced by factor 19

Visualization of the g-factor (simulation of noise)

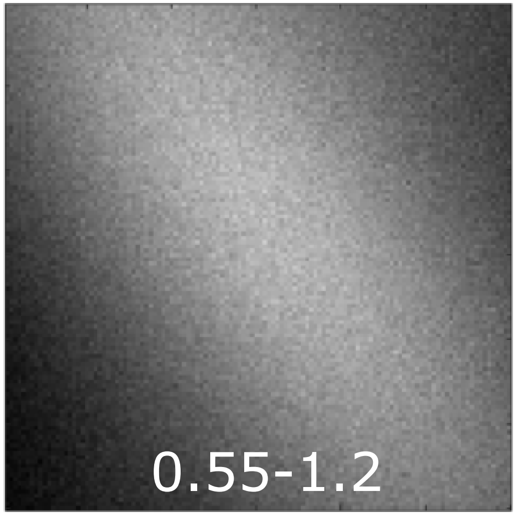
$$g(\boldsymbol{r}) = \frac{SNR_{\mathrm{R}=1}(\boldsymbol{r})}{SNR_{\mathrm{R}}(\boldsymbol{r})} = \frac{\sigma_{\mathrm{R}}(\boldsymbol{r})}{\sigma_{\mathrm{R}=1}(\boldsymbol{r})}$$

std for $R = 4$ and $\sigma = 1$



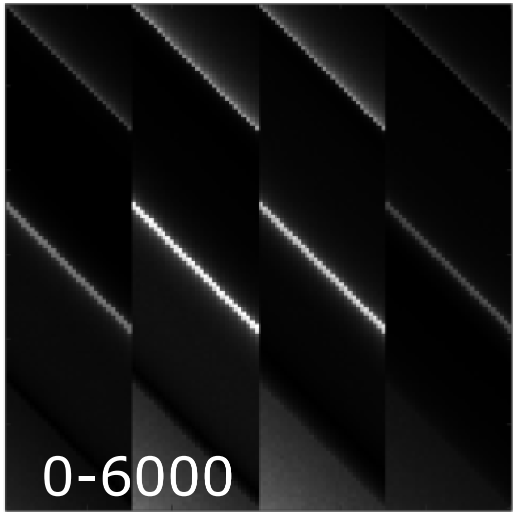
:

std for $R = 1$ and $\sigma = 1$



=

$g(\boldsymbol{r})$



$$= \sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

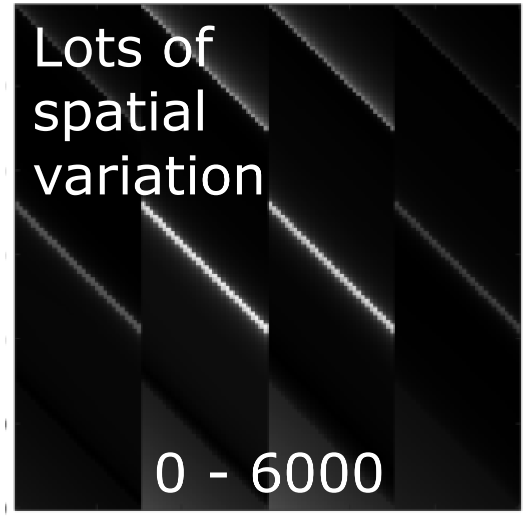
$$= \sigma \cdot \sqrt{\left(C^T_{\mathrm{R}=1}(\boldsymbol{r}_1)C_{\mathrm{R}=1}(\boldsymbol{r}_1)\right)^{-1}} = \sigma / \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}}$$

$$= \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}} \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

The C-terms computed

$$g(\boldsymbol{r}_1) = \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

std for $R = 4$ and $\sigma = 1$



=

$$\sigma \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

:

std for $R = 1$ and $\sigma = 1$

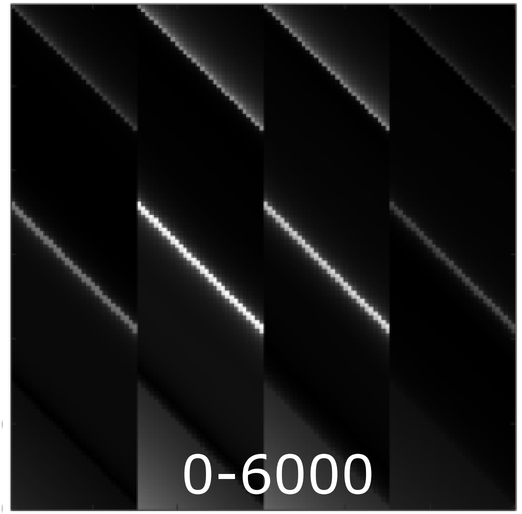
=

$$\sigma \cdot \sqrt{\left(C^T_{\mathrm{R=1}}(\boldsymbol{r}_1)C_{\mathrm{R=1}}(\boldsymbol{r}_1)\right)^{-1}}$$

$$= \sigma / \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}}$$

=

$g(\boldsymbol{r})$



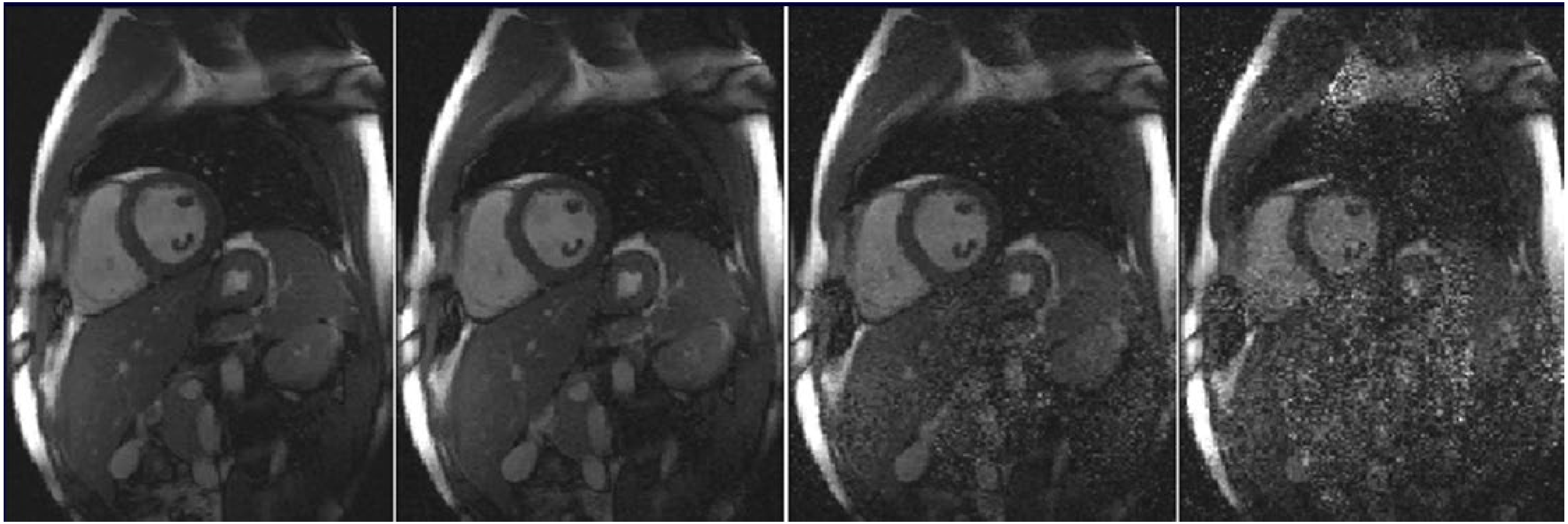
=

$$\sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}} \cdot \sqrt{\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

Image examples: cardiac imaging

- acceleration reduces the SNR

$R = 1$ $R = 2$ $R = 3$ $R = 4$



Friedrich-Alexander-Universität
Medizinische Fakultät

Uniklinikum Erlangen

One missing ingredient

One thing is missing

$$SNR_{\mathrm{R}}(\boldsymbol{r}) = \frac{SNR_{\mathrm{R=1}}(\boldsymbol{r})}{g(\boldsymbol{r})} \cdot \frac{1}{\sqrt{R}}$$

this term can be optimized by good coil design

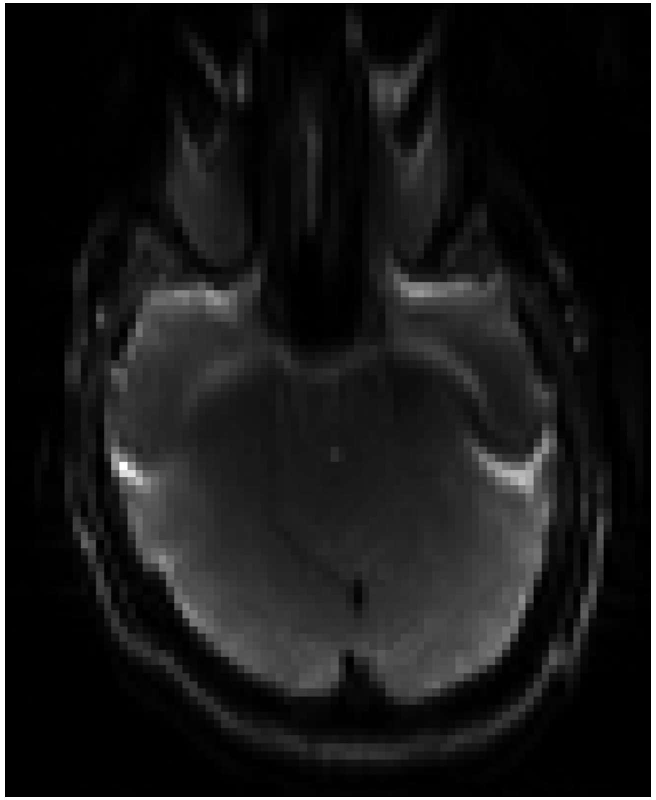
this term is unavoidable

Why the factor of $\sqrt{R}$?

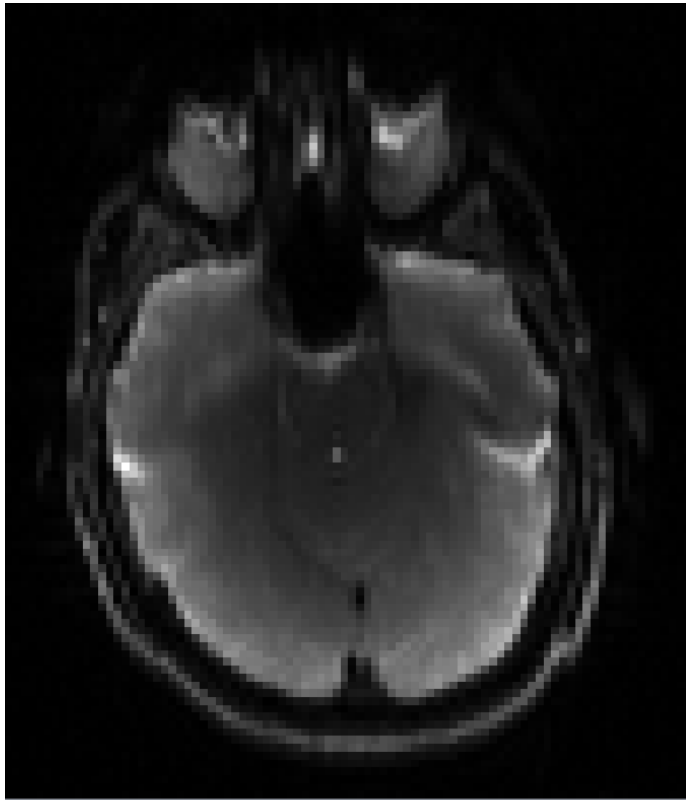
Because $R \propto$ (time spent on sampling data)$^{-1}$.

Echo planar imaging & parallel imaging

reduced image distortions



$R = 1$



$R = 2$

Thanks to A. Riexinger

Friedrich-Alexander-Universität
Medizinische Fakultät

Uniklinikum Erlangen

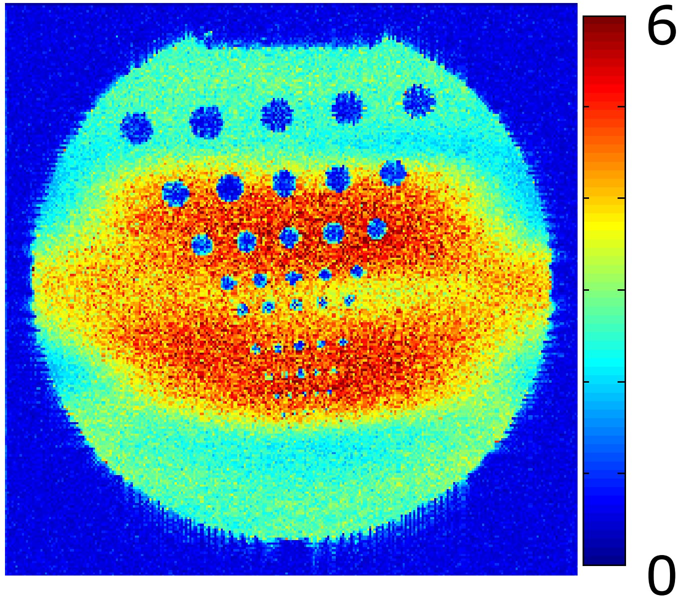
What is parallel imaging good for?

- Parallel imaging is good for accelerating the acquisition
- It does not increase SNR
- In fact, one has to pay with SNR for the acceleration

Summary

Geometry-factor introduced

$$g(\boldsymbol{r}) = \frac{SNR_{\mathrm{R}=1}(\boldsymbol{r})}{SNR_{\mathrm{R}}(\boldsymbol{r})}$$



Describes imperfection of coil setup

Explicit formula derived

$$g(\boldsymbol{r}_n) = \sqrt{[C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)]_{n,n}\left[\left(C^T(\boldsymbol{r}_1)C(\boldsymbol{r}_1)\right)^{-1}\right]_{n,n}}$$

Less time spent on sampling data

→ additional factor $R^{-1/2}$

$$SNR_{\mathrm{R}}(\boldsymbol{r}) = \frac{SNR_{\mathrm{R}=1}(\boldsymbol{r})}{g(\boldsymbol{r})} \cdot \frac{1}{\sqrt{R}}$$

k-Space based parallel imaging

Grappa

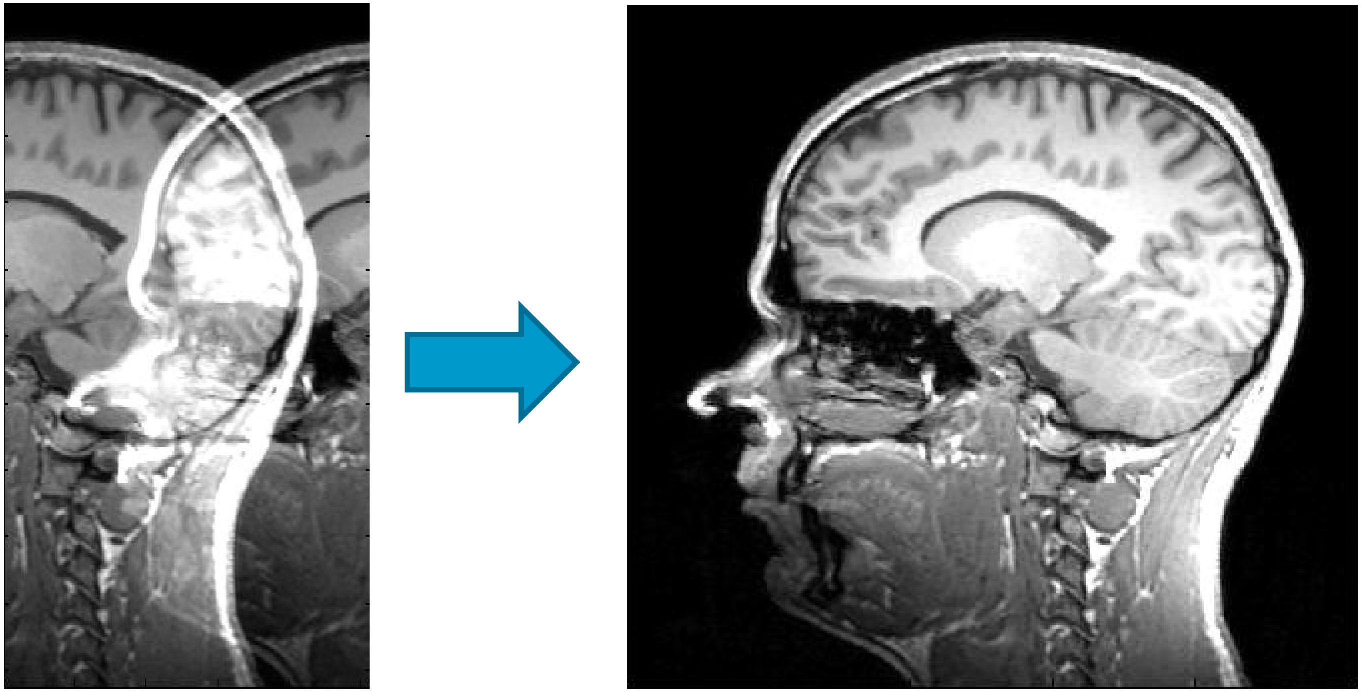
(Generalized Autocalibrating Partially Parallel Acquisitions)

Griswold et al., Magn Reson Med 47:1202-10 (2002)

Grappa: The basic principle

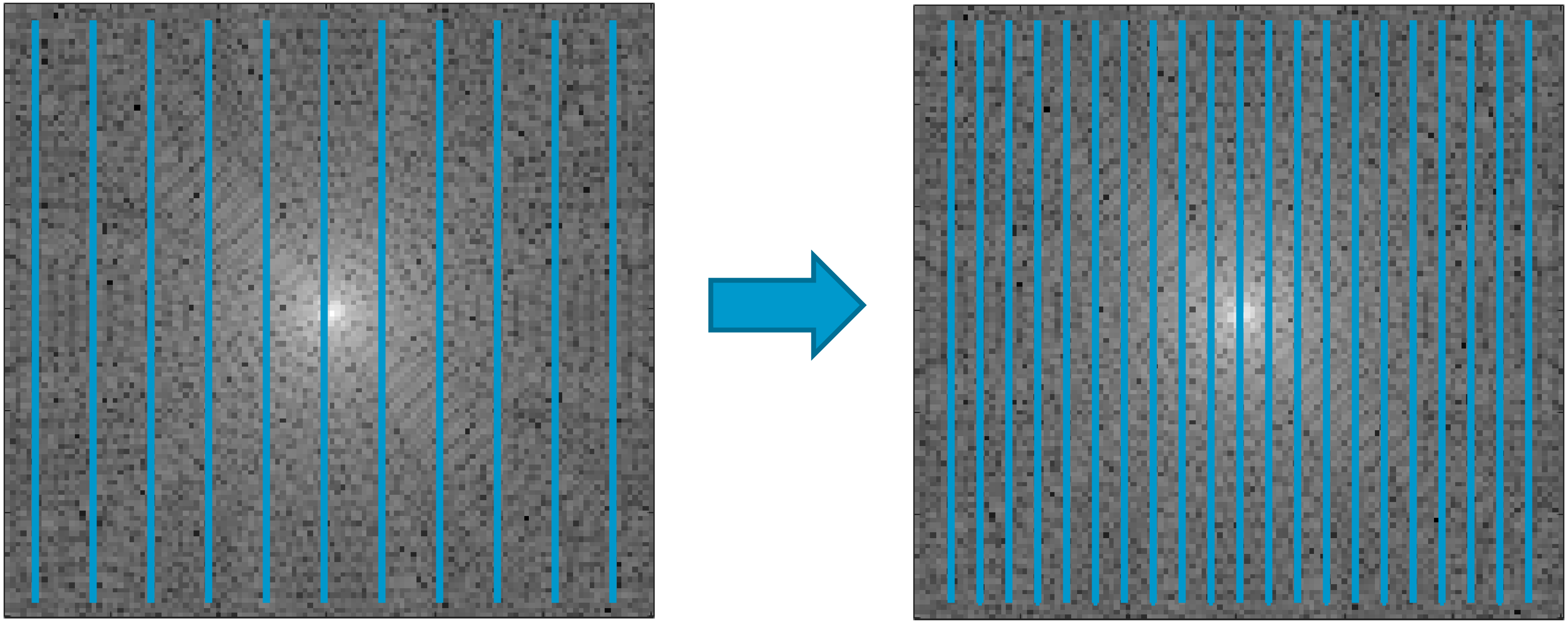
Parallel Imaging
Basic Principle

Basic principle 1 (SENSE): unwrap the image



Parallel Imaging
Basic Principle 2

Basic principle 2 (GRAPPA): fill the k-space

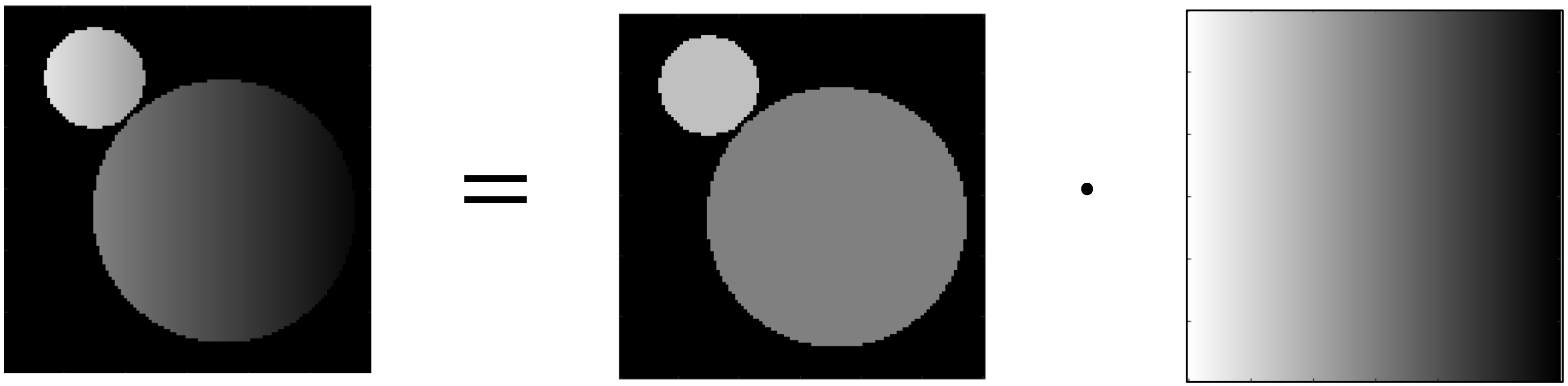


FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
MEDIZINISCHE FAKULTÄT

Universitätsklinikum Erlangen

Motivation:
From x space to k space

$$S_{\mathrm{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathscr{s}_{\mathrm{coil1}}(\boldsymbol{r})$$

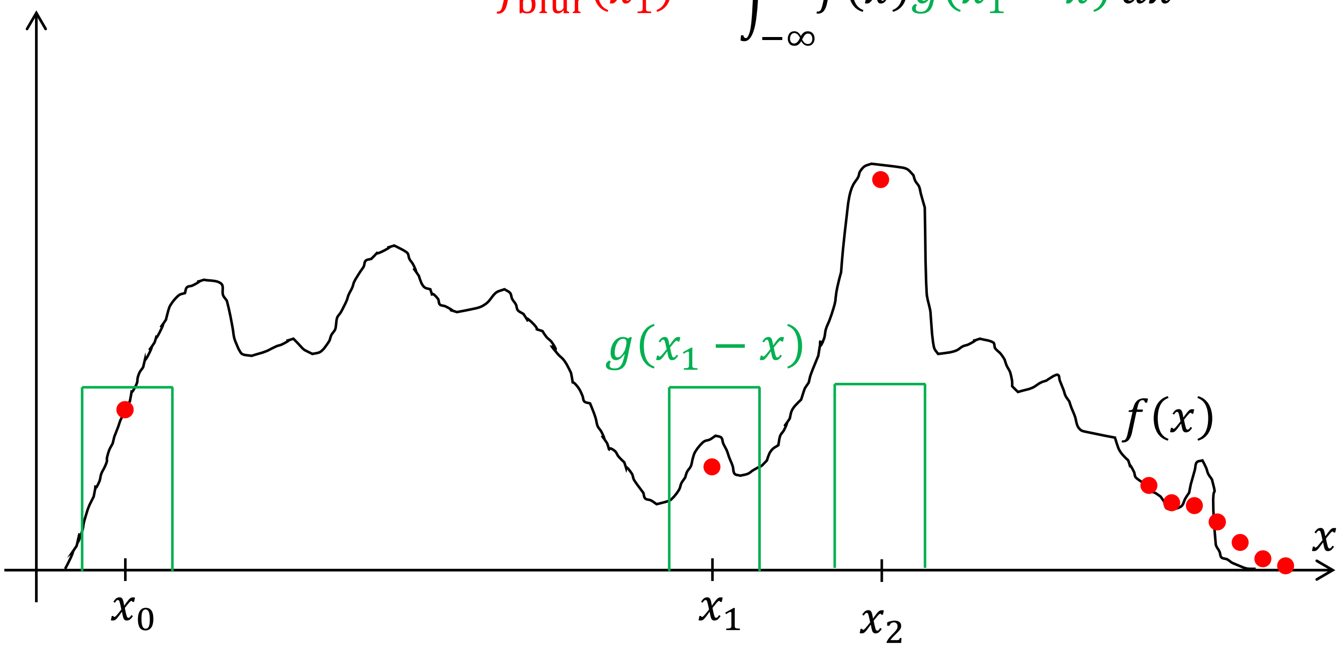


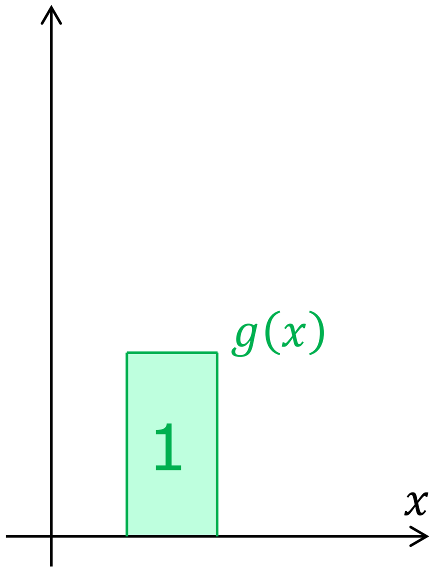
$$\Rightarrow \quad \tilde{S}_{\mathrm{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{\mathscr{s}}_{\mathrm{coil1}}(\boldsymbol{k})$$

Reminder: Convolution with a boxcar function of width Δx and unit area under the curve

$$f_{\text{blur}}(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(x')g(x-x')dx'$$

$$f_{\text{blur}}(x_1) = \int_{-\infty}^{\infty} f(x)g(x_1-x)\,dx$$

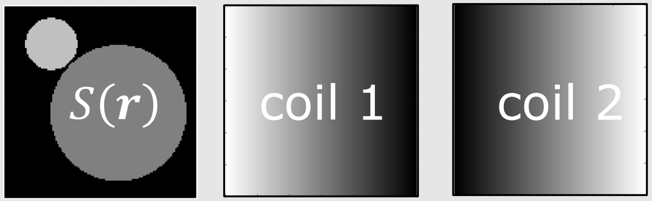




Pop quiz

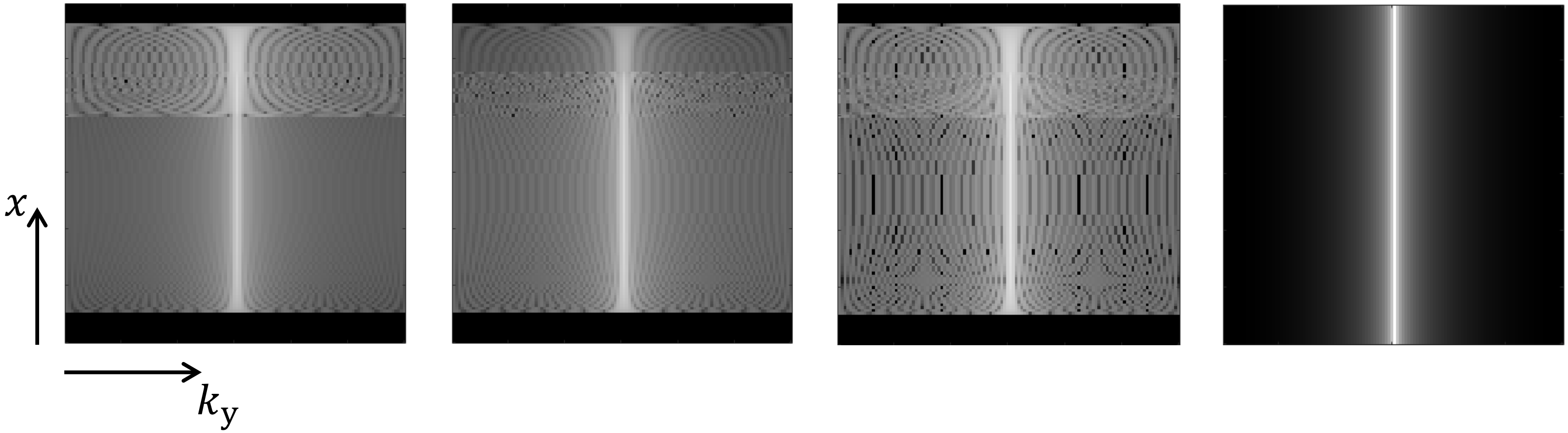
$$S_{\mathrm{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot s_{\mathrm{coil1}}(\boldsymbol{r})$$

$$\tilde{S}_{\mathrm{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{s}_{\mathrm{coil1}}(\boldsymbol{k})$$



Assign correctly:

$\tilde{S}$ $\qquad$ $\tilde{S}_{\mathrm{coil2}}$ $\qquad$ $\tilde{s}_{\mathrm{coil1}}$ $\qquad$ $\tilde{S}_{\mathrm{coil1}}$



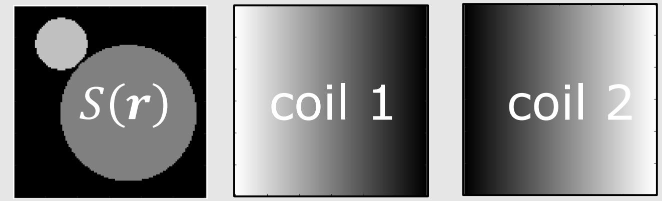
(for now: sensitivity profiles have no x-dependency)

(k-space plots: $\log(|\tilde{S}|)$ is plotted)

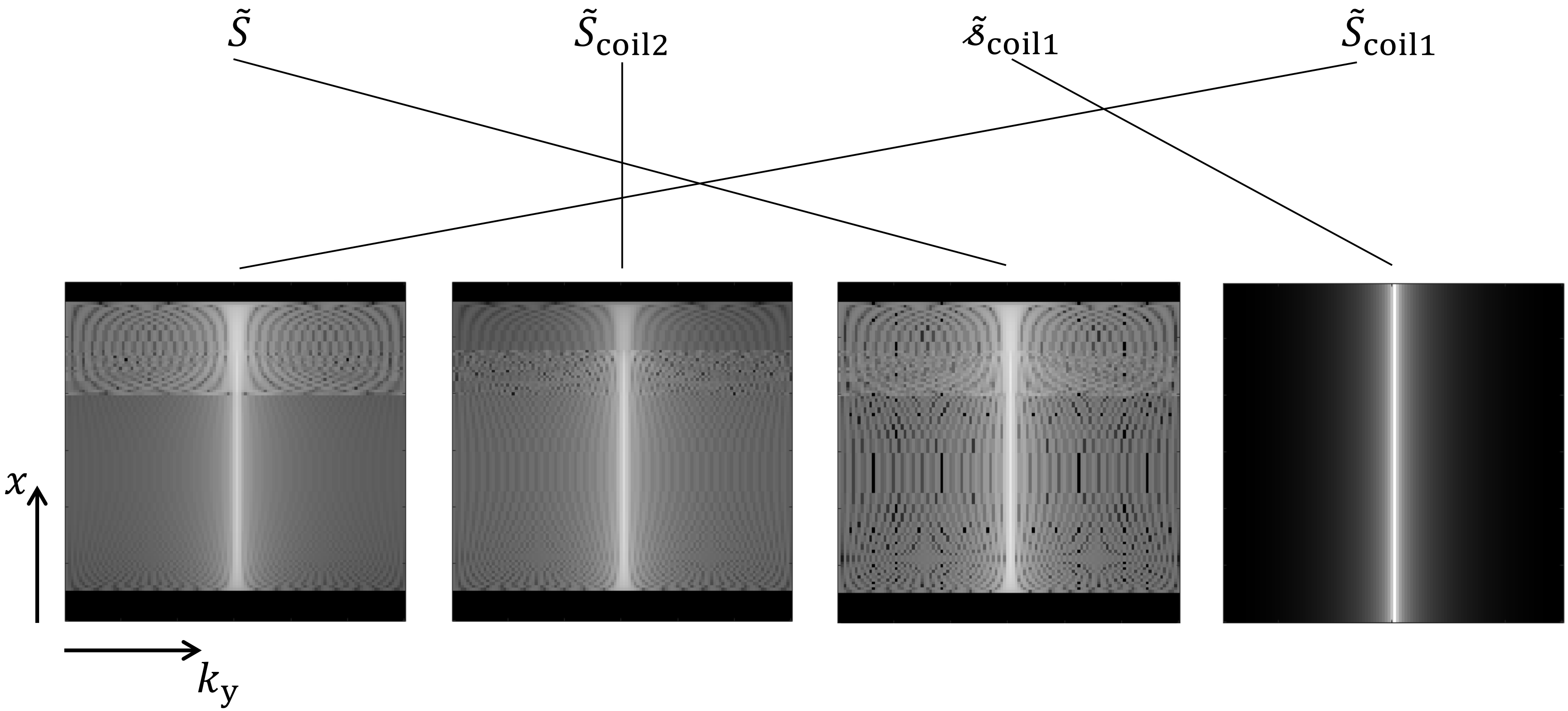
Pop quiz

$$S_{\text{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathcal{s}_{\text{coil1}}(\boldsymbol{r})$$

$$\tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{\mathcal{s}}_{\text{coil1}}(\boldsymbol{k})$$



Assign correctly:

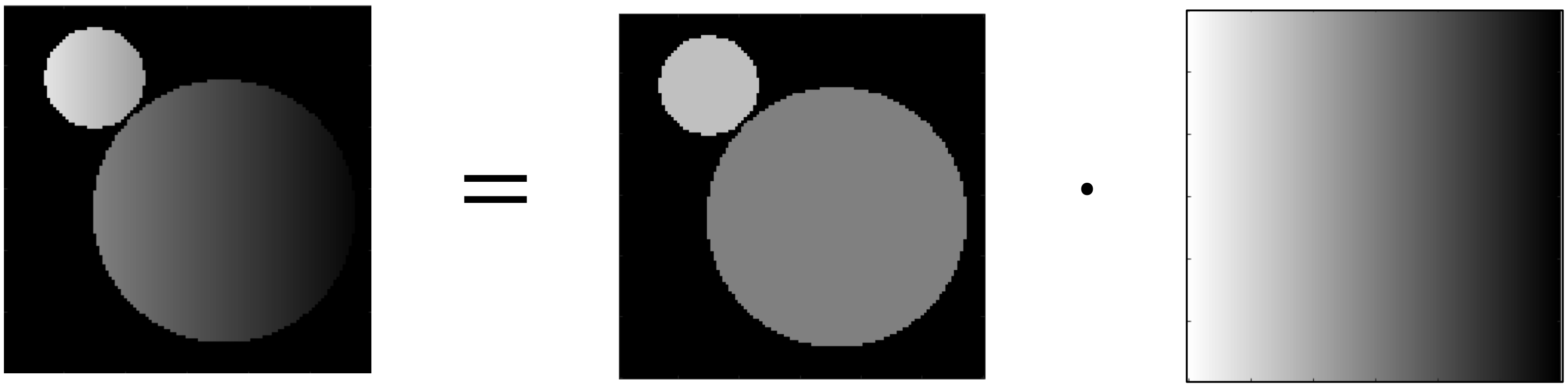


(for now: sensitivity profiles have no x-dependency)

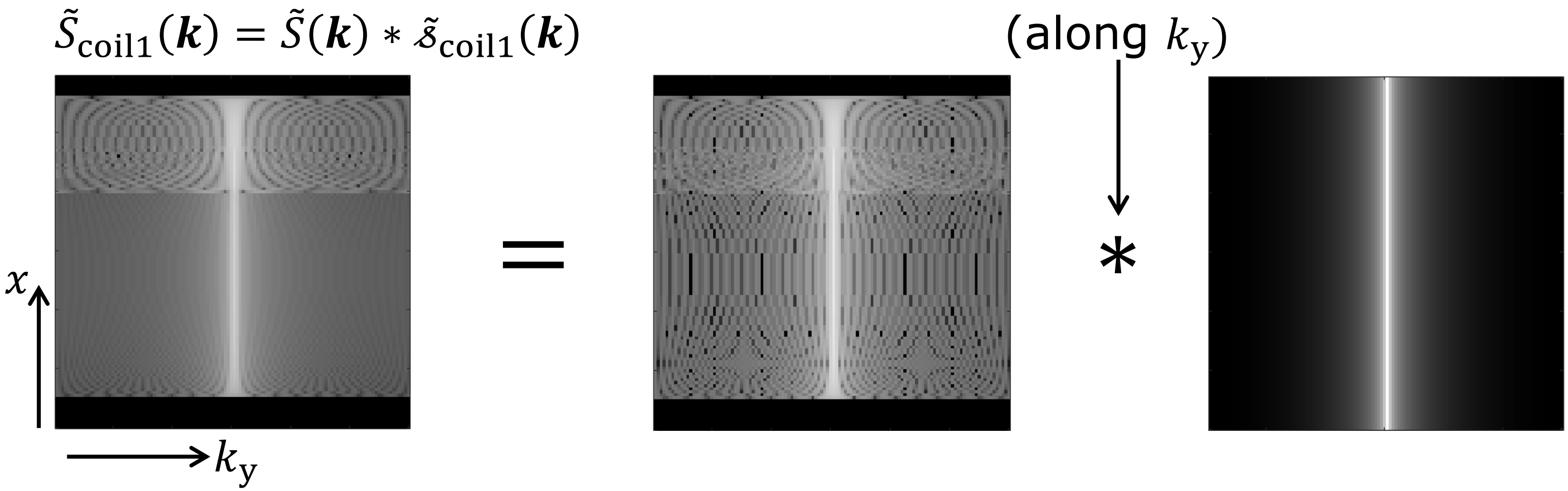
(k-space plots: $\log(|\tilde{S}|)$ is plotted)

From x space to k space

$$S_{\mathrm{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathscr{s}_{\mathrm{coil1}}(\boldsymbol{r})$$

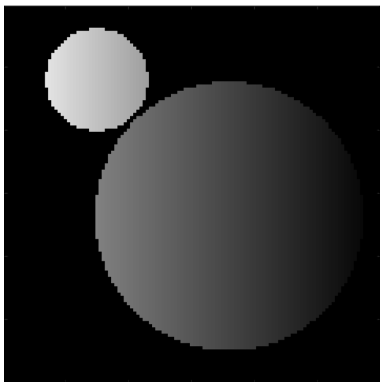


$$\tilde{S}_{\mathrm{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{\mathscr{s}}_{\mathrm{coil1}}(\boldsymbol{k})$$

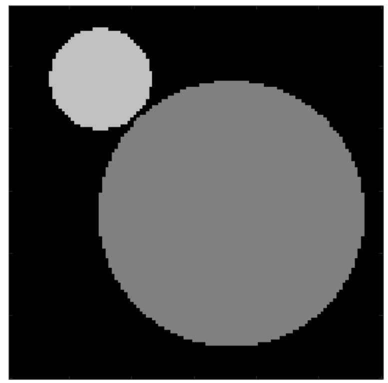


Motivation:
From x space to k space

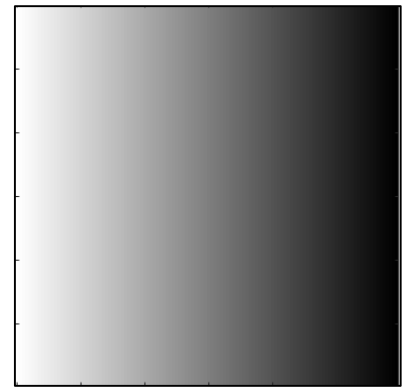
$$S_{\text{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathscr{s}_{\text{coil1}}(\boldsymbol{r}) \quad \Longrightarrow \quad \tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{\mathscr{s}}_{\text{coil1}}(\boldsymbol{k})$$



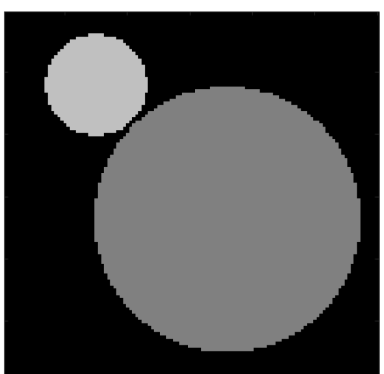
=



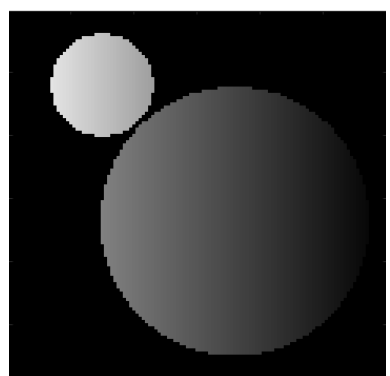
•



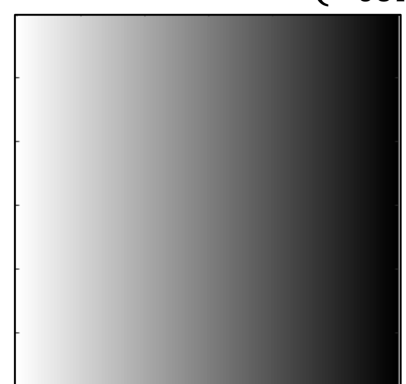
Alternatively: $S(\boldsymbol{r}) = S_{\text{coil1}}(\boldsymbol{r}) \cdot \frac{1}{\mathscr{s}_{\text{coil1}}(\boldsymbol{r})} \quad \Longrightarrow \quad \tilde{S}(\boldsymbol{k}) = \tilde{S}_{\text{coil1}}(\boldsymbol{k}) * \mathcal{F}\left\{\frac{1}{\mathscr{s}_{\text{coil1}}(\boldsymbol{r})}\right\}$



=



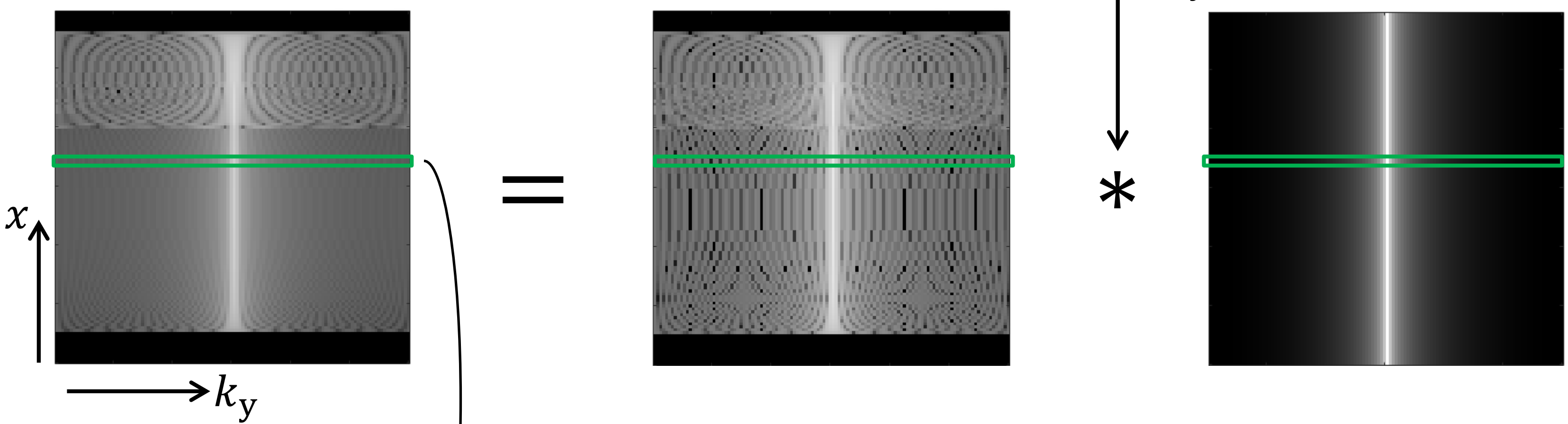
/



1D notation

$$\tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{s}_{\text{coil1}}(\boldsymbol{k})$$

(along k_{y})



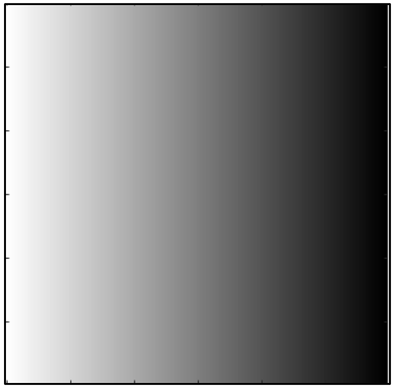
1D notation

$$\tilde{S}_{\text{coil1}}(k = m \cdot \Delta k) = \sum_{n} \tilde{s}_{\text{coil1}}(m \cdot \Delta k - n \cdot \Delta k) \cdot \tilde{S}(n \cdot \Delta k)$$

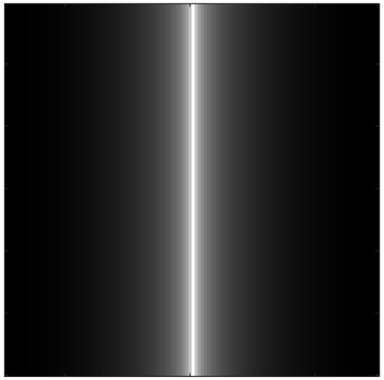
Simplified notation: $\tilde{S}_{\text{coil1}}(m) = \sum_n \tilde{s}_{\text{coil1}}(m - n) \cdot \tilde{S}(n)$

(for now: sensitivity profiles have no x-dependency)

Properties of $\tilde{s}_{\text{coil1}}(\boldsymbol{k})$



$s_{\text{coil1}}(\boldsymbol{r})$ has only low spatial frequencies



→ $\tilde{s}_{\text{coil1}}(\boldsymbol{k})$ is small at large k

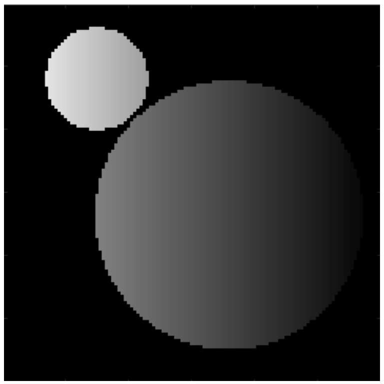
$\tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{s}_{\text{coil1}}(\boldsymbol{k})$

$\tilde{S}_{\text{coil1}}(\boldsymbol{k}_1)$ connected only to points of $\tilde{S}(\boldsymbol{k}_2)$ with small $|k_2 - k_1|$

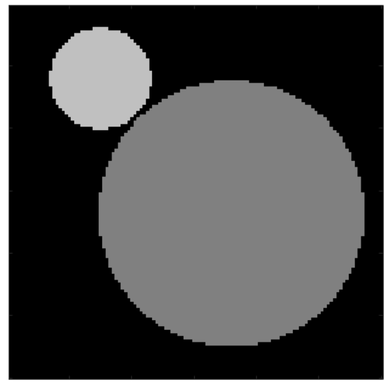
Vice versa:
$\tilde{S}(\boldsymbol{k}_2)$ connected only to points of $\tilde{S}_{\text{coil1}}(\boldsymbol{k}_1)$ with small $|k_2 - k_1|$ i

Summary

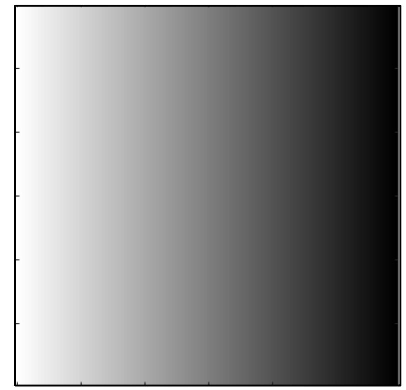
We can easily go from position space



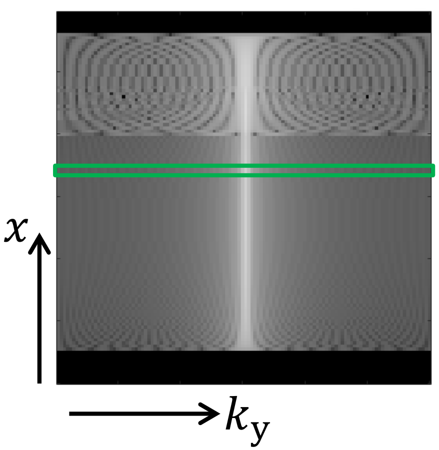
$=$



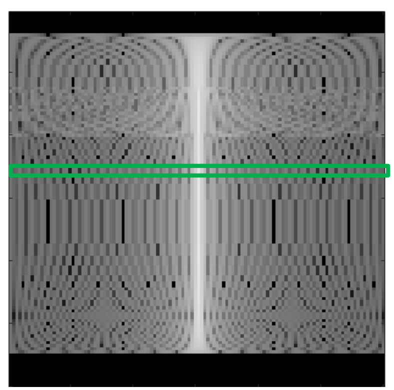
$\cdot$



to k-space

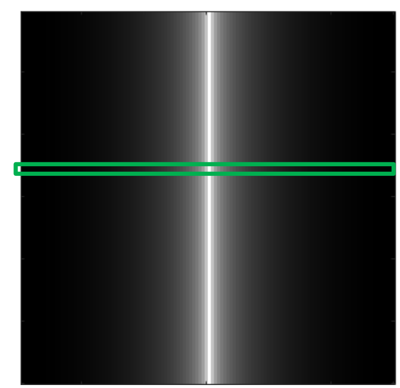


$=$



(along k_y)

$*$



Grappa: Weights

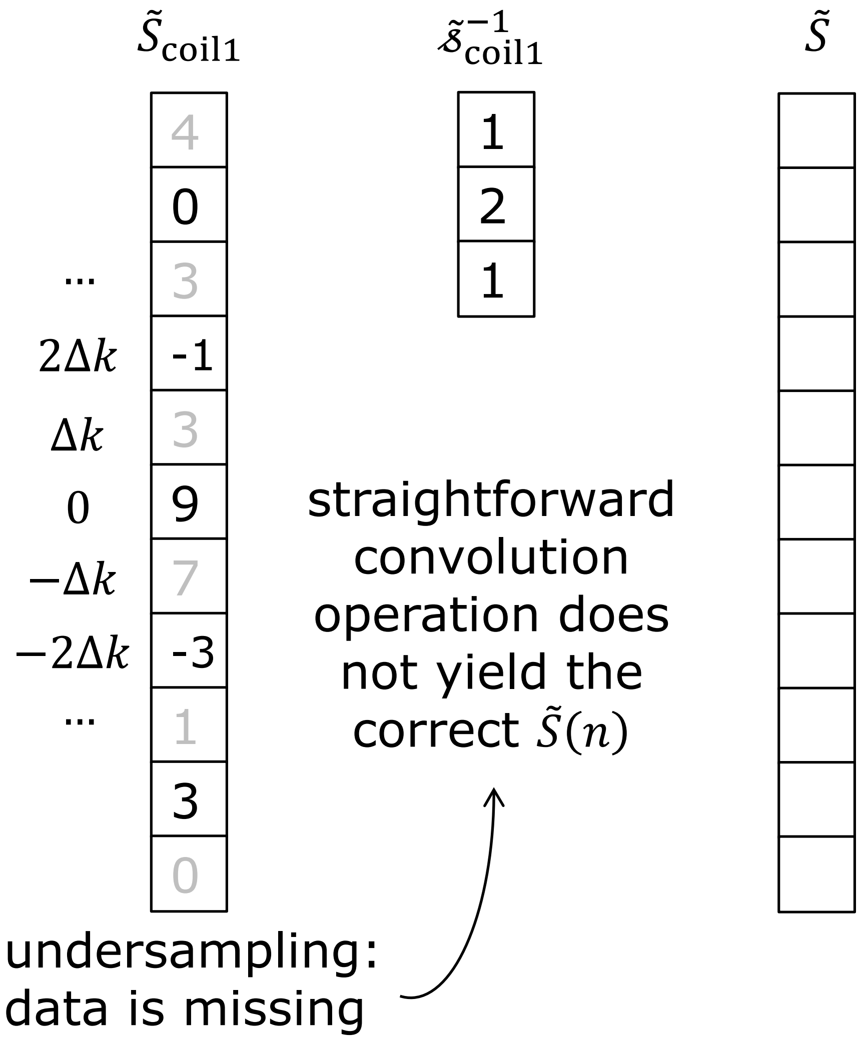
Signal reconstruction with weights

$$\tilde{S}(m) = \sum_{n} \tilde{s}^{-1}_{\text{coil1}}(n) \cdot \tilde{S}_{\text{coil1}}(m-n)$$

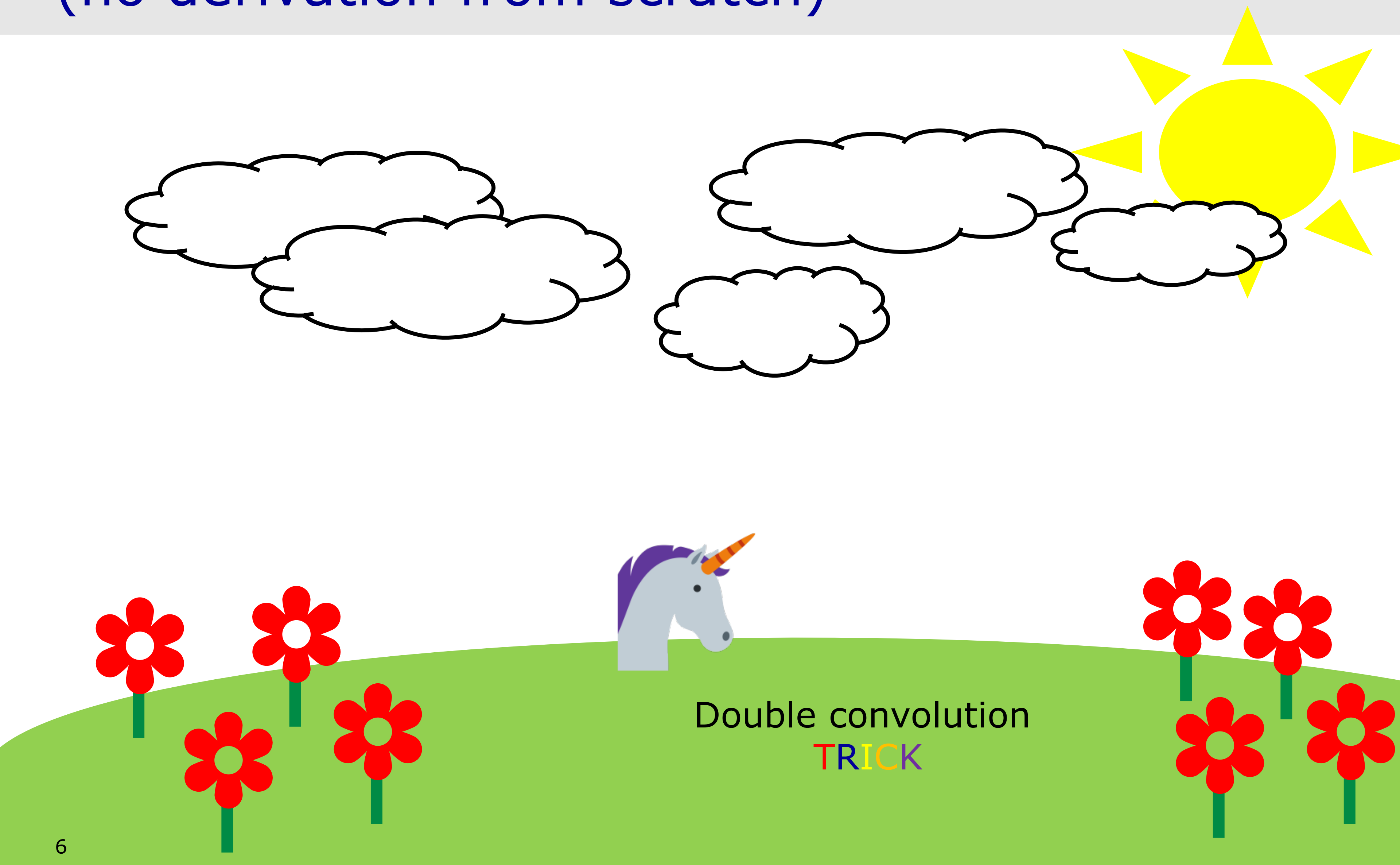
	$\tilde{S}_{\text{coil1}}$	$\tilde{s}^{-1}_{\text{coil1}}$	$\tilde{S}$	
	4	1		
	0	2	7	= 4·1+0·2+3·1
…	3	1	5	= 0·1+3·2+(-1)·1
$2\Delta k$	-1		4	= 3·1+(-1)·2+3·1
Δk	3		14	
0	9		28	
$-\Delta k$	7		20	
$-2\Delta k$	-3		2	
…	1		2	
	3		7	
	0			

Signal reconstruction with weights

$$\tilde{S}(m) = \sum_n \tilde{s}_{\text{coil1}}^{-1}(n) \cdot \tilde{S}_{\text{coil1}}(m-n)$$



Magic trick
(no derivation from scratch)



Signal reconstruction with weights

	$\tilde{S}_{\text{coil1}}$	$\widetilde{w}_1$	$\tilde{S}$	$\widetilde{w}_2$	$\tilde{S}_{\text{coil2}}$
	4	1		1	2
	0	2		3	1
…	3	1		1	3
$2\Delta k$	-1				2
Δk	3				3
0	9				8
$-\Delta k$	7				4
$-2\Delta k$	-3				-1
…	1				1
	3				3
	0				1

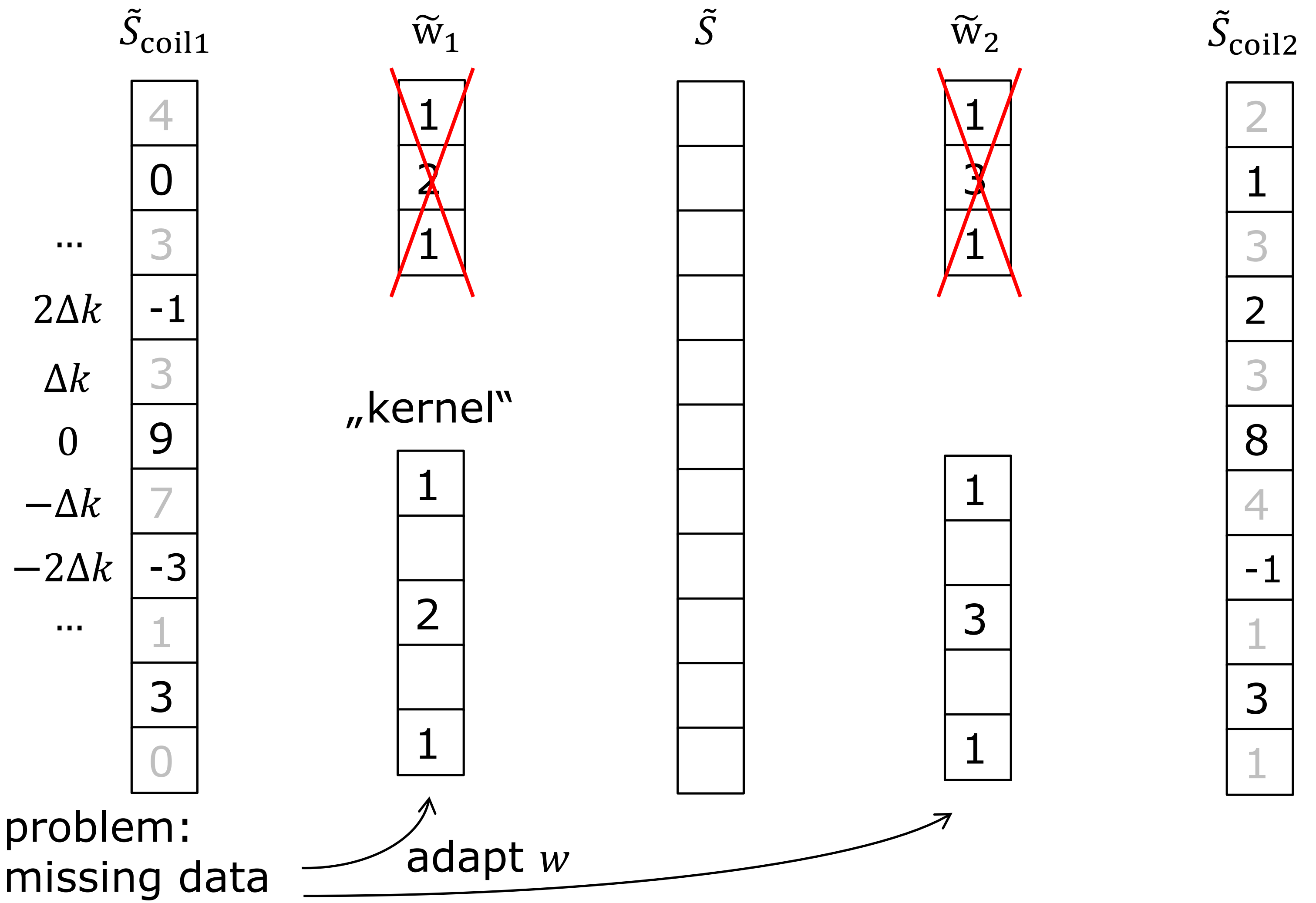
straightforward convolution operation does not yield the correct $\tilde{S}(n)$

TRICK
Use 2nd coil and
$\tilde{S} = w_1 * \tilde{S}_{\text{coil1}} + w_2 * \tilde{S}_{\text{coil2}}$
with appropriately chosen weights w_1 and w_2

undersampling:
gray data is missing

Representation of kernelse

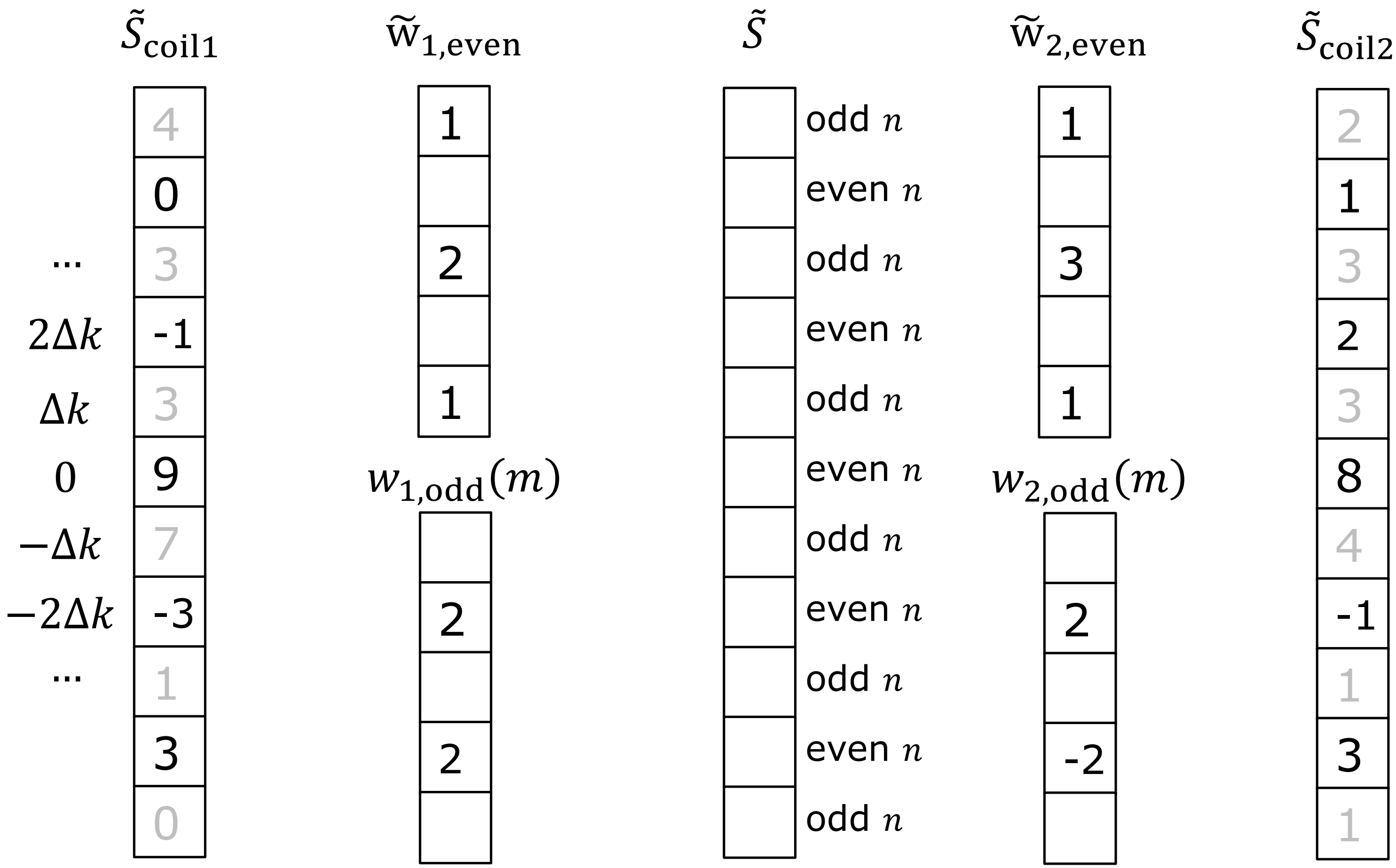
TRICK: $\tilde{S} = w_1 * \tilde{S}_{\text{coil1}} + w_2 * \tilde{S}_{\text{coil2}}$
with well chosen weights w_1 and w_2



Odd and even kernels

TRICK:

$$\tilde{S}_{\text{odd}} = w_{1,\text{odd}} * \tilde{S}_{\text{coil1}} + w_{2,\text{odd}} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{even}} = w_{1,\text{even}} * \tilde{S}_{\text{coil1}} + w_{2,\text{even}} * \tilde{S}_{\text{coil2}}$$



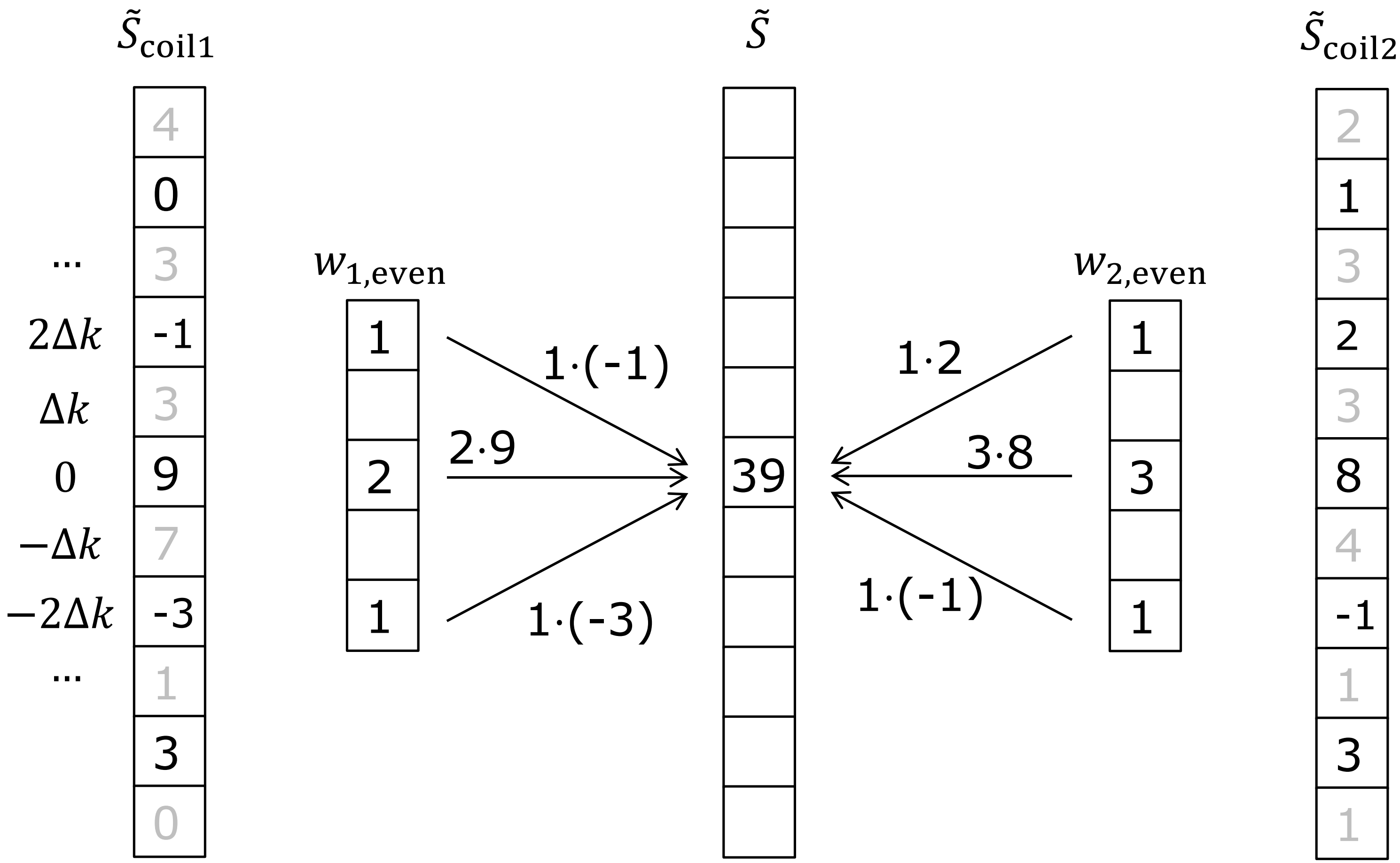
problem: odd and even points are different

w_{odd} and w_{even} are needed

Signal reconstruction with even kernel

TRICK:

$$\tilde{S}_{\mathrm{odd}} = w_{1,\mathrm{odd}} * \tilde{S}_{\mathrm{coil1}} + w_{2,\mathrm{odd}} * \tilde{S}_{\mathrm{coil2}}$$
$$\tilde{S}_{\mathrm{even}} = w_{1,\mathrm{even}} * \tilde{S}_{\mathrm{coil1}} + w_{2,\mathrm{even}} * \tilde{S}_{\mathrm{coil2}}$$

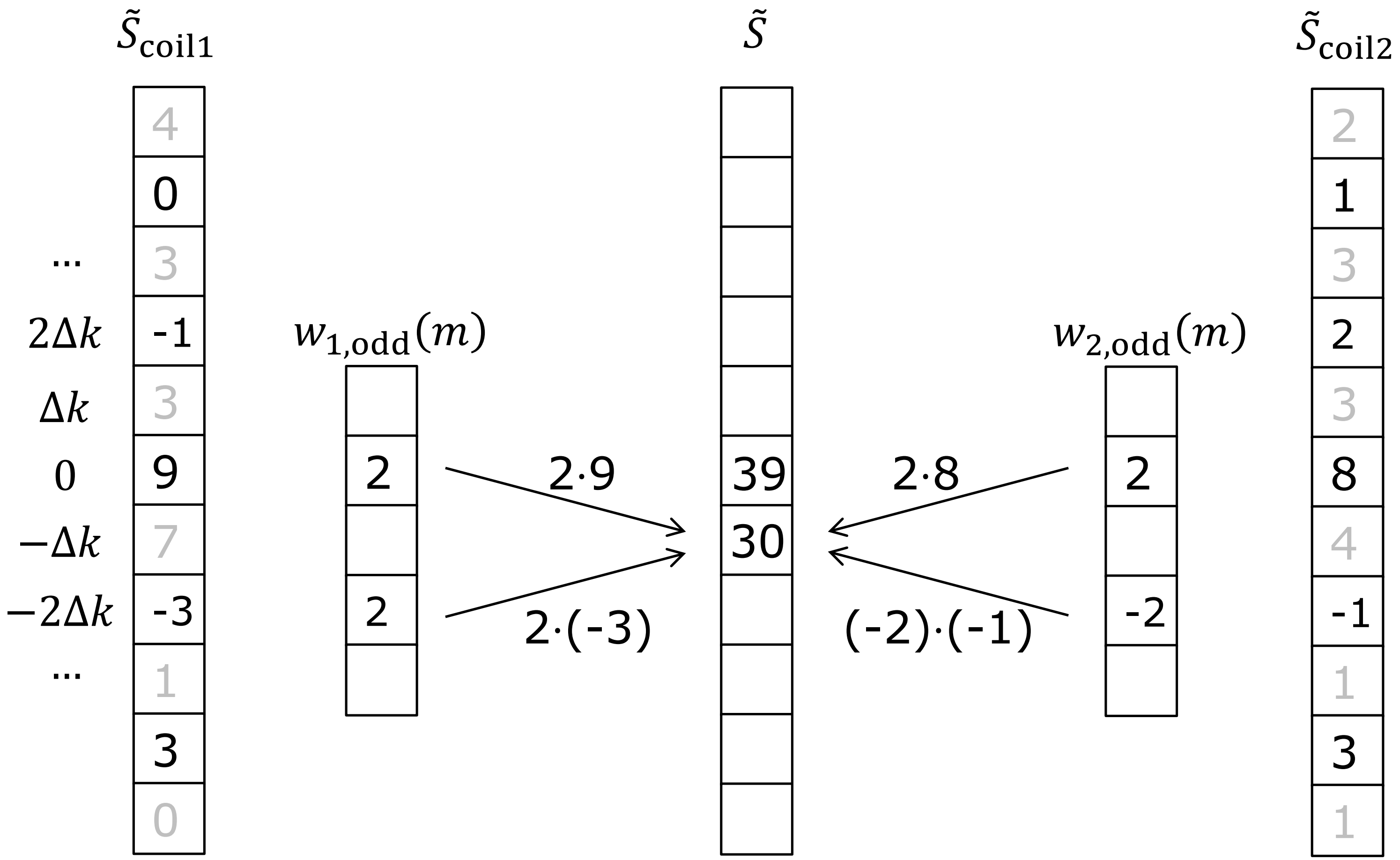


1·(-1) + 2·9 + 1·(-3) + 1·2 + 3·8 + 1·(-1) = 39

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{odd}} = w_{1,\text{odd}} * \tilde{S}_{\text{coil1}} + w_{2,\text{odd}} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{even}} = w_{1,\text{even}} * \tilde{S}_{\text{coil1}} + w_{2,\text{even}} * \tilde{S}_{\text{coil2}}$$

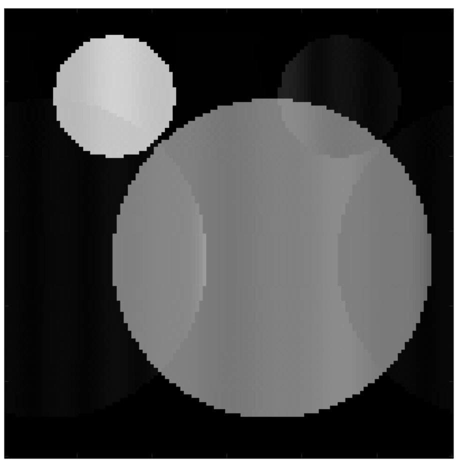


2·9 + 2·(-3) + 2·8 + (-2)·(-1) = 30

Image examples with small kernels

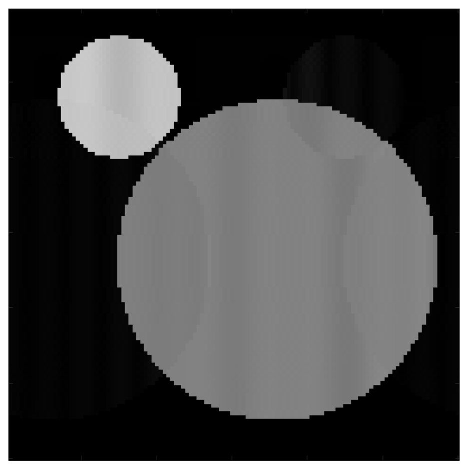
1		2		1

kernel size = 3



1		2		3		2		1

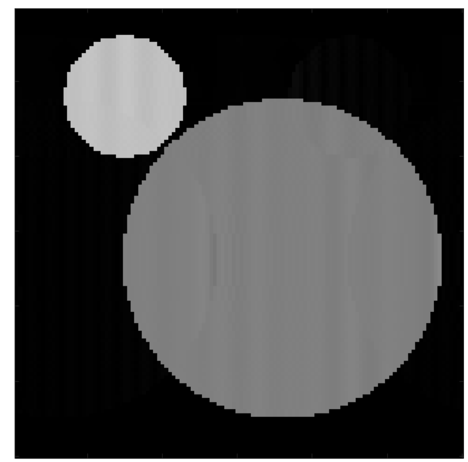
kernel size =5



1		2		1		2		3		4		1		2		3		2		1

(arbitrary numbers)

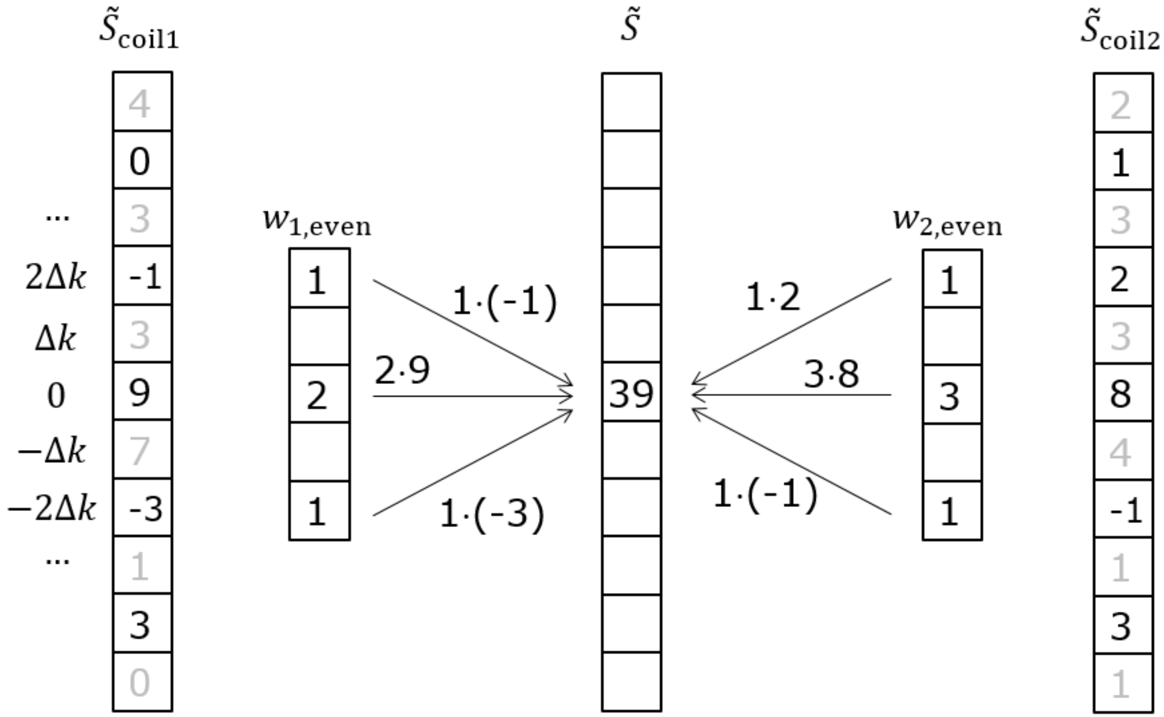
kernel size =11



Summary

Double convolution TRICK

$$\tilde{S}_{\text{odd}} = w_{1,\text{odd}} * \tilde{S}_{\text{coil1}} + w_{2,\text{odd}} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{even}} = w_{1,\text{even}} * \tilde{S}_{\text{coil1}} + w_{2,\text{even}} * \tilde{S}_{\text{coil2}}$$



Grappa: Coil to coil fit

Different approaches

Approach 1:

undersampled $\tilde{S}_{\text{coil1}}, \tilde{S}_{\text{coil2}}, \ldots$ $\Longrightarrow$ fully sampled $\tilde{S}$

Approach 2:

undersampled $\tilde{S}_{\text{coil1}}, \tilde{S}_{\text{coil2}}, \ldots$ $\Longrightarrow$ fully sampled $\tilde{S}_{\text{coil1}}, \tilde{S}_{\text{coil2}}, \ldots$

Coil to coil

There exists a convolution-like relation between $\tilde{S}_{\text{coil1}}(\boldsymbol{k})$ and $\tilde{S}_{\text{coil2}}(\boldsymbol{k})$ because:

$$S_{\text{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathcal{s}_{\text{coil1}}(\boldsymbol{r})$$

$$S_{\text{coil2}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathcal{s}_{\text{coil2}}(\boldsymbol{r})$$

$$\Rightarrow$$

$$S_{\text{coil2}}(\boldsymbol{r}) = S_{\text{coil1}}(\boldsymbol{r}) \cdot \frac{\mathcal{s}_{\text{coil2}}(\boldsymbol{r})}{\mathcal{s}_{\text{coil1}}(\boldsymbol{r})}$$

$$S_{\text{coil1}}(\boldsymbol{r}) = S_{\text{coil2}}(\boldsymbol{r}) \cdot \frac{\mathcal{s}_{\text{coil1}}(\boldsymbol{r})}{\mathcal{s}_{\text{coil2}}(\boldsymbol{r})}$$

$$\Rightarrow$$

$$\tilde{S}_{\text{coil2}}(\boldsymbol{k}) = \tilde{S}_{\text{coil1}}(\boldsymbol{k}) * \mathcal{F}\left\{\frac{\mathcal{s}_{\text{coil2}}(\boldsymbol{r})}{\mathcal{s}_{\text{coil1}}(\boldsymbol{r})}\right\}$$

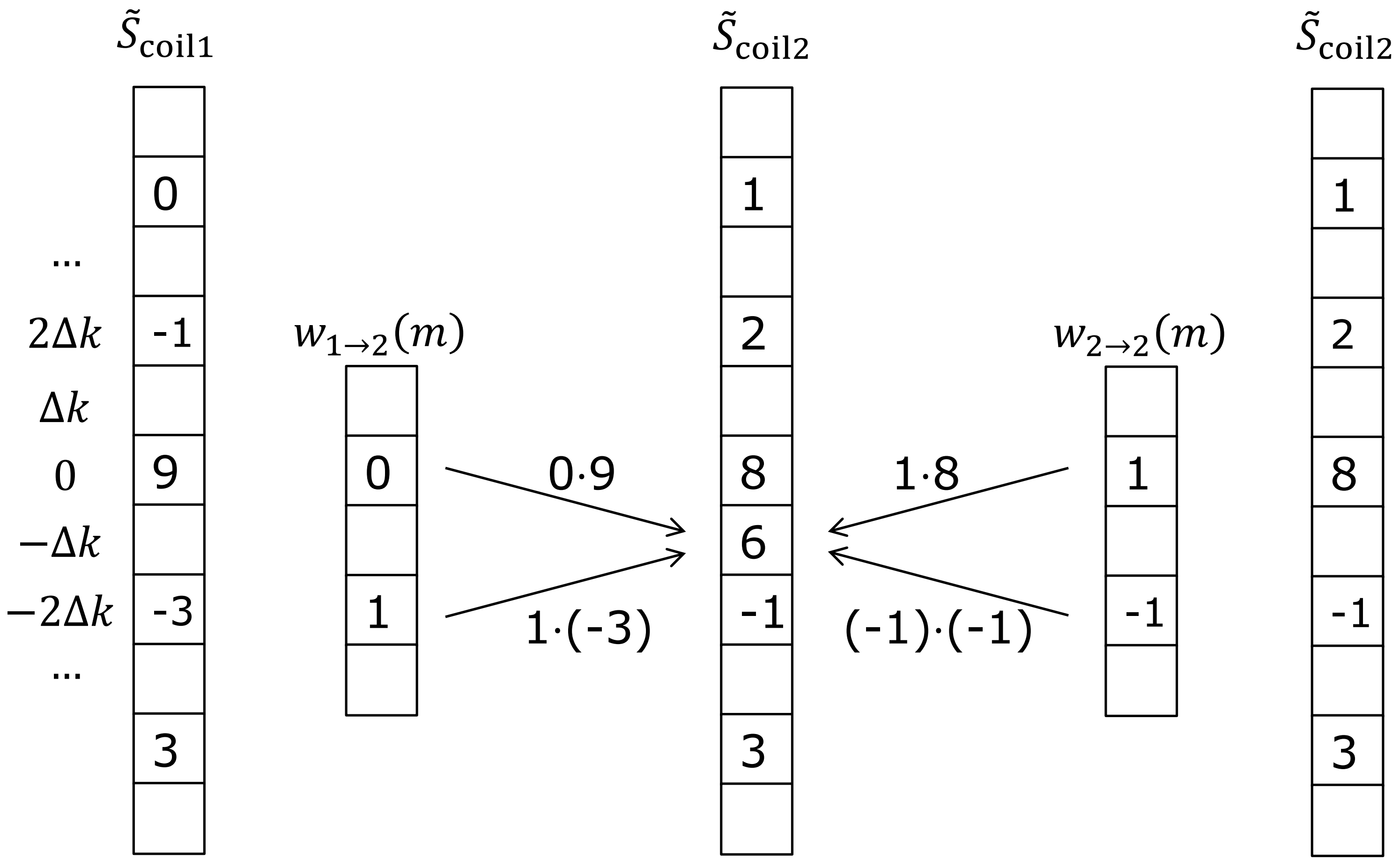
$$\tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}_{\text{coil2}}(\boldsymbol{k}) * \mathcal{F}\left\{\frac{\mathcal{s}_{\text{coil1}}(\boldsymbol{r})}{\mathcal{s}_{\text{coil2}}(\boldsymbol{r})}\right\}$$

→ Coil to coil fitting should be possible

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to1} * \tilde{S}_{\text{coil1}} + w_{2\to1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to2} * \tilde{S}_{\text{coil1}} + w_{2\to2} * \tilde{S}_{\text{coil2}}$$

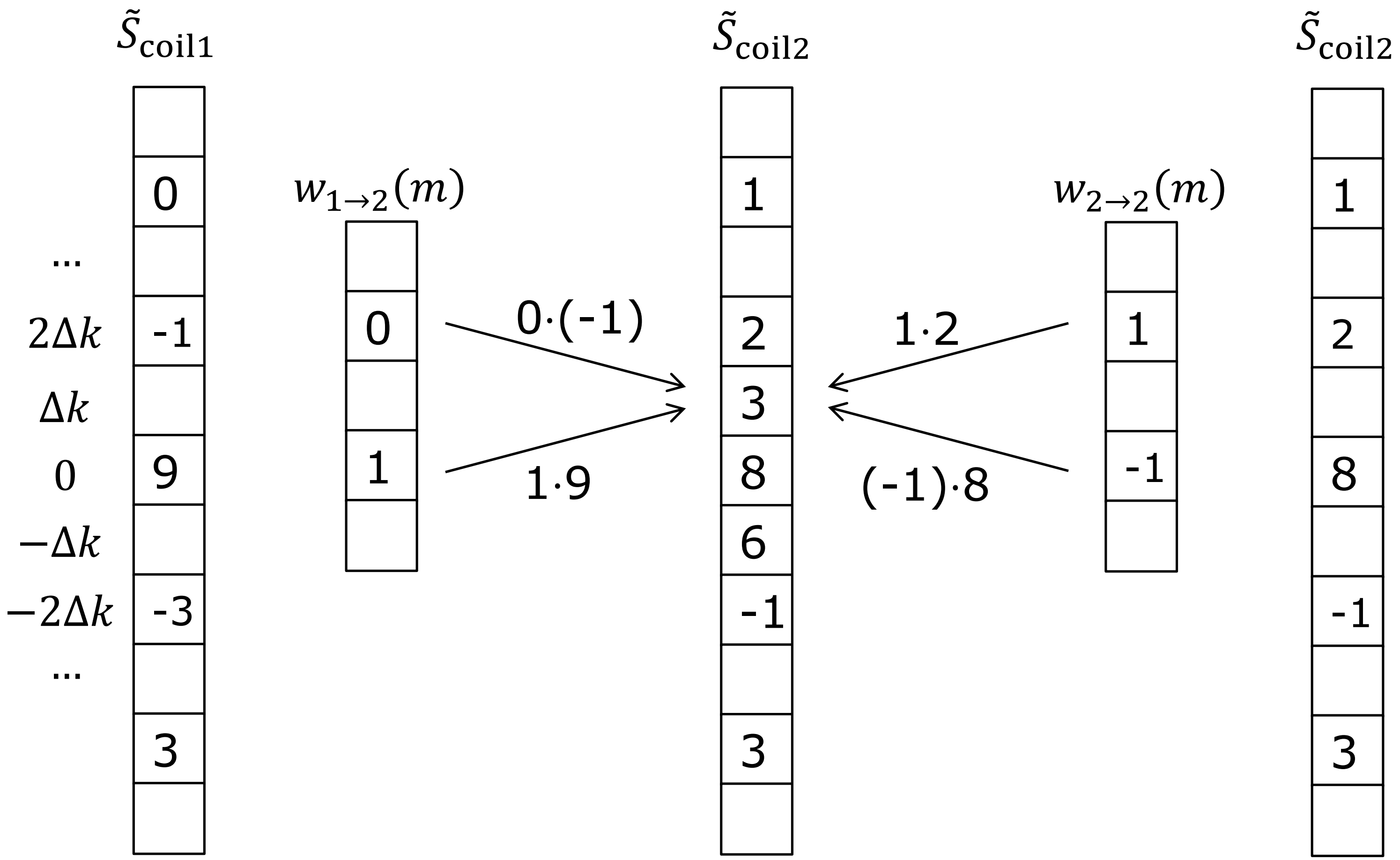


0·9 + 1·(-3) + 1·8 + (-1)·(-1) = 6

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to 1} * \tilde{S}_{\text{coil1}} + w_{2\to 1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to 2} * \tilde{S}_{\text{coil1}} + w_{2\to 2} * \tilde{S}_{\text{coil2}}$$

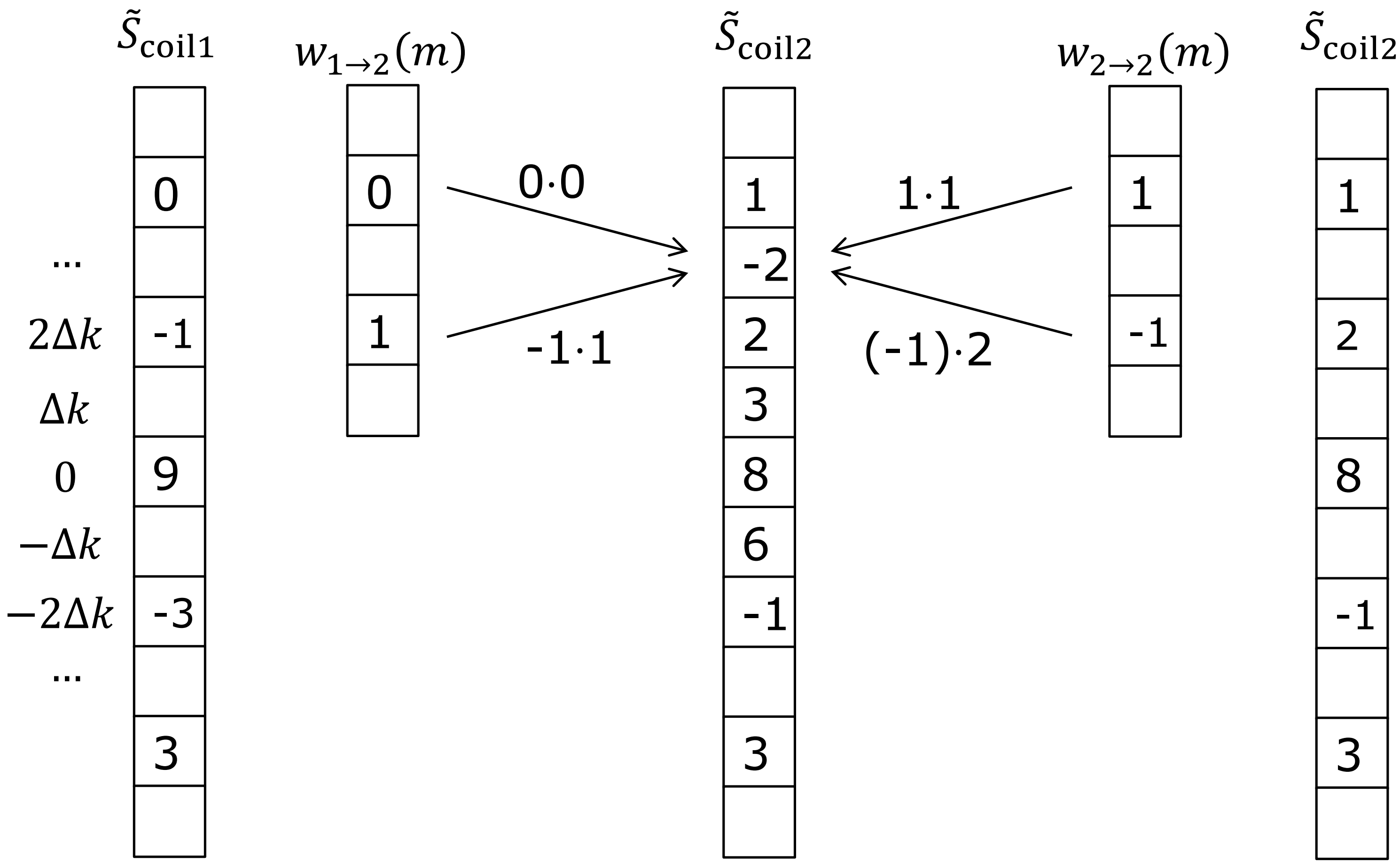


0·(-1) + 1·9 + 1·2 + (-1)·8 = 3

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to1} * \tilde{S}_{\text{coil1}} + w_{2\to1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to2} * \tilde{S}_{\text{coil1}} + w_{2\to2} * \tilde{S}_{\text{coil2}}$$

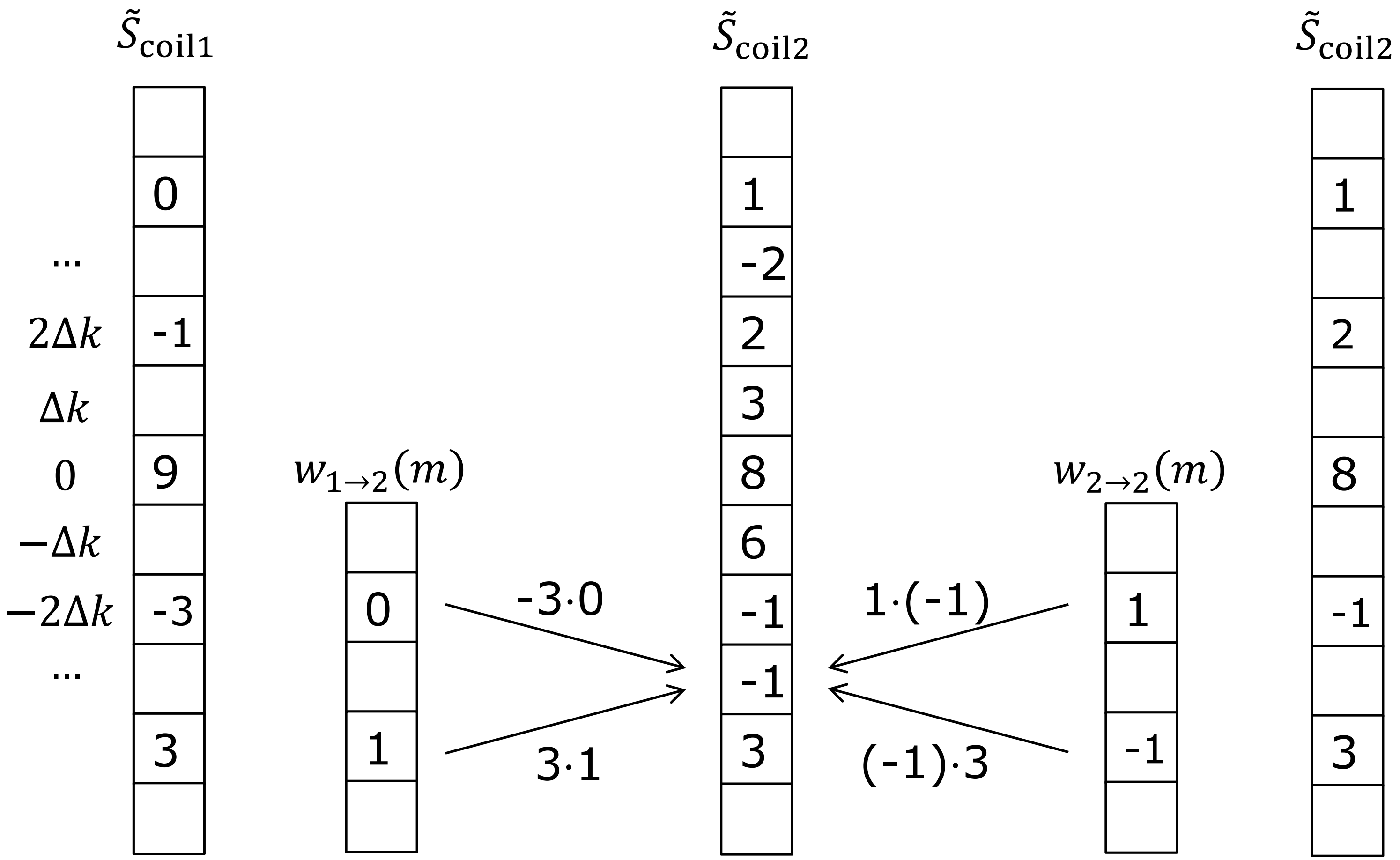


0·0 + (-1)·1 + 1·1 + (-1)·2 = -2

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to1} * \tilde{S}_{\text{coil1}} + w_{2\to1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to2} * \tilde{S}_{\text{coil1}} + w_{2\to2} * \tilde{S}_{\text{coil2}}$$

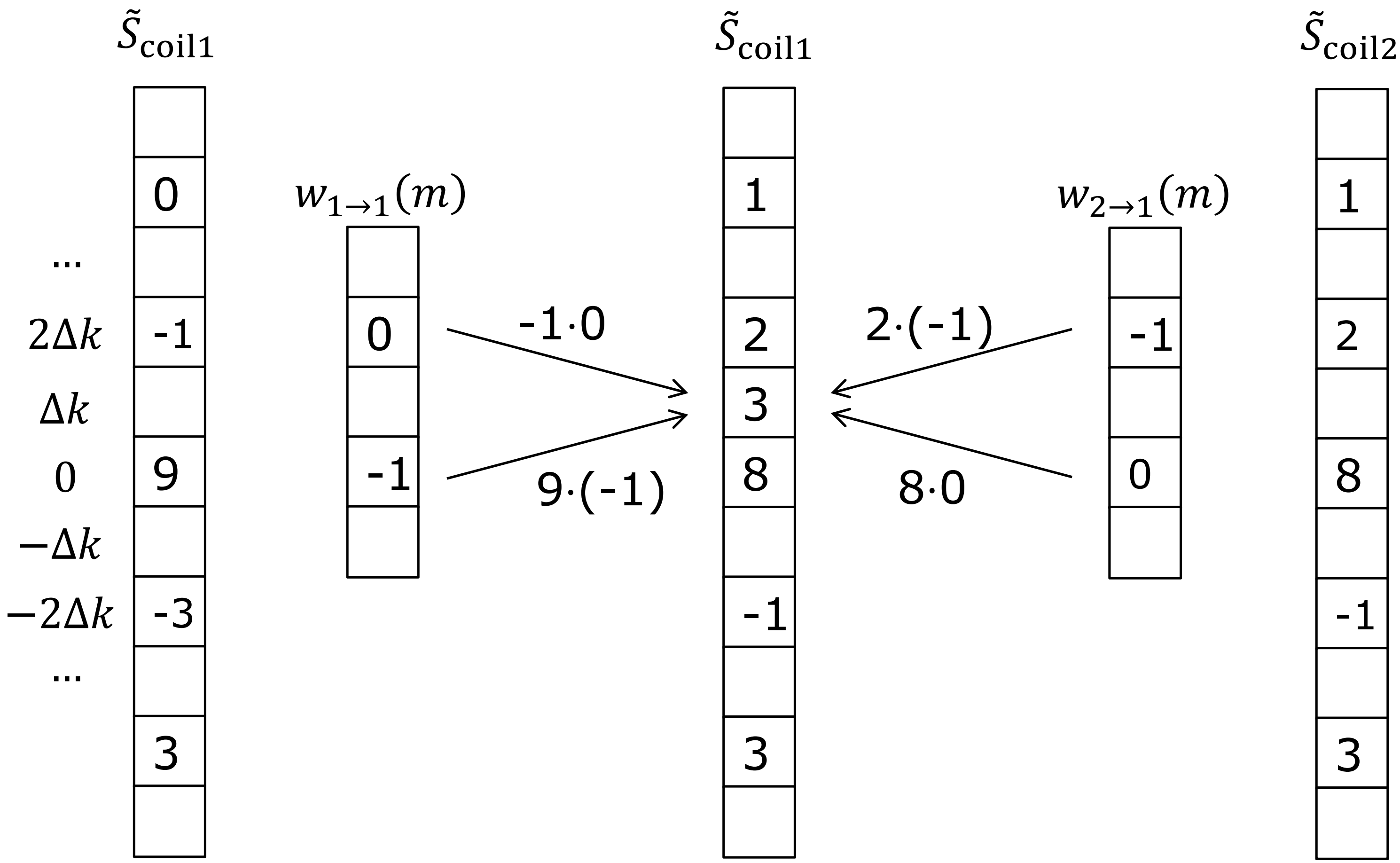


-3·0 + 3·1 + 1·(-1) + (-1)·3 = -2

Signal reconstruction with odd kernel

TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to1} * \tilde{S}_{\text{coil1}} + w_{2\to1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to2} * \tilde{S}_{\text{coil1}} + w_{2\to2} * \tilde{S}_{\text{coil2}}$$



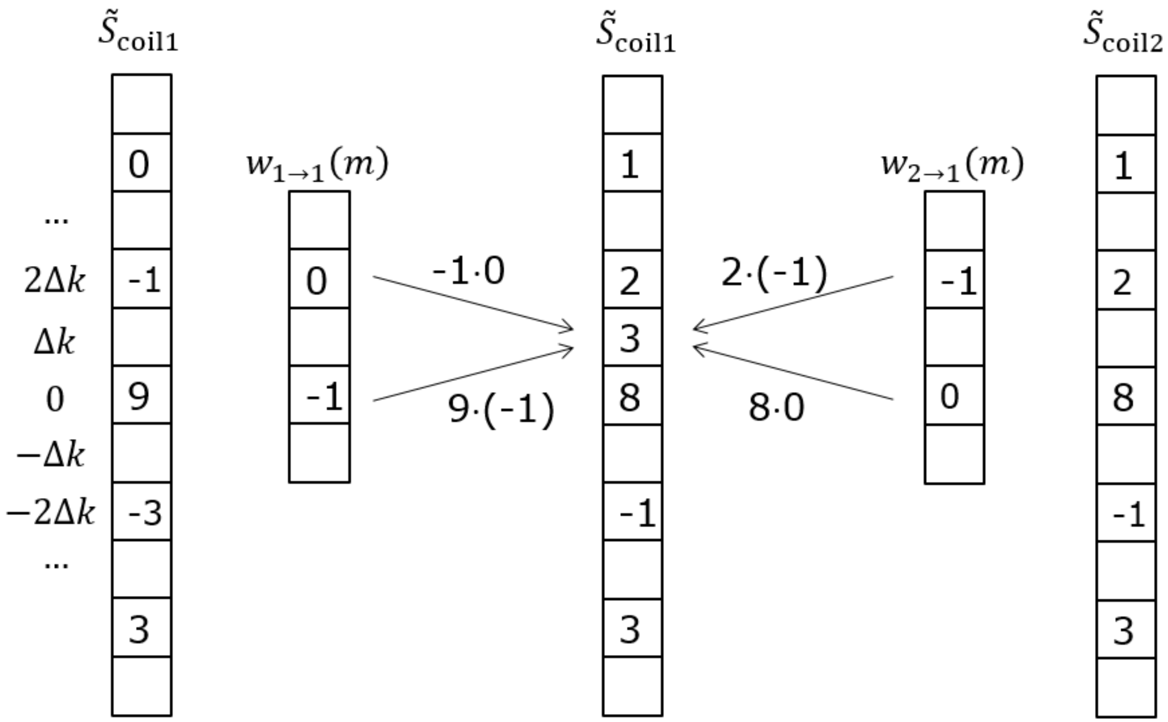
Take care: the numerical values were chosen for demonstration purposes (not truly consistant)

Summary

- Grappa uses a coil to coil fit

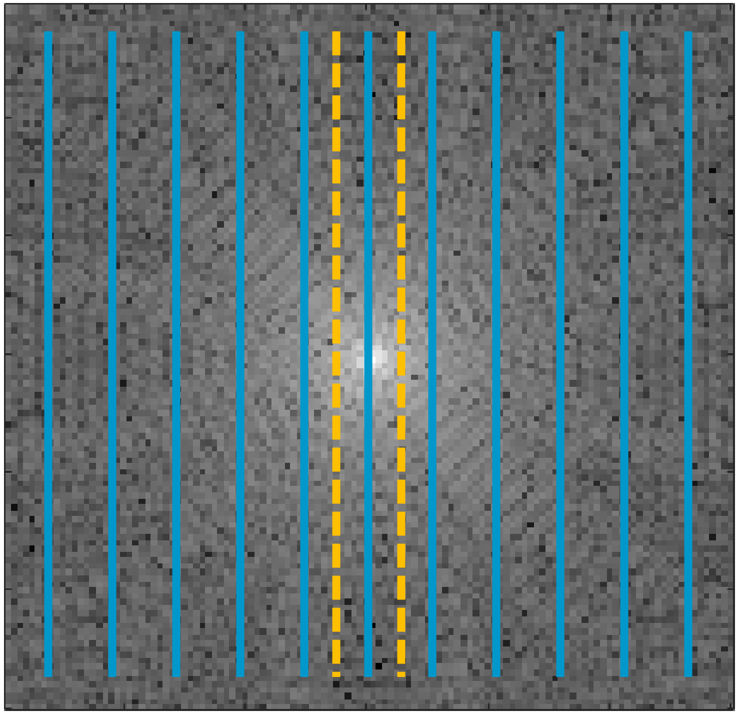
New TRICK:

$$\tilde{S}_{\text{coil1}} = w_{1\to 1} * \tilde{S}_{\text{coil1}} + w_{2\to 1} * \tilde{S}_{\text{coil2}}$$
$$\tilde{S}_{\text{coil2}} = w_{1\to 2} * \tilde{S}_{\text{coil1}} + w_{2\to 2} * \tilde{S}_{\text{coil2}}$$

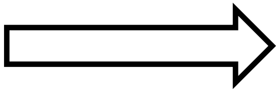


Grappa: Finding the weights

Autocalibration signal (ACS)



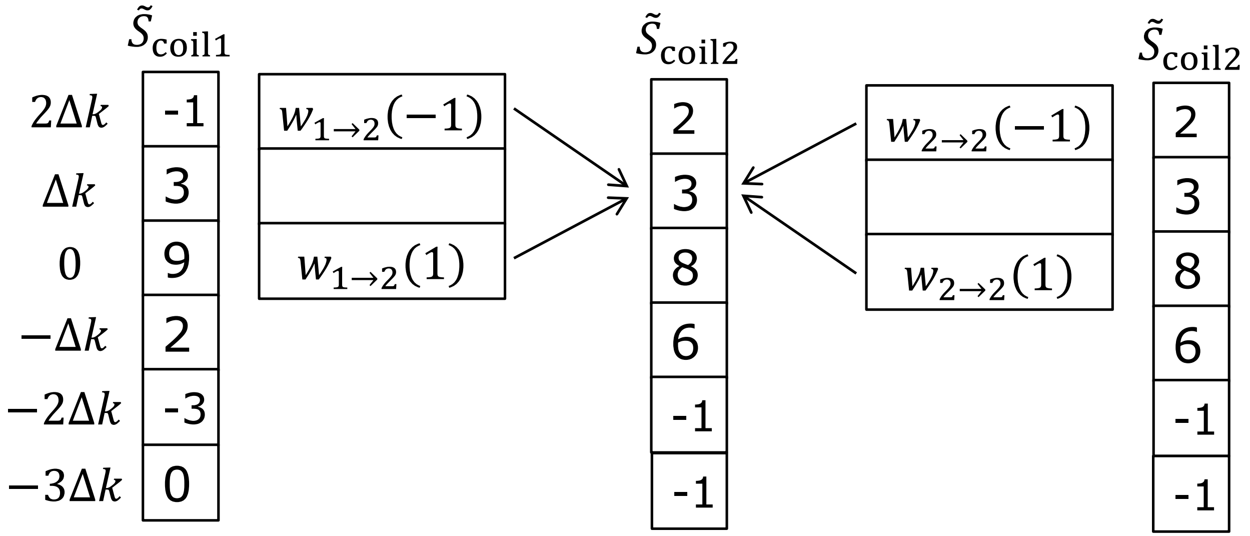
measure additional auto calibration signal (ACS)
(ACS = dashed orange lines)



estimate weights from the ACS

Signal reconstruction with odd kernel

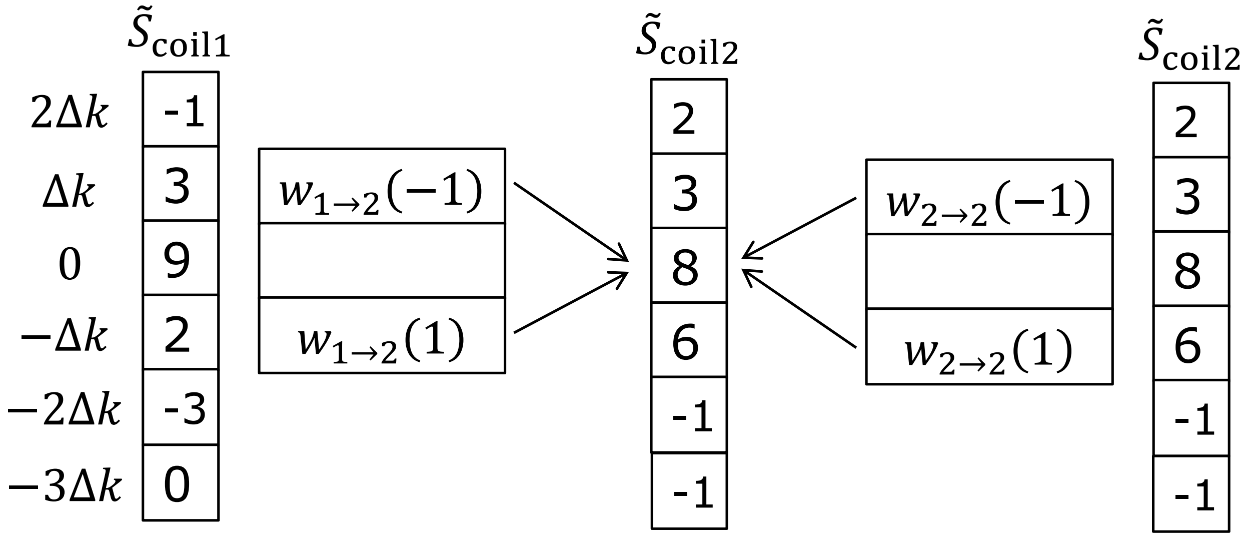
Fully sampled ACS → weights?



$$-1 \cdot w_{1\to2}(-1) + 9 \cdot w_{1\to2}(1) + 2 \cdot w_{2\to2}(-1) + 8 \cdot w_{2\to2}(1) = 3$$

Signal reconstruction with odd kernel

Fully sampled ACS → weights?

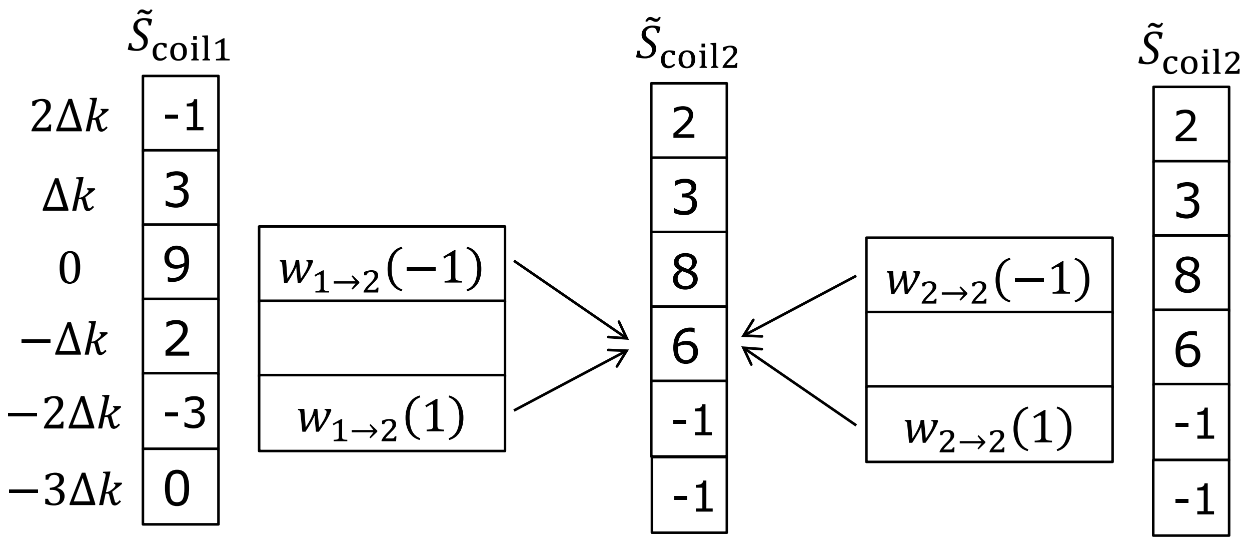


$$-1 \cdot w_{1\to2}(-1) + 9 \cdot w_{1\to2}(1) + 2 \cdot w_{2\to2}(-1) + 8 \cdot w_{2\to2}(1) = 3$$

$$3 \cdot w_{1\to2}(-1) + 2 \cdot w_{1\to2}(1) + 3 \cdot w_{2\to2}(-1) + 6 \cdot w_{2\to2}(1) = 3$$

Signal reconstruction with odd kernel

Fully sampled ACS $\rightarrow$ weights?



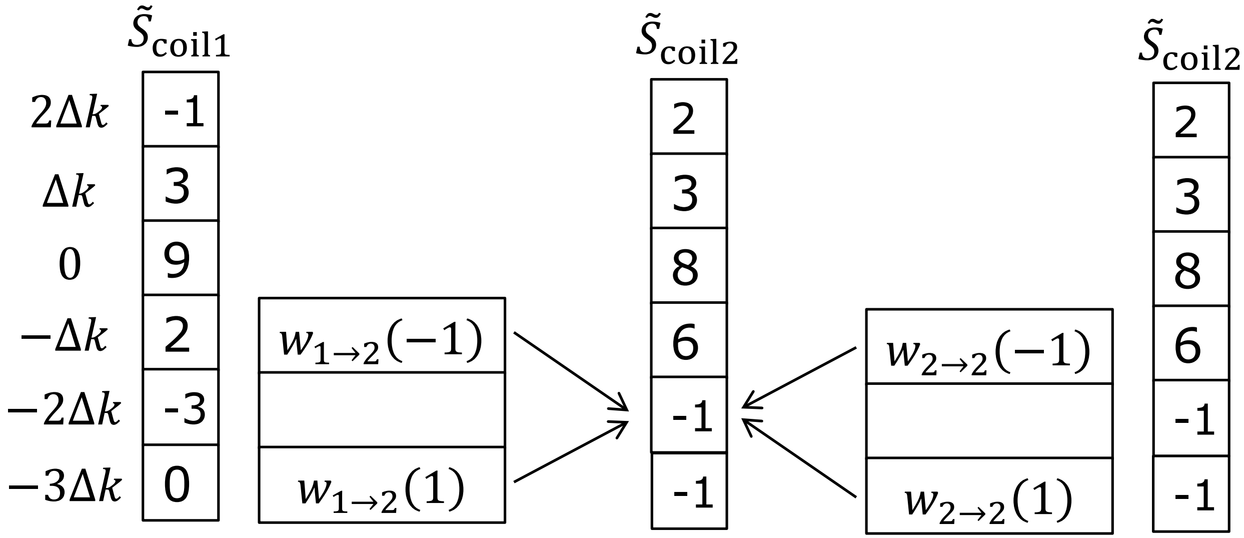
$$-1 \cdot w_{1\to2}(-1) + 9 \cdot w_{1\to2}(1) + 2 \cdot w_{2\to2}(-1) + 8 \cdot w_{2\to2}(1) = 3$$

$$3 \cdot w_{1\to2}(-1) + 2 \cdot w_{1\to2}(1) + 3 \cdot w_{2\to2}(-1) + 6 \cdot w_{2\to2}(1) = 3$$

$$9 \cdot w_{1\to2}(-1) - 3 \cdot w_{1\to2}(1) + 8 \cdot w_{2\to2}(-1) - 1 \cdot w_{2\to2}(1) = 6$$

Signal reconstruction with odd kernel

Fully sampled ACS → weights?



$$-1 \cdot w_{1\to2}(-1) + 9 \cdot w_{1\to2}(1) + 2 \cdot w_{2\to2}(-1) + 8 \cdot w_{2\to2}(1) = 3$$

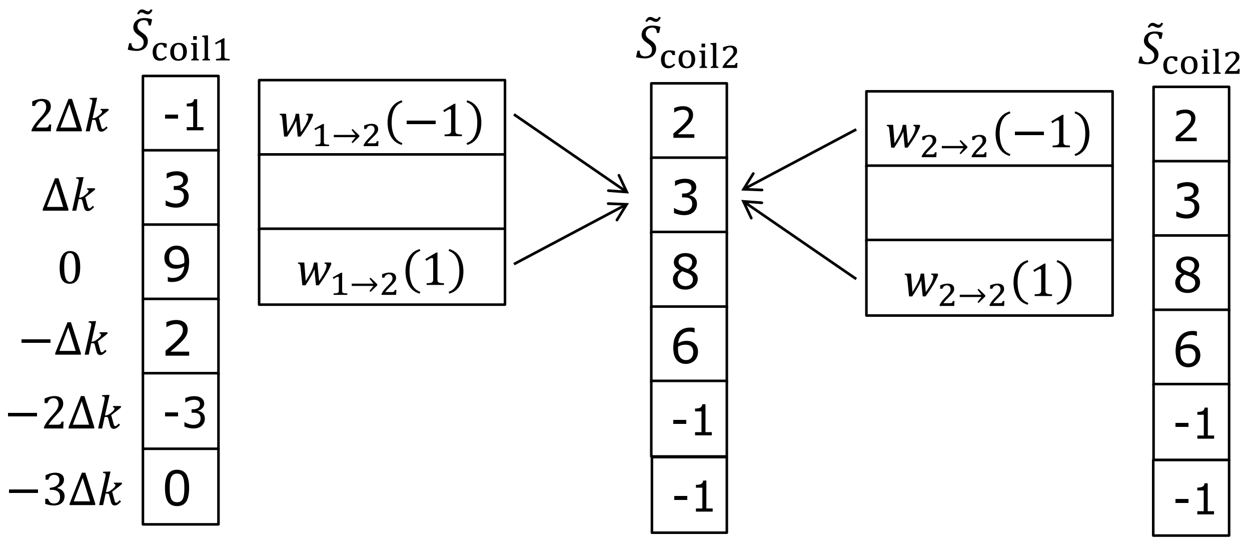
$$3 \cdot w_{1\to2}(-1) + 2 \cdot w_{1\to2}(1) + 3 \cdot w_{2\to2}(-1) + 6 \cdot w_{2\to2}(1) = 3$$

$$9 \cdot w_{1\to2}(-1) - 3 \cdot w_{1\to2}(1) + 8 \cdot w_{2\to2}(-1) - 1 \cdot w_{2\to2}(1) = 6$$

$$2 \cdot w_{1\to2}(-1) + 0 \cdot w_{1\to2}(1) + 6 \cdot w_{2\to2}(-1) - 1 \cdot w_{2\to2}(1) = -1$$

Signal reconstruction with odd kernel

Fully sampled ACS → weights?



$$-1 \cdot w_{1\to2}(-1) + 9 \cdot w_{1\to2}(1) + 2 \cdot w_{2\to2}(-1) + 8 \cdot w_{2\to2}(1) = 3$$

$$\tilde{S}_1(2) \cdot w_{1\to2}(-1) + \tilde{S}_1(0) \cdot w_{1\to2}(1) + \tilde{S}_2(2) \cdot w_{2\to2}(-1) + \tilde{S}_2(0) \cdot w_{2\to2}(1) = \tilde{S}_2(1)$$

Signal reconstruction with odd kernel

$$\tilde{S}_1(2)\cdot w_{1\to2}(-1)+\tilde{S}_1(0)\cdot w_{1\to2}(1)+\tilde{S}_2(2)\cdot w_{2\to2}(-1)+\tilde{S}_2(0)\cdot w_{2\to2}(1)=\tilde{S}_2(1)$$

$$\tilde{S}_1(1)\cdot w_{1\to2}(-1)+\tilde{S}_1(-1)\cdot w_{1\to2}(1)+\tilde{S}_2(1)\cdot w_{2\to2}(-1)+\tilde{S}_2(-1)\cdot w_{2\to2}(1)=\tilde{S}_2(0)$$

$$\tilde{S}_1(0)\cdot w_{1\to2}(-1)+\tilde{S}_1(-2)\cdot w_{1\to2}(1)+\tilde{S}_2(0)\cdot w_{2\to2}(-1)+\tilde{S}_2(-2)\cdot w_{2\to2}(1)=\tilde{S}_2(-1)$$

$$\tilde{S}_1(-1)\cdot w_{1\to2}(-1)+\tilde{S}_1(-3)\cdot w_{1\to2}(1)+\tilde{S}_2(-1)\cdot w_{2\to2}(-1)+\tilde{S}_2(-3)\cdot w_{2\to2}(1)=\tilde{S}_2(-2)$$

Cast in matrix form:

$$\begin{pmatrix}\tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0)\\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1)\\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2)\\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3)\end{pmatrix}\cdot\begin{pmatrix}w_{1\to2}(-1)\\ w_{1\to2}(1)\\ w_{2\to2}(-1)\\ w_{2\to2}(1)\end{pmatrix}=\begin{pmatrix}\tilde{S}_2(1)\\ \tilde{S}_2(0)\\ \tilde{S}_2(-1)\\ \tilde{S}_2(-2)\end{pmatrix}$$

Similarly for coil 1:

$$\begin{pmatrix}\tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0)\\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1)\\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2)\\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3)\end{pmatrix}\cdot\begin{pmatrix}w_{1\to1}(-1)\\ w_{1\to1}(1)\\ w_{2\to1}(-1)\\ w_{2\to1}(1)\end{pmatrix}=\begin{pmatrix}\tilde{S}_1(1)\\ \tilde{S}_1(0)\\ \tilde{S}_1(-1)\\ \tilde{S}_1(-2)\end{pmatrix}$$

Signal reconstruction with odd kernel

Coil 2:

$$\begin{pmatrix} \tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0) \\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1) \\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2) \\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3) \end{pmatrix} \cdot \begin{pmatrix} w_{1\to 2}(-1) \\ w_{1\to 2}(1) \\ w_{2\to 2}(-1) \\ w_{2\to 2}(1) \end{pmatrix} = \begin{pmatrix} \tilde{S}_2(1) \\ \tilde{S}_2(0) \\ \tilde{S}_2(-1) \\ \tilde{S}_2(-2) \end{pmatrix}$$

Coil 1:

$$\begin{pmatrix} \tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0) \\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1) \\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2) \\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3) \end{pmatrix} \cdot \begin{pmatrix} w_{1\to 1}(-1) \\ w_{1\to 1}(1) \\ w_{2\to 1}(-1) \\ w_{2\to 1}(1) \end{pmatrix} = \begin{pmatrix} \tilde{S}_1(1) \\ \tilde{S}_1(0) \\ \tilde{S}_1(-1) \\ \tilde{S}_1(-2) \end{pmatrix}$$

Combine:

$$\begin{pmatrix} \tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0) \\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1) \\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2) \\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3) \end{pmatrix} \cdot \begin{pmatrix} w_{1\to 1}(-1) & w_{1\to 2}(-1) \\ w_{1\to 1}(1) & w_{1\to 2}(1) \\ w_{2\to 1}(-1) & w_{2\to 2}(-1) \\ w_{2\to 1}(1) & w_{2\to 2}(1) \end{pmatrix} = \begin{pmatrix} \tilde{S}_1(1) & \tilde{S}_2(1) \\ \tilde{S}_1(0) & \tilde{S}_2(0) \\ \tilde{S}_1(-1) & \tilde{S}_2(-1) \\ \tilde{S}_1(-2) & \tilde{S}_2(-2) \end{pmatrix}$$

Signal reconstruction with odd kernel

Combine:

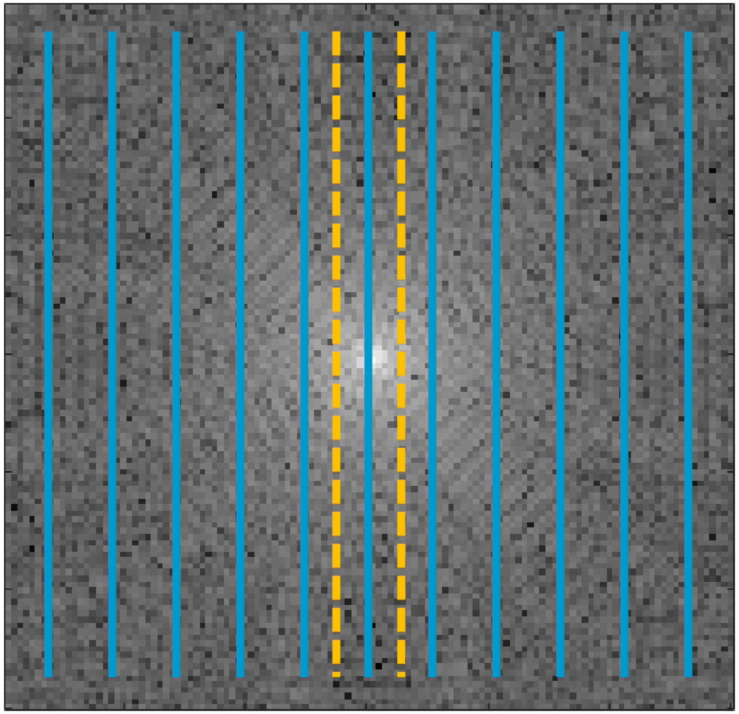
$$\begin{pmatrix} \tilde{S}_1(2) & \tilde{S}_1(0) & \tilde{S}_2(2) & \tilde{S}_2(0) \\ \tilde{S}_1(1) & \tilde{S}_1(-1) & \tilde{S}_2(1) & \tilde{S}_2(-1) \\ \tilde{S}_1(0) & \tilde{S}_1(-2) & \tilde{S}_2(0) & \tilde{S}_2(-2) \\ \tilde{S}_1(-1) & \tilde{S}_1(-3) & \tilde{S}_2(-1) & \tilde{S}_2(-3) \end{pmatrix} \cdot \begin{pmatrix} w_{1\to 1}(-1) & w_{1\to 2}(-1) \\ w_{1\to 1}(1) & w_{1\to 2}(1) \\ w_{2\to 1}(-1) & w_{2\to 2}(-1) \\ w_{2\to 1}(1) & w_{2\to 2}(1) \end{pmatrix} = \begin{pmatrix} \tilde{S}_1(1) & \tilde{S}_2(1) \\ \tilde{S}_1(0) & \tilde{S}_2(0) \\ \tilde{S}_1(-1) & \tilde{S}_2(-1) \\ \tilde{S}_1(-2) & \tilde{S}_2(-2) \end{pmatrix}$$

Short-hand: $S_{\text{ACS,matrix1}} \cdot W = S_{\text{ACS,matrix2}}$

$\rightarrow W = S_{\text{ACS,matrix1}}^{-1} \cdot S_{\text{ACS,matrix2}}$

Summary

Autocalibration signal (ACS)



estimate weights from the ACS

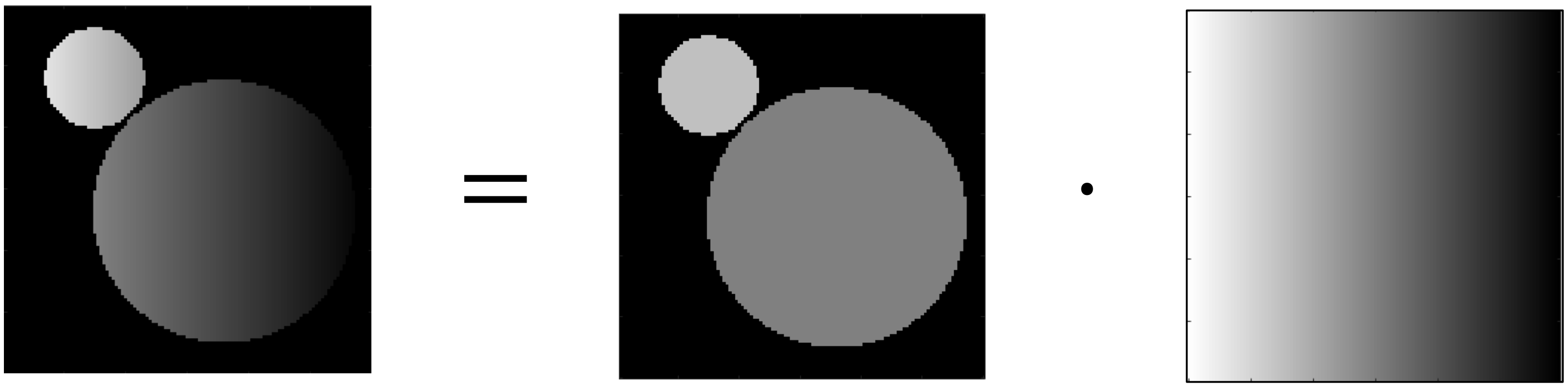
$$W = S_{\text{ACS,matrix1}}^{-1} \cdot S_{\text{ACS,matrix2}}$$

Grappa: 2D

So far: 1D

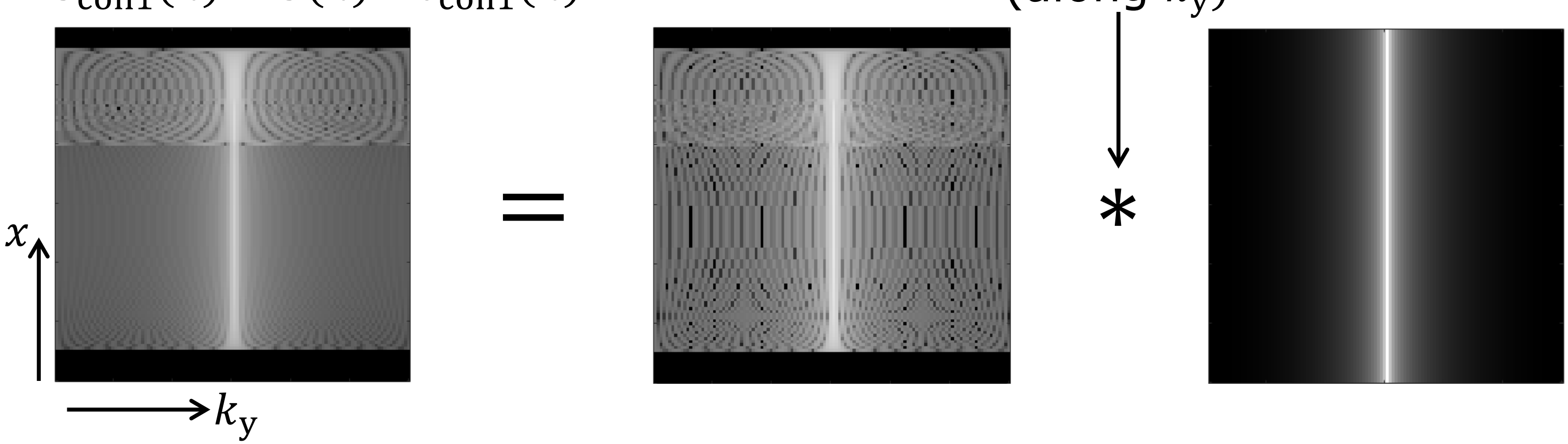
This was essentially a 1D sensitivity profile

$S_{\mathrm{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot s_{\mathrm{coil1}}(\boldsymbol{r})$



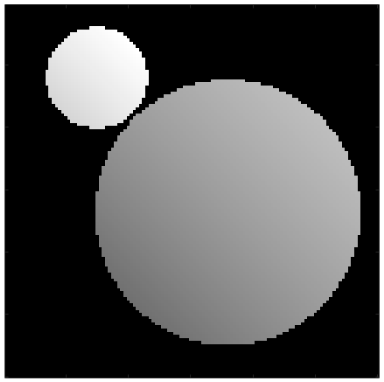
$\tilde{S}_{\mathrm{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{s}_{\mathrm{coil1}}(\boldsymbol{k})$

(along k_{y})

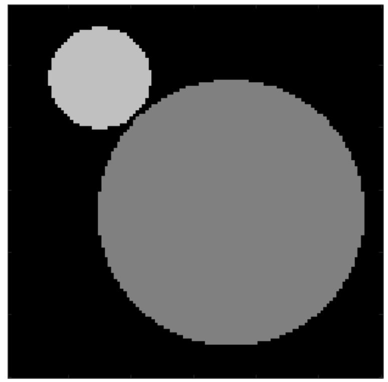


Now 2D

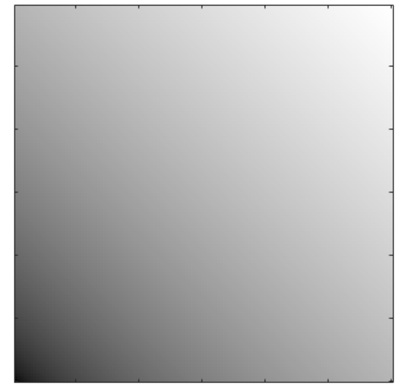
$S_{\text{coil1}}(\boldsymbol{r}) = S(\boldsymbol{r}) \cdot \mathcal{s}_{\text{coil1}}(\boldsymbol{r})$



$=$



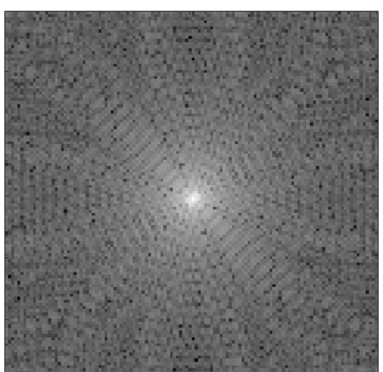
$\cdot$



$\tilde{S}_{\text{coil1}}(\boldsymbol{k}) = \tilde{S}(\boldsymbol{k}) * \tilde{\mathcal{s}}_{\text{coil1}}(\boldsymbol{k})$

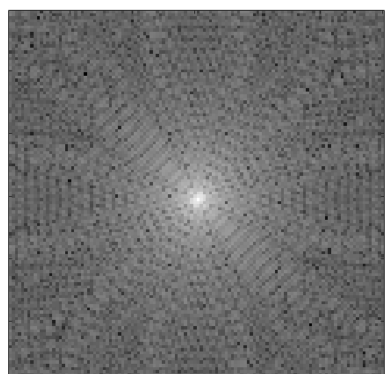
(along k_{x} and k_{y})

k_{x}

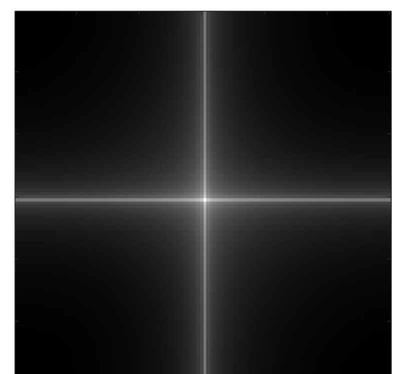


k_{y}

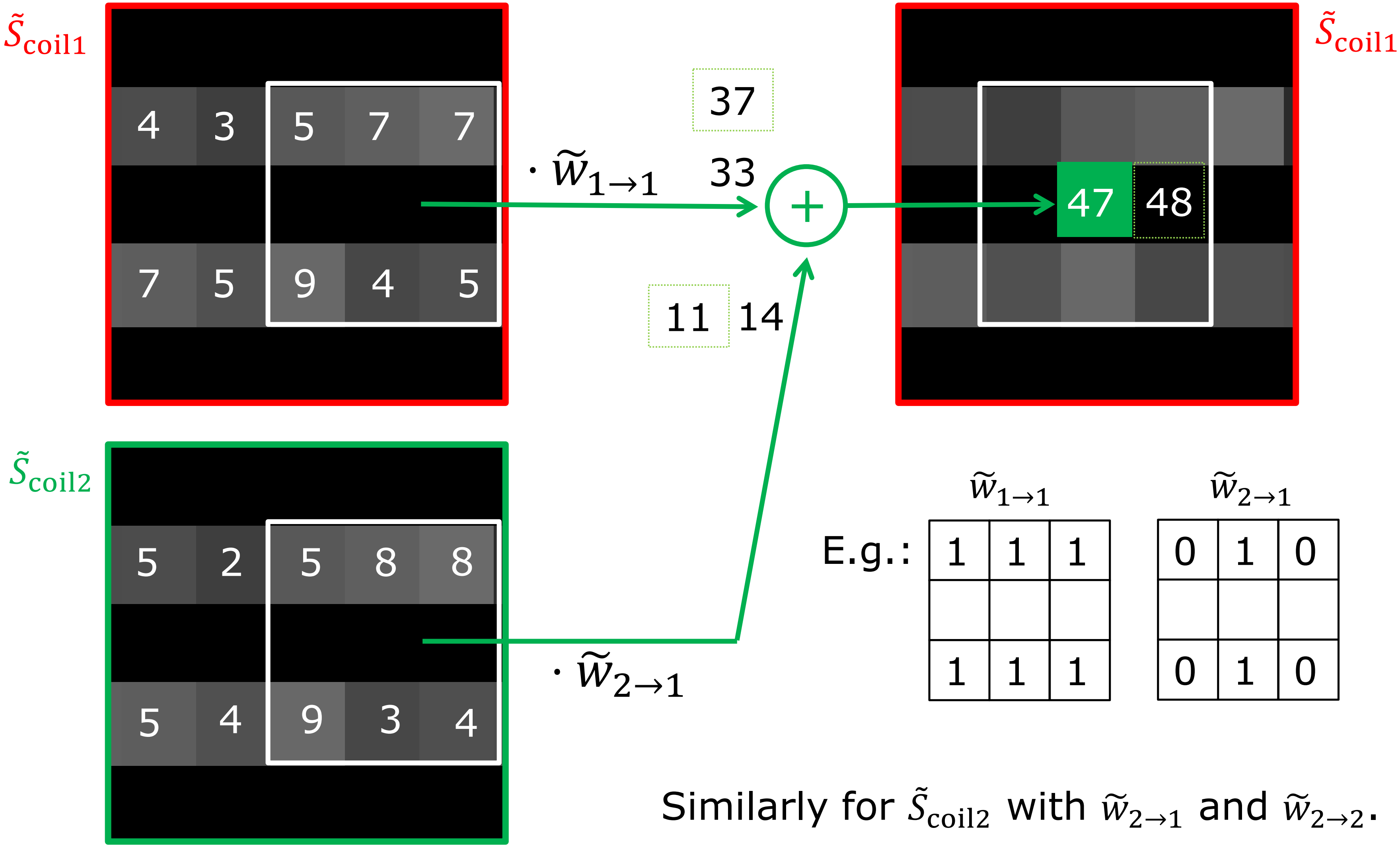
$=$



$*$



Visualization: The kernel becomes also 2D



E.g.:

$\tilde{w}_{1\to1}$

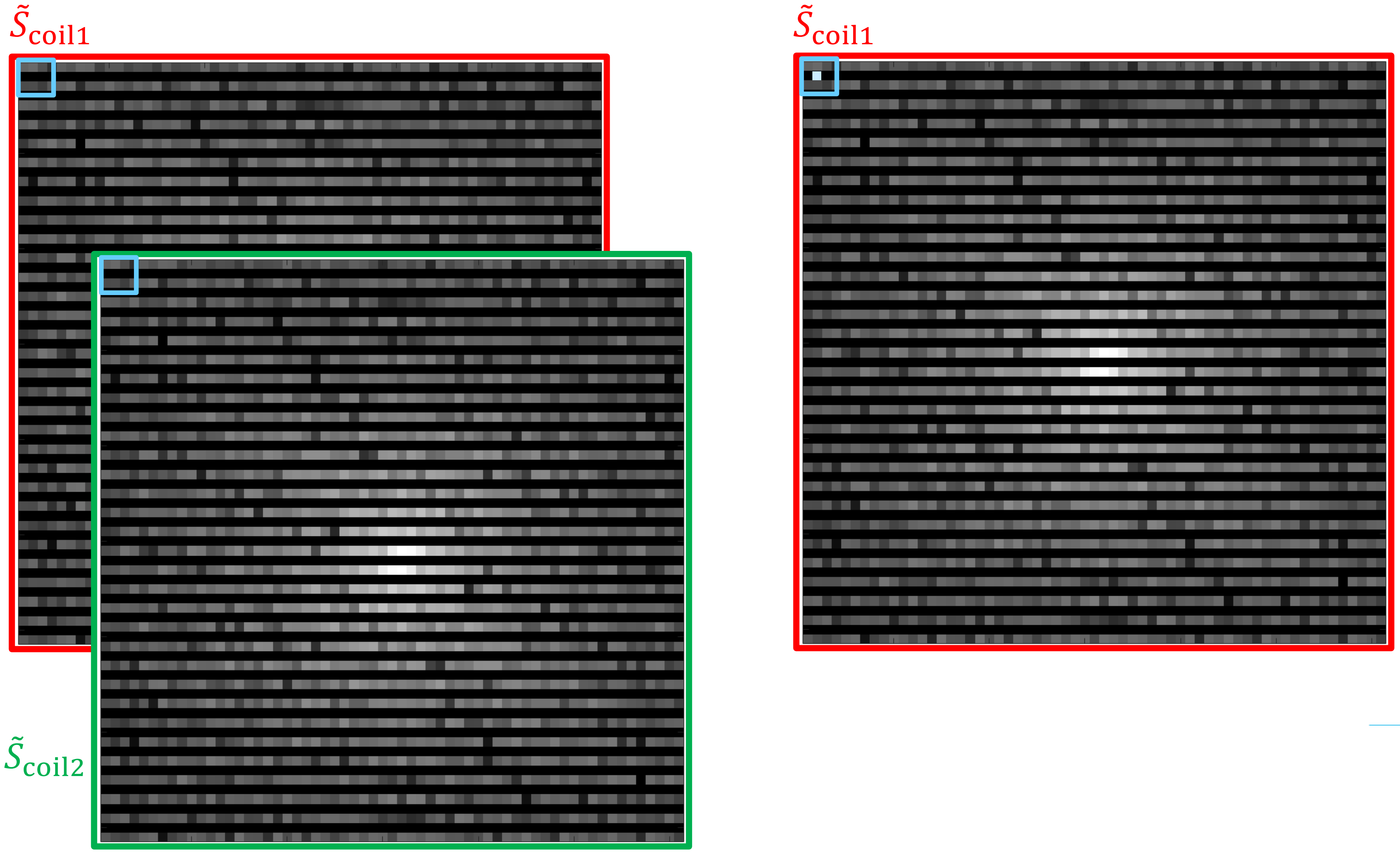
1	1	1
1	1	1

$\tilde{w}_{2\to1}$

0	1	0
0	1	0

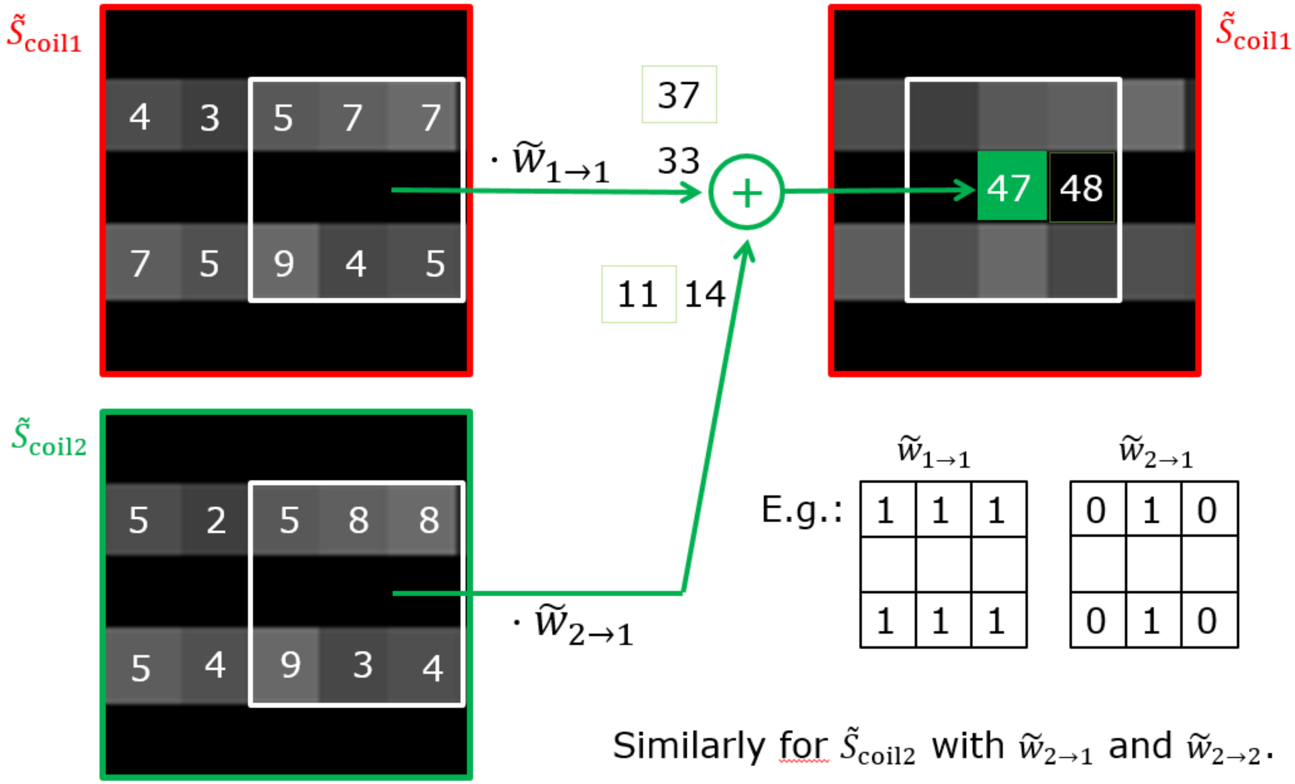
Similarly for $\tilde{S}_{\text{coil2}}$ with $\tilde{w}_{2\to1}$ and $\tilde{w}_{2\to2}$.

Reconstruction of all missing data (this is a convolution-like operation)



Summary

- Adaptation to 2D in principle simple



E.g.:

$\widetilde{w}_{1\to 1}$

1	1	1
1	1	1

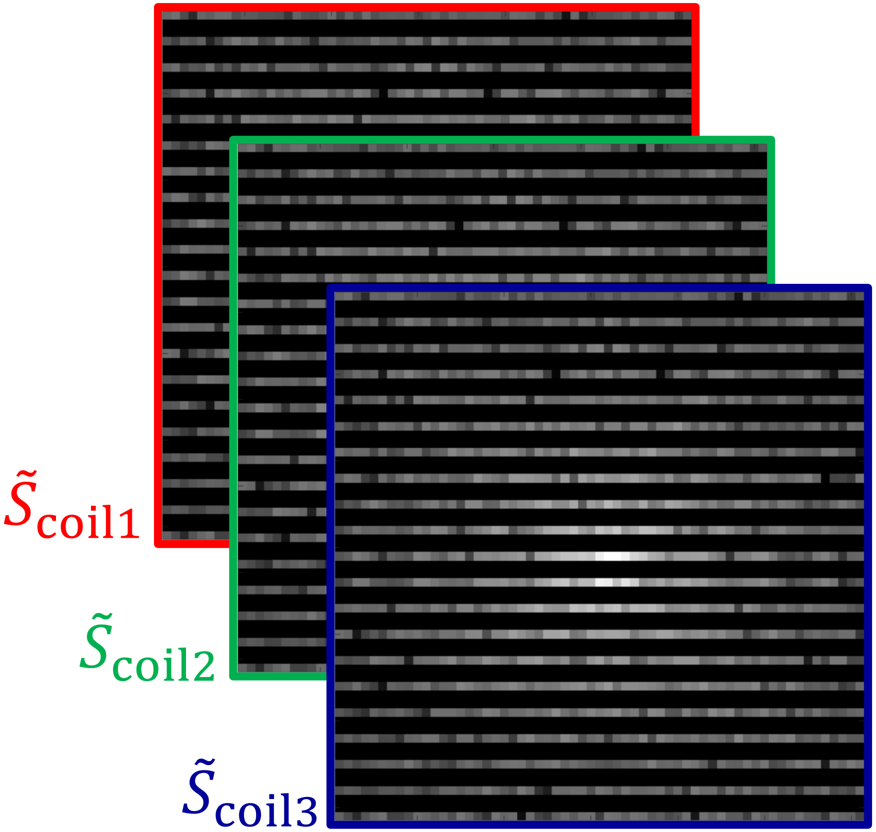
$\widetilde{w}_{2\to 1}$

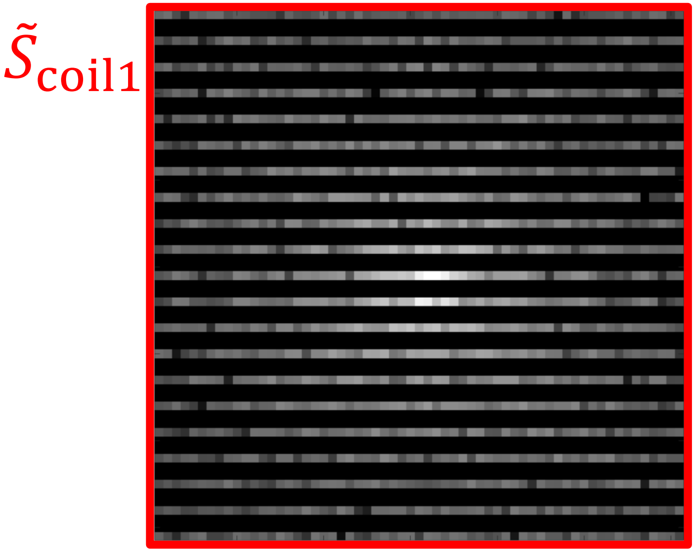
0	1	0
0	1	0

Similarly for $\tilde{S}_{\mathrm{coil2}}$ with $\widetilde{w}_{2\to 1}$ and $\widetilde{w}_{2\to 2}$.

Grappa: $R > 2$

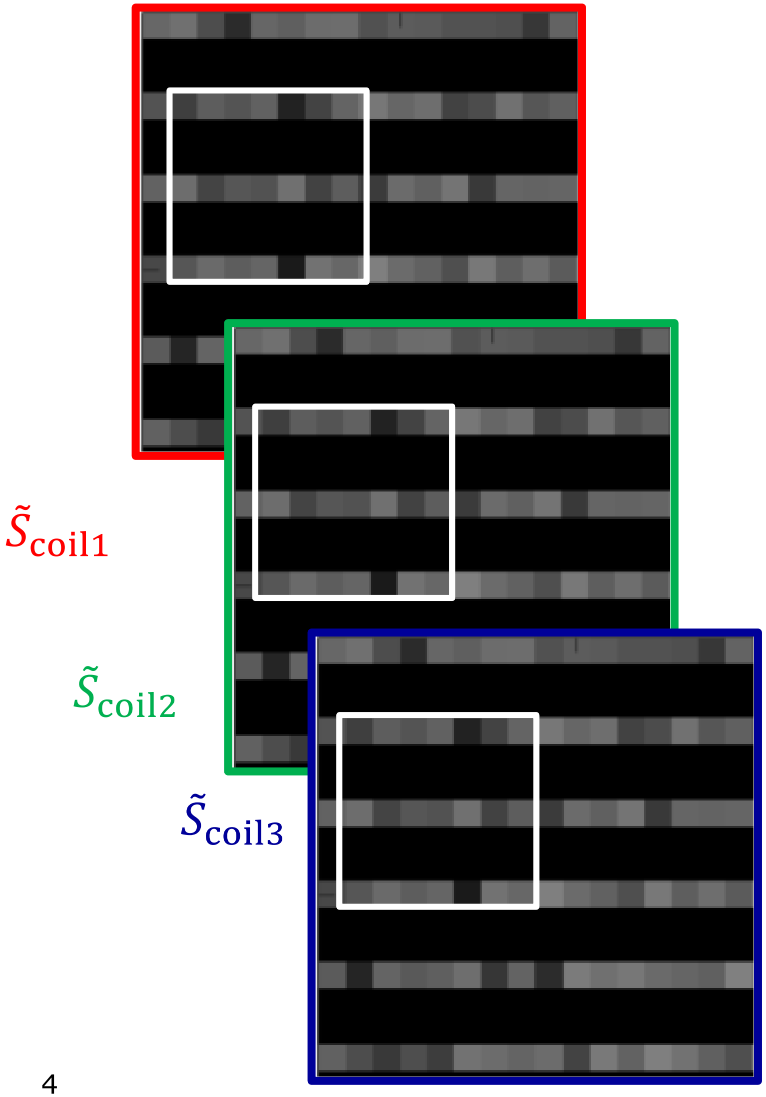
Grappa: $R > 2$

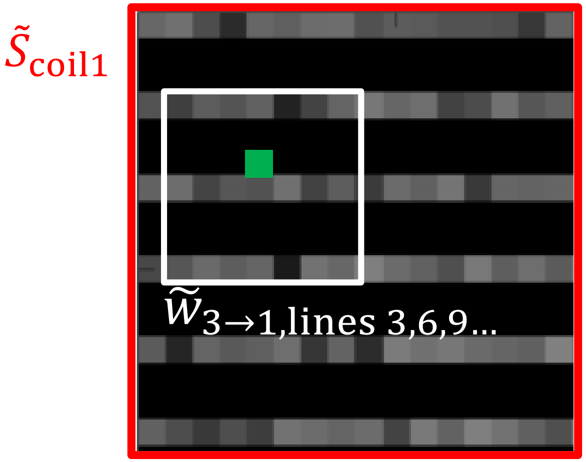




- On needs at least three coils
- More weights: e.g. $\tilde{w}_{3\to 1}$

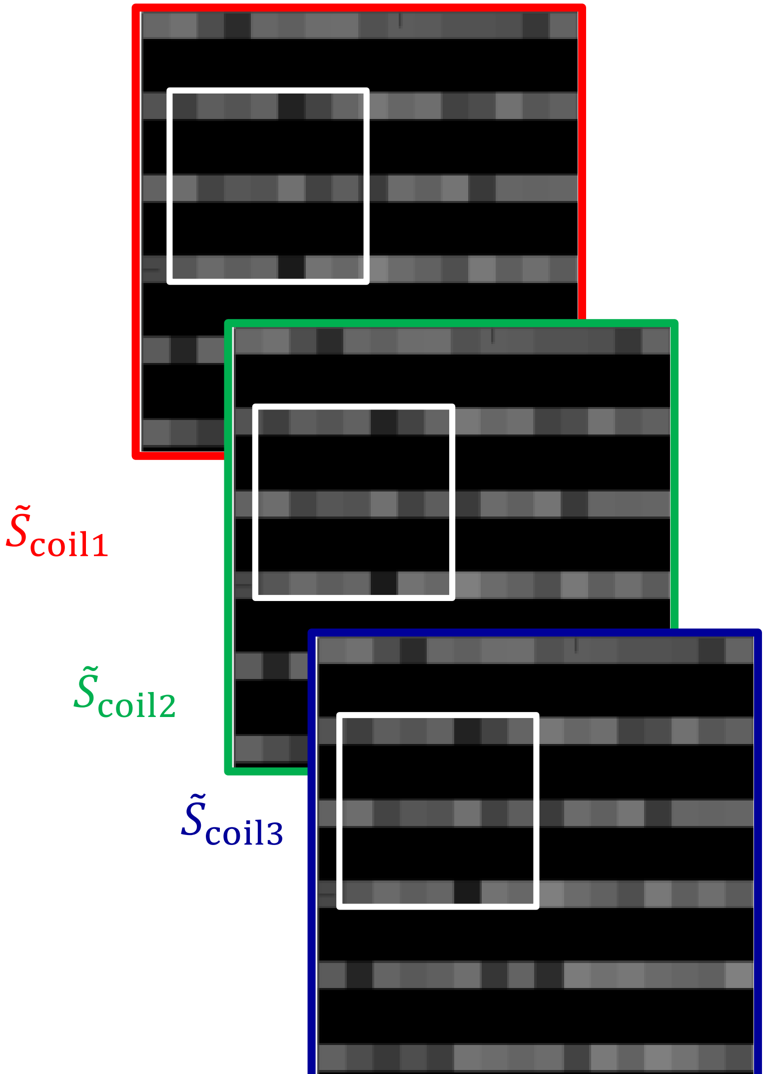
Grappa: $R > 2$

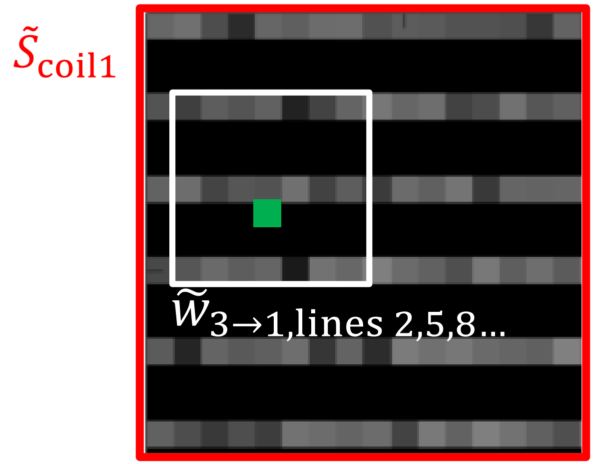




- Even more weights:
  - E.g. $\widetilde{w}_{3\to1,\text{lines }3,6,9\ldots}$

Grappa: $R > 2$





- Even more weights:
  - E.g. $\widetilde{w}_{3\to 1,\text{lines } 3,6,9\ldots}$
  - E.g. $\widetilde{w}_{3\to 1,\text{lines } 2,5,8\ldots}$

Grappa: Final Steps

Finals steps

- Compute $S_{\mathrm{coil1}}(\boldsymbol{r})$, $S_{\mathrm{coil2}}(\boldsymbol{r})$, … from $\tilde{S}_{\mathrm{coil1}}(\boldsymbol{k})$, $\tilde{S}_{\mathrm{coil2}}(\boldsymbol{k})$, …
- Combine $S_{\mathrm{coil1}}(\boldsymbol{r})$, $S_{\mathrm{coil2}}(\boldsymbol{r})$, … into a single image
  - E.g. via a sum of squares operation

Grappa: Pros & Cons

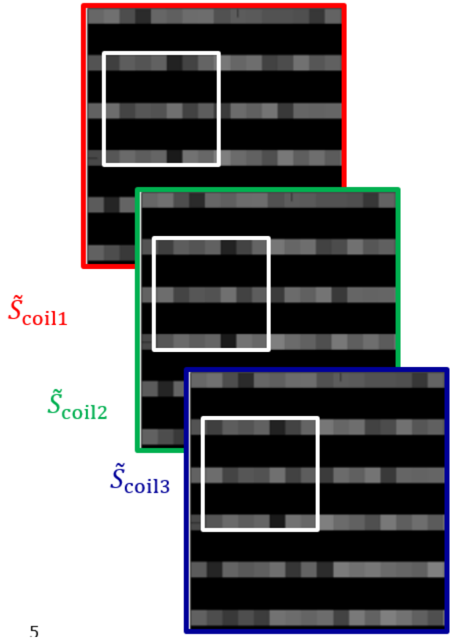
Pros & Cons

- Good: No sensivity profile is needed
- They can be difficult to obtain at times
  - E.g. when motion is present (breathing, cardiac, …)
- The ACS lines can be used as actual image data

- If the sensitivity profiles are known, SENSE may perform better
  - The truncation of the kernel (which is an approximation) is not needed with SENSE

Summary

- $R > 2$ possible



- Final steps: Compute single-coil images
- Combine them into a single image

- Grappa is used a lot