Total variation regularized nonlinear inversion for parallel MRI with variable density sampling patterns

Christian Clason[1] Florian Knoll[2]

[1] Institute for Mathematics and Scientific Computing, Karl-Franzens-Universität Graz

[2] Institute of Medical Engineering, Graz University of Technology

Workshop on Novel Reconstruction Strategies in NMR and MRI
Göttingen, September 11, 2010

Parallel MRI as inverse problem

Given

- sampling operator $\mathcal{F}_s$ (defined by trajectory)
- acquired k-space coil data $g = (g_1, \ldots, g_N)^T$

Find

- spin density u
- coil sensitivities $c = (c_1, \ldots, c_N)^T$

such that

$$F(u, c) := (\mathcal{F}_S(u \cdot c_1), \ldots, \mathcal{F}_S(u \cdot c_N))^T = g$$

nonlinear inverse problem, ill-posed $\rightsquigarrow$ solve using IRGN method

Iteratively regularized Gauß-Newton method

Choose $x^0 = (u^0, c^0)$, α_0, $q < 1$
repeat
  Solve for $\delta x = (\delta u, \delta c)$ (e.g., by CG on normal equations)

$$\min_{\delta x} \frac{1}{2}\|F'(x^k)\delta x + F(x^k) - g\|^2 + \frac{\alpha_k}{2}\|W(c^k + \delta c)\|^2 + \frac{\alpha_k}{2}\|u^k + \delta u\|^2$$

  Set $x^{k+1} = x^k + \delta x$, $\alpha_{k+1} = \alpha_k q$, $k = k + 1$
until $\|F(x^k) - g\| < tol$

W high-order differential operator (enforces smooth sensitivities)
F' Fréchet derivative with adjoint F'^*

Nonlinear inverse problem approach

Advantage:

Flexibility in

- Sampling strategy (choice of $\mathcal{F}_S$)
- Incorporation of a priori information (choice of penalty)
- Minimization method (choice of gradient descent method requiring only application of $\mathcal{F}_s, \mathcal{F}_s^T$)

Disadvantage:

Can be less efficient than specialized methods

Choice of sampling strategy

Trajectory should:

1. Minimize acquisition time
   $\rightsquigarrow$ traverse only part of k-space
2. Minimize subsampling artifacts
   $\rightsquigarrow$ denser sampling of center of k-space (auto-calibration)
3. Allow fast reconstruction
   $\rightsquigarrow$ availability of (N)FFT

Here:

- radial sampling
- adapted Cartesian random sampling

Cartesian random sampling

Advantages of Cartesian random sampling patterns:

- Easy to implement: standard FFT/gradients + binary mask
- Incoherent aliasing artifacts
- Allows non-uniform sampling by non-uniform probability for sampling points

Open question: Good choice for non-uniform probability (how to sample middle frequencies?)

Idea: look at coefficient distribution of (reasonably similar) template images (only magnitude important, not phase!)

Adapted Cartesian random sampling

Procedure

1: choose template image u_t (same anatom. region, resolution)

2: set $p = |\mathcal{F} u_t|$, (apply smoothing/averaging,) rescale

3: **repeat**

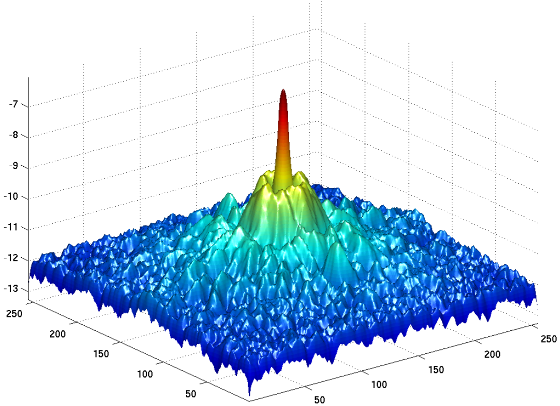
4: draw sampling points from Cartesian grid points using Monte Carlo method with p.d.f. p

5: **until** desired acceleration factor is reached

6: (add postprocessing to avoid holes)

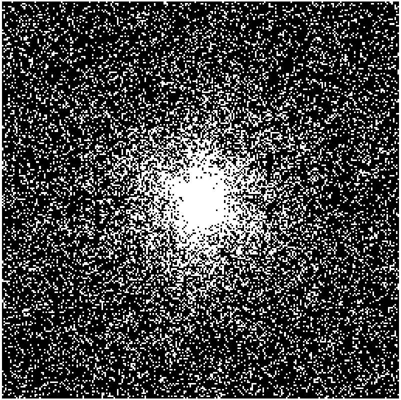
Main advantage: Good results without parameter tuning, robust

Adapted random sampling: Example

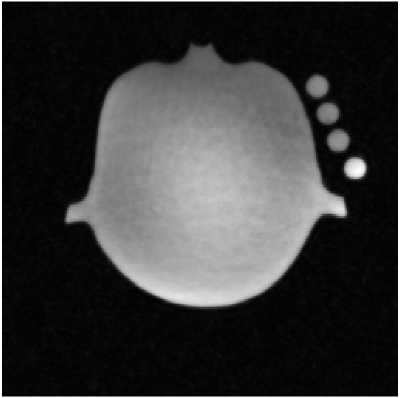


log-plot of probability density function (generated from raw data)

Adapted random sampling: Example

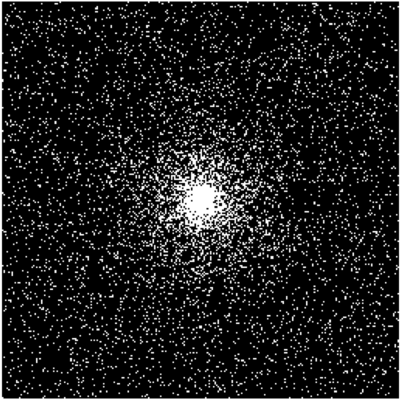


(a) pattern $R = 4$

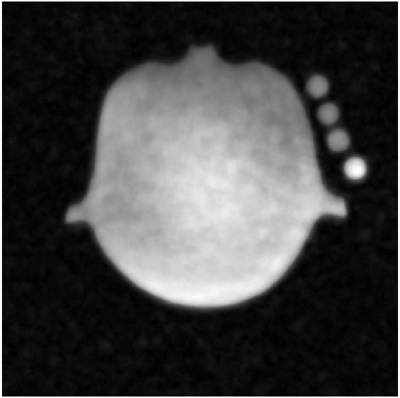


(b) zero-filled SOS (no dens. comp.)

Adapted random sampling: Example

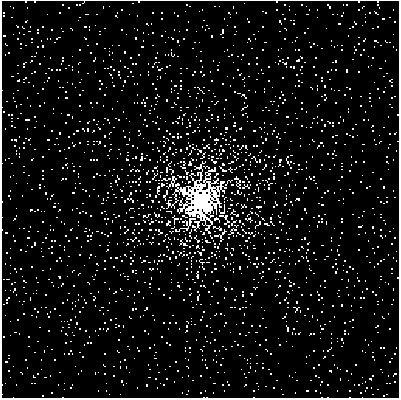


(c) pattern $R = 10$

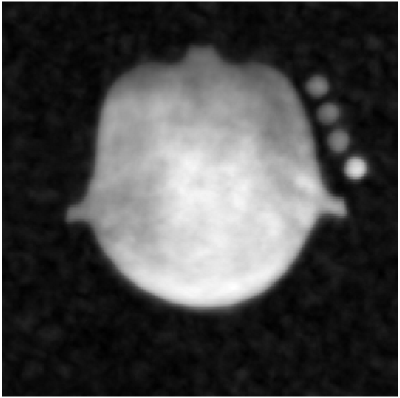


(d) zero-filled SOS (no dens. comp.)

Adapted random sampling: Example



(e) pattern $R = 18$



(f) zero-filled SOS (no dens. comp.)

Choice of penalty

- IRGN suffers from noise amplification when α_k too small
- aliasing artifacts are incoherent, noise-like

$\rightsquigarrow$ add stronger penalty for image content

Here:

Total variation

$$TV(u) = \int |\nabla u|_2 \, dx$$

Pro: preserves edges while removing smooth variations

Con: non-quadratic, non-differentiable

IRGNTV

Replace L^2 penalty on u^{k+1} with TV:

Choose $x^0 = (u^0, c^0)$, α_0, β_0, $q < 1$

repeat

  Solve for $\delta x = (\delta u, \delta c)$

$$\min_{\delta x} \frac{1}{2}\|F'(x^k)\delta x + F(x^k) - g\|^2 + \frac{\alpha_k}{2}\|W(c^k + \delta c)\|^2 + \beta_k TV(u^k + \delta u)$$

  Set $x^{k+1} = x^k + \delta x$, $\alpha_{k+1} = \alpha_k q$, $\beta_{k+1} = \beta_k q$, $k = k + 1$

until $\|F(x^k) - g\| < tol$

return u, c

Solution of TV subproblems

Set $J(\delta x) := \frac{1}{2}\|F'(x^k)\delta x + F(x^k) - g\|^2 + \frac{\alpha_k}{2}\|W(c^k + \delta c)\|^2$

Step 3

$$\min_{\delta u, \delta c} J(\delta u, \delta c) + \beta_k TV(u^k + \delta u)$$

non-smooth, convex optimization problem $\rightsquigarrow$ use **convex duality**

$$\beta TV(u) = \sup_{\{|p(x)|_2 \leq \beta\}} \langle u, -\operatorname{div} p \rangle$$

Solution of TV subproblems

Saddle point problem

$$\min_{\delta u, \delta c} \max_{p \in C_{\beta_k}} J(\delta u, \delta c) + \langle u^k + \delta u, -\operatorname{div} p \rangle$$

with $C_\beta = \{p : |p(x)|_2 \leq \beta \text{ for all } x\}$ convex, J differentiable

$\rightsquigarrow$ use **projected gradient descent/ascent method**:

- Requires only application of F', F'^* (i.e., $\mathcal{F}_s, \mathcal{F}_s^*$)
- Straightforward parallelization
- Order-optimal algorithms available

Here: **Primal-dual extragradient algorithm**, based on *Pock/Cremers/Bischof/Chambolle* (2009)

Primal-dual extragradient method

function $\text{TV}\textsc{solve}(u, c, \alpha, \beta, \sigma_u, \sigma_c, \tau)$
  $\delta u, \overline{\delta u}, \delta c, \overline{\delta c}, p \leftarrow 0$
  repeat
    $p \leftarrow \text{proj}_\beta(p + \tau\nabla(u + \overline{\delta u}))$
    $\delta u_{old} \leftarrow \delta u, \ \delta c_{old} \leftarrow \delta c$
    $\delta u \leftarrow \delta u - \sigma_u(\partial_u J(u, c)(\overline{\delta u}, \overline{\delta c}) - \text{div}\, p)$
    $\delta c \leftarrow \delta c - \sigma_c(\partial_c J(u, c)(\overline{\delta u}, \overline{\delta c}))$
    $\overline{\delta u} \leftarrow 2\delta u - \delta u_{old}$
    $\overline{\delta c} \leftarrow 2\delta c - \delta c_{old}$
  until convergence
  return $\delta u, \delta c$
end function

Algorithm

- Compute projection on C_β pointwise by

$$\mathrm{proj}_\beta(q)(x) = \frac{q(x)}{\max(1, \beta^{-1}|q(x)|_2)}$$

- Computation of $\partial_u J(u,c)(\overline{\delta u}, \overline{\delta c})$ and $\partial_c J(u,c)(\overline{\delta u}, \overline{\delta c})$ identical to CG iteration for IRGN

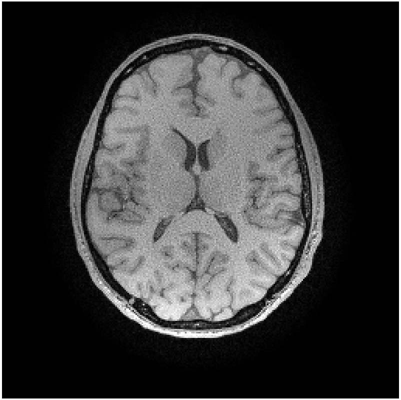
details

- Step lengths σ_u, σ_c, τ related to Lipschitz constants of $F'(u^k, c^k), \nabla$

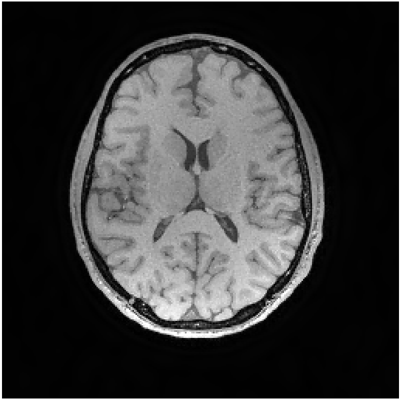
Examples: random sampling

- raw data fom brain and phantom
- 3D gradient echo sequence, 3T system, 12 channel head coil
- 8 (phantom: 9) virtual channels (SVD) used for reconstruction
- sequence modified using binary 2D mask to define subsampling pattern
- subsampling $R = 4$ (10)
- sequence parameters
  - repetition time TR=20ms
  - echo time TE=5ms
  - flip angle FA=18°
  - matrix size (x,y,z)=256x256x256
  - FOV=250mm
  - slice thickness brain 1mm (phantom 5mm)

Reconstructions: random ($R = 4$)

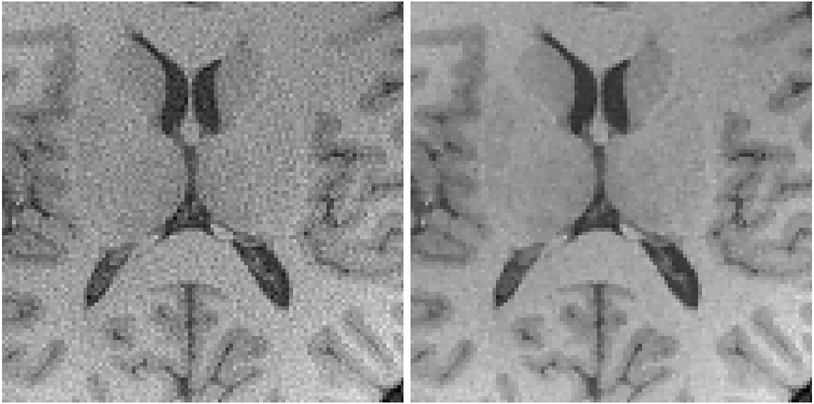


(a) IRGN



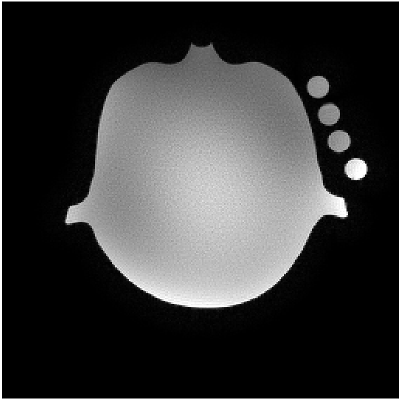
(b) IRGNTV

Reconstructions: random ($R = 4$)

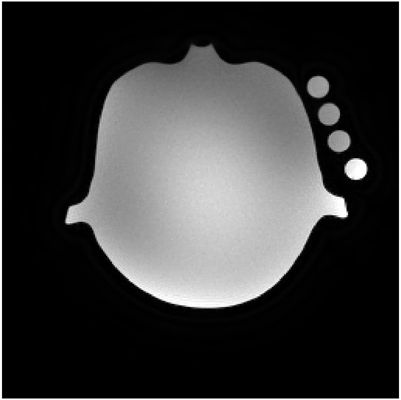


(a) IRGN (detail) (b) IRGNTV (detail)

Reconstructions: random ($R = 4$)



(a) IRGN



(b) IRGNTV

Effect of TV

Since $\beta_k \to 0$, final TV effect is not very strong

Pro: No introduction of typical TV-artifacts (cartooning, stair-casing)

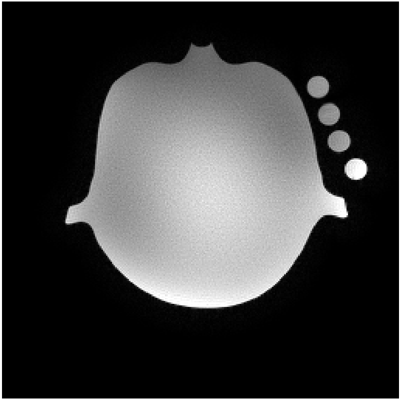
Con: Strong effect can be desired if piecewise constant is a good prior (i.e., for higher acceleration, cf. phantom)

$\rightsquigarrow$ stop decreasing TV penalty parameter at desired value:

$$\alpha_{k+1} = \alpha_k q$$
$$\beta_{k+1} = \max(\beta_{\min}, \beta_k q)$$

For illustration: Phantom with $\beta_{\min} = 5 \cdot 10^{-3}$

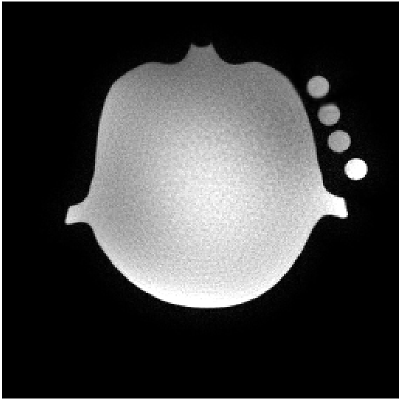
Effect of TV ($R = 4$)



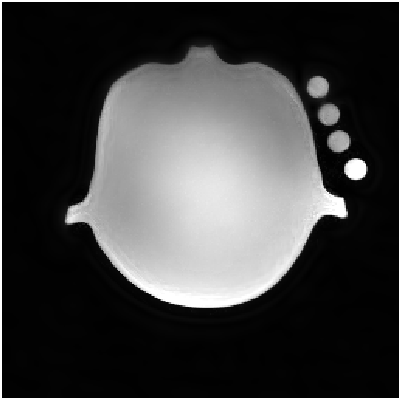
(a) IRGN

(b) IRGNTV

Effect of TV ($R = 10$)



(a) IRGN



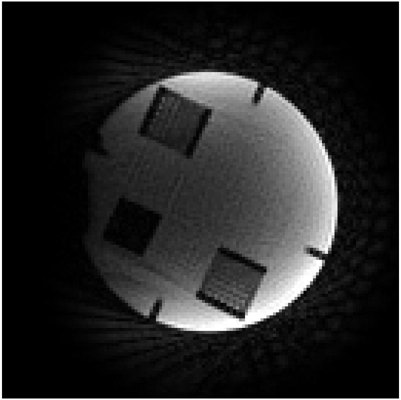
(b) IRGNTV

Examples: radial sampling

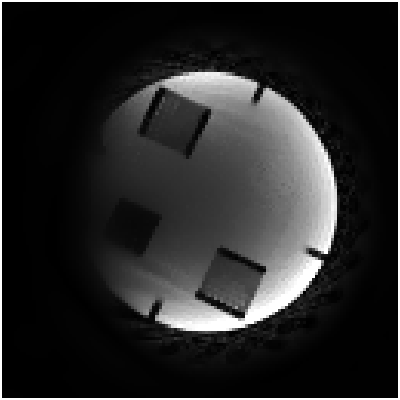
- raw data of phantom and heart
- radial FLASH sequence, 3T System, 32 channel coil
- 8 (cardiac: 12) virtual channels (SVD) used for reconstruction
- 25 (19) projections, $R \approx 8$ (10.5)
- **No postprocessing, temporal view sharing**
- sequence parameters
  - repetition time TR=2.0ms
  - echo time TE=1.3ms
  - flip angle FA=8°
  - 256 points per proj. (2x oversampling) $\rightsquigarrow$ matrix 128x128
  - slice thickness 8mm, in plane resolution 2mm x 2mm

(data courtesy of Martin Uecker)

Radial sampling: phantom (25 proj)

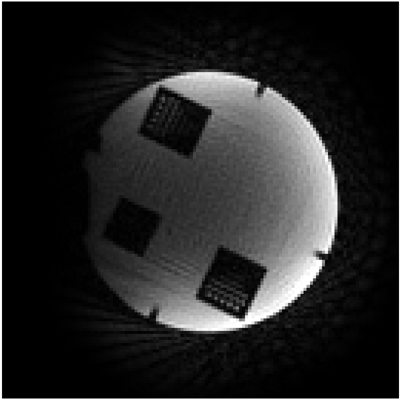


(a) IRGN

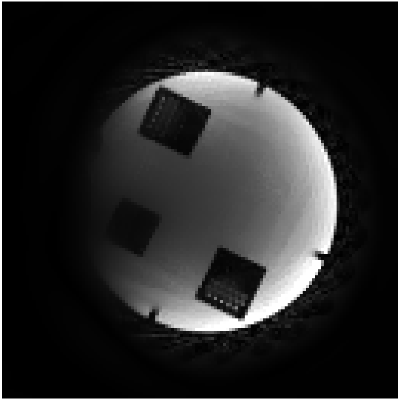


(b) IRGNTV

Radial sampling: phantom (25 proj)

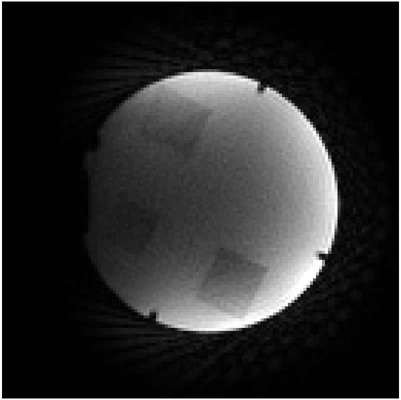


(a) IRGN

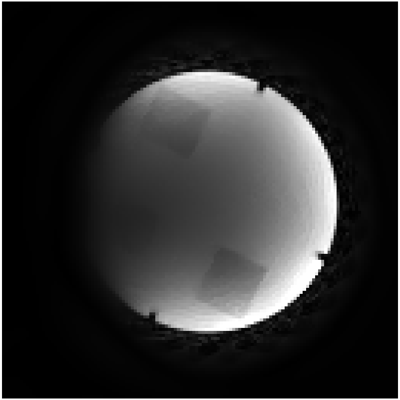


(b) IRGNTV

Radial sampling: phantom (25 proj)

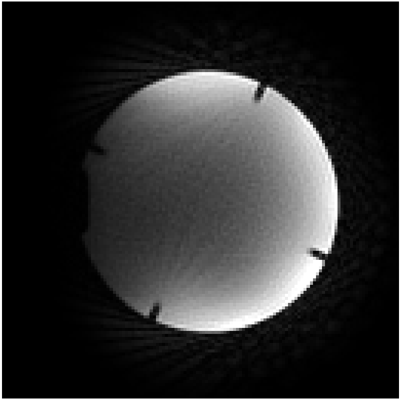


(a) IRGN

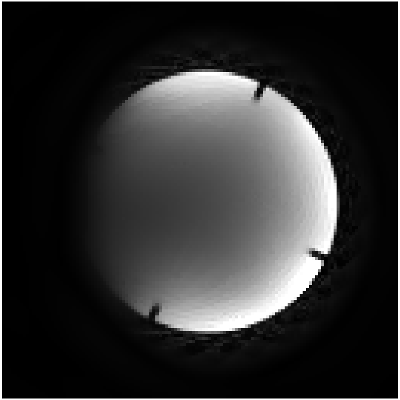


(b) IRGNTV

Radial sampling: phantom (25 proj)

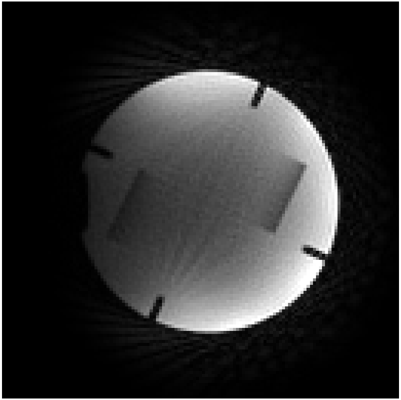


(a) IRGN

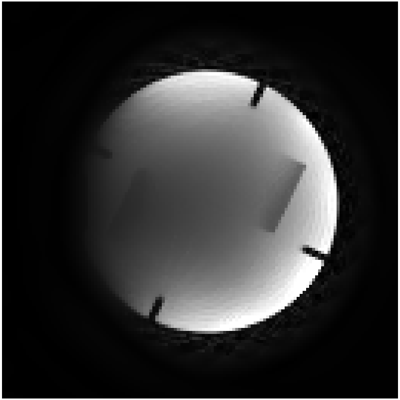


(b) IRGNTV

Radial sampling: phantom (25 proj)

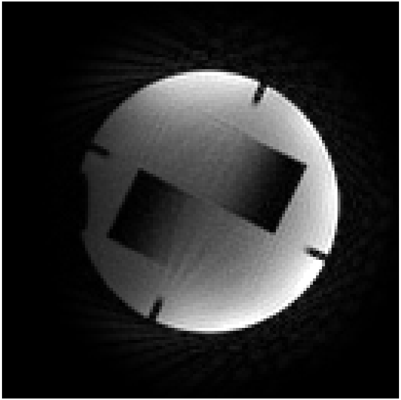


(a) IRGN

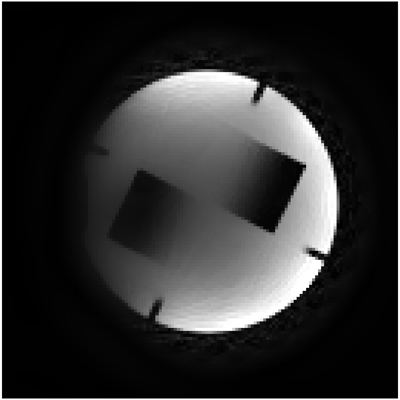


(b) IRGNTV

Radial sampling: phantom (25 proj)

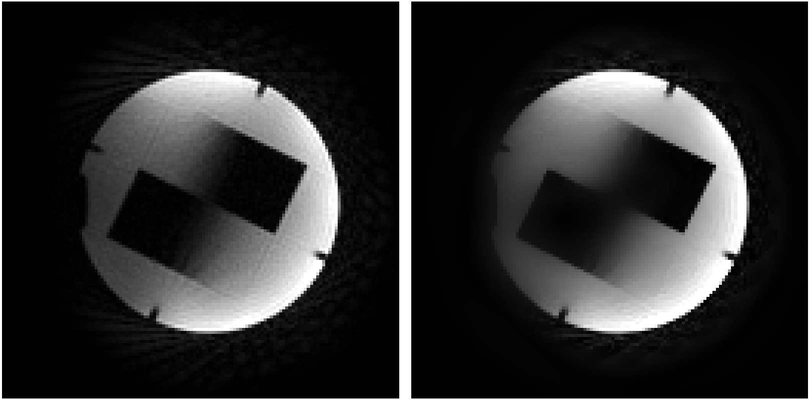


(a) IRGN



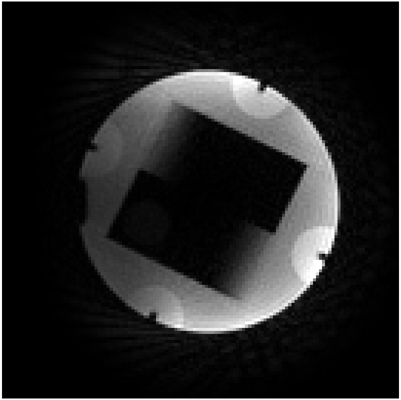
(b) IRGNTV

Radial sampling: phantom (25 proj)

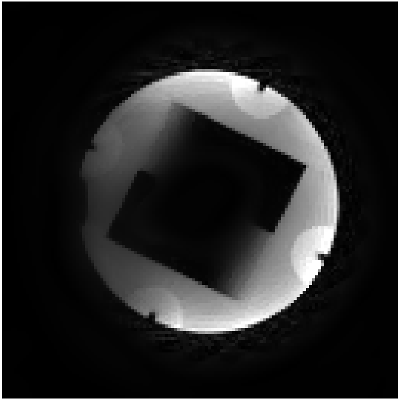


(a) IRGN (b) IRGNTV

Radial sampling: phantom (25 proj)

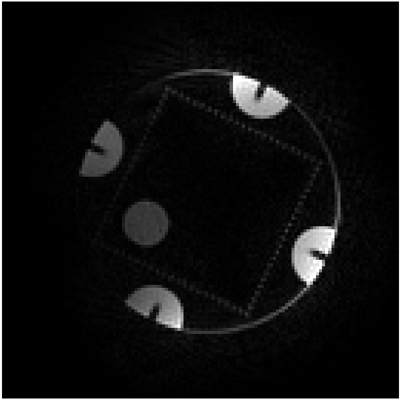


(a) IRGN

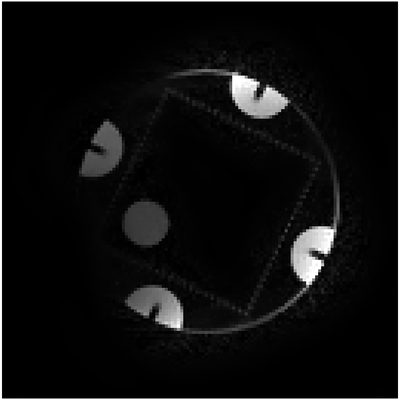


(b) IRGNTV

Radial sampling: phantom (25 proj)

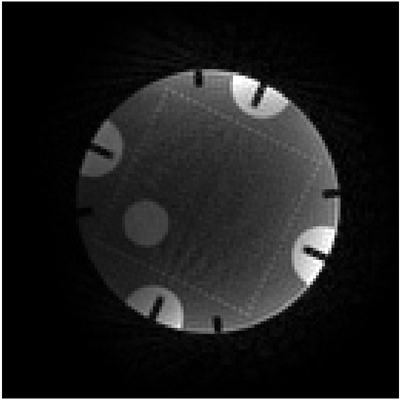


(a) IRGN

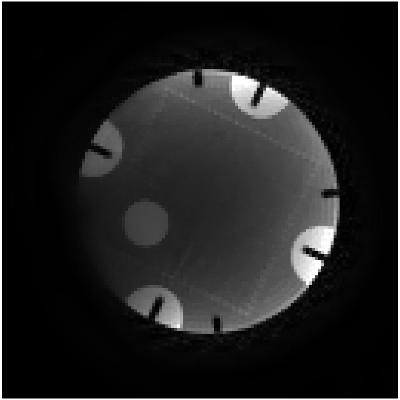


(b) IRGNTV

Radial sampling: phantom (25 proj)

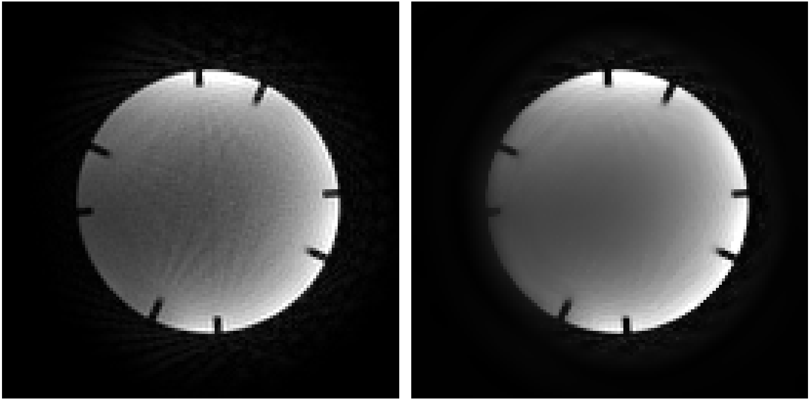


(a) IRGN



(b) IRGNTV

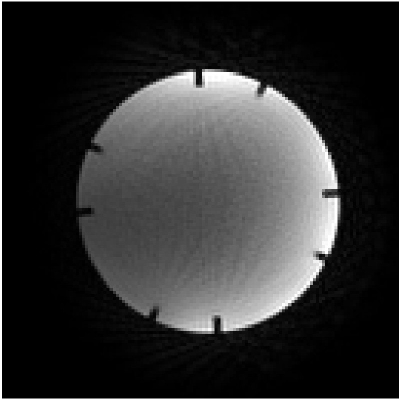
Radial sampling: phantom (25 proj)



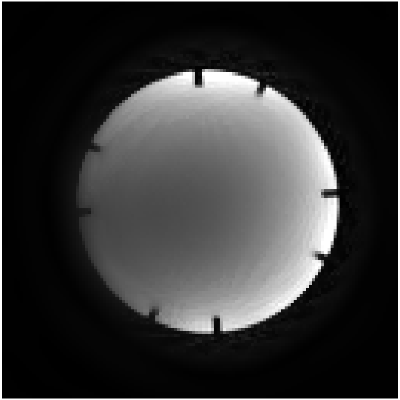
(a) IRGN

(b) IRGNTV

Radial sampling: phantom (25 proj)

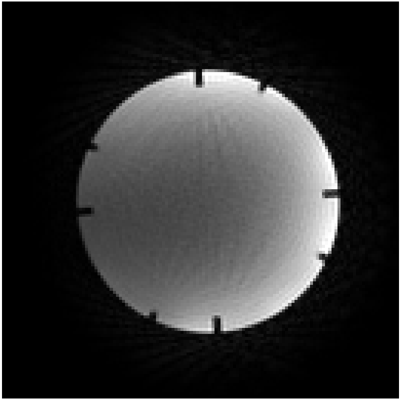


(a) IRGN

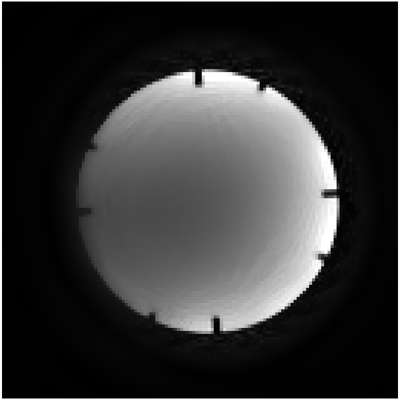


(b) IRGNTV

Radial sampling: phantom (25 proj)

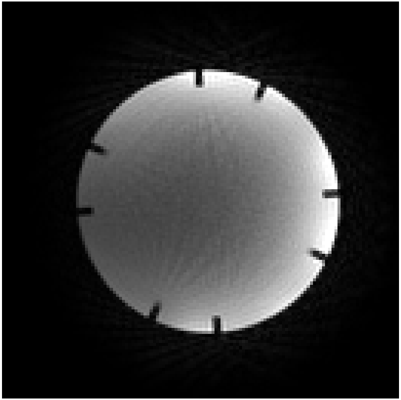


(a) IRGN

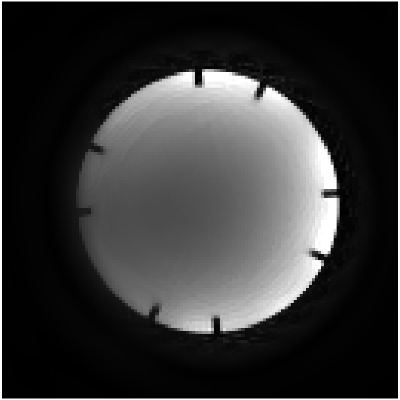


(b) IRGNTV

Radial sampling: phantom (25 proj)

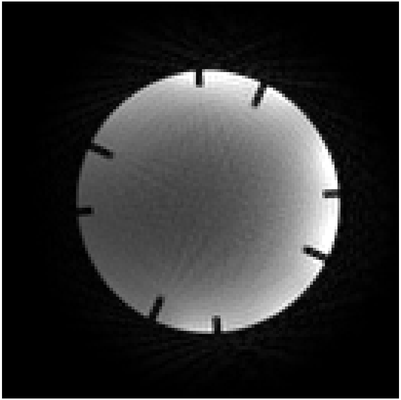


(a) IRGN

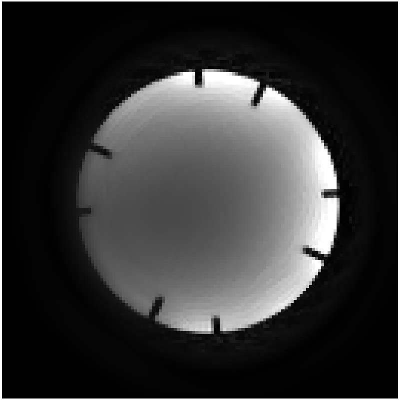


(b) IRGNTV

Radial sampling: phantom (25 proj)

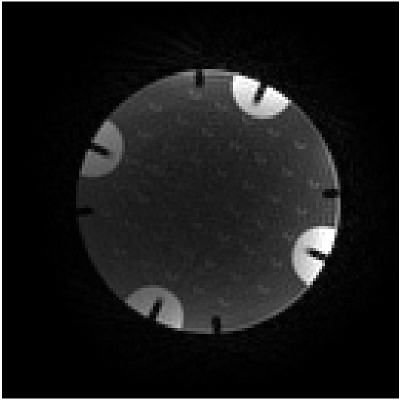


(a) IRGN

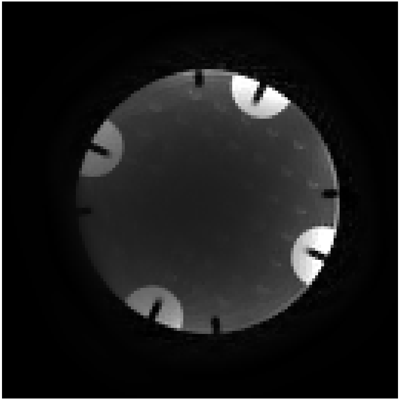


(b) IRGNTV

Radial sampling: phantom (25 proj)

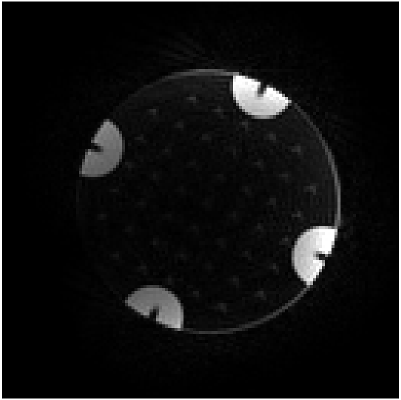


(a) IRGN

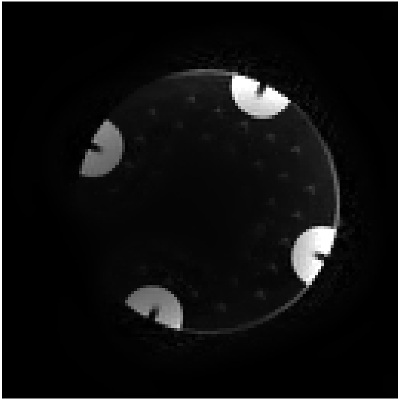


(b) IRGNTV

Radial sampling: phantom (25 proj)

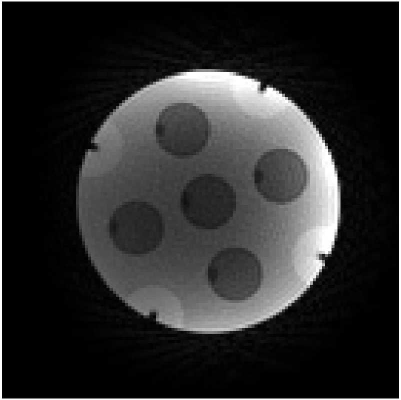


(a) IRGN

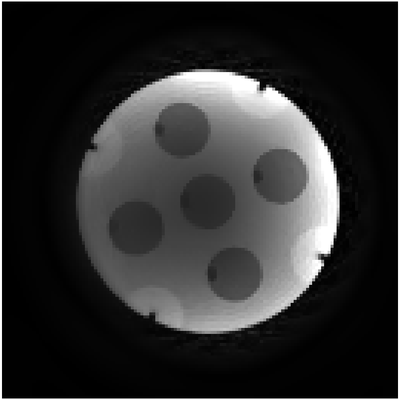


(b) IRGNTV

Radial sampling: phantom (25 proj)

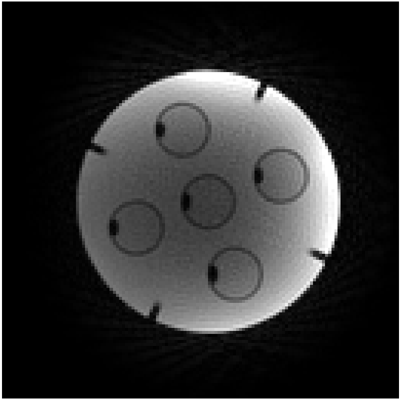


(a) IRGN

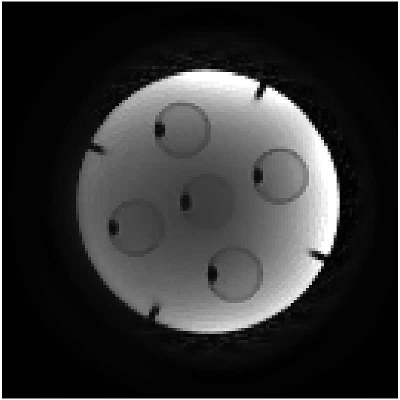


(b) IRGNTV

Radial sampling: phantom (25 proj)

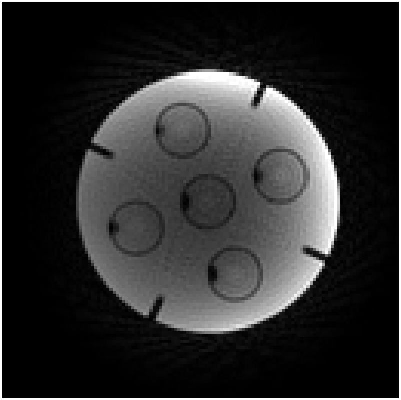


(a) IRGN

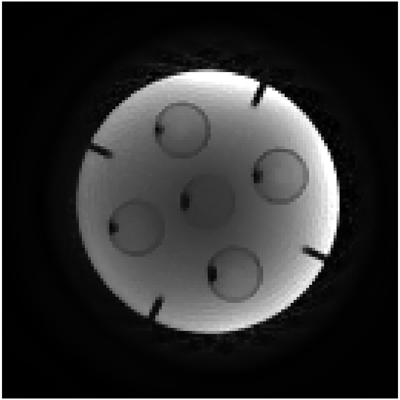


(b) IRGNTV

Radial sampling: phantom (25 proj)

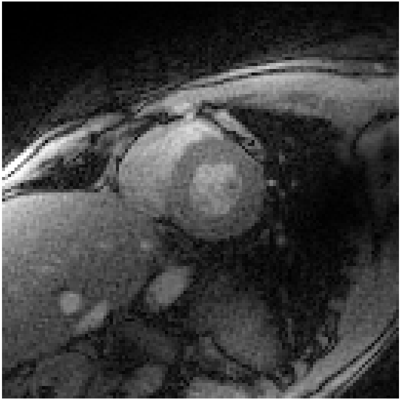


(a) IRGN

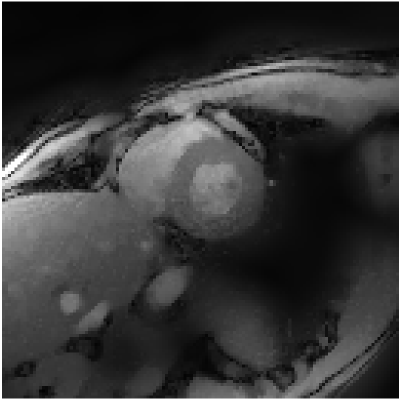


(b) IRGNTV

Radial sampling: cardiac (25 proj $\approx$ 20 fps)

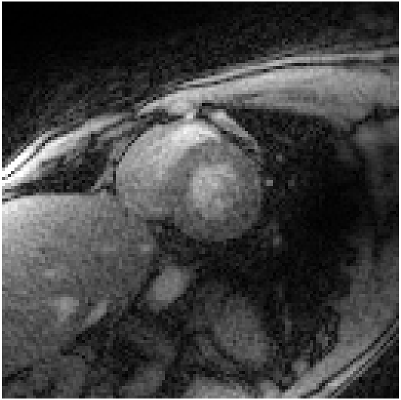


(a) IRGN

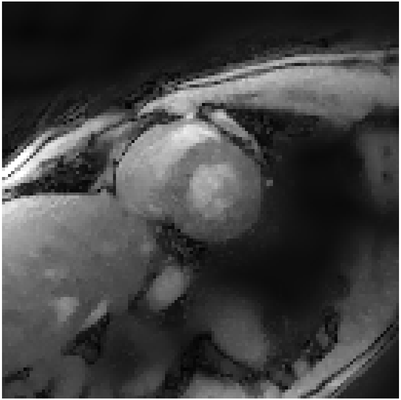


(b) IRGNTV

Radial sampling: cardiac (19 proj $\approx$ 26 fps)



(a) IRGN



(b) IRGNTV

Total generalized variation (TGV)

Large TV penalty leads to stair-casing $\rightsquigarrow$ include penalty on higher derivatives, promoting piecewise smooth reconstruction

Here: second order **total generalized variation**

$$\beta TGV^2(u) = \sup_{v \in C^2_\beta} \langle u, \operatorname{div}^2 v \rangle$$

with

$$C^2_\beta = \left\{ v \in \mathcal{C}^2_c(\Omega, \mathcal{S}^{d\times d}) : \ \|v\|_\infty \leq 2\beta, \|\operatorname{div} v\|_\infty \leq \beta \right\}$$

(see http://math.uni-graz.at/mobis/publications/SFB-Report-2010-023.pdf for details)

IRGNTGV

Replace *TV* penalty on u^{k+1} with *TGV*:

1: Choose $x^0 = (u^0, c^0)$, α_0, β_0, $q < 1$
2: **repeat**
3: Solve for $\delta x = (\delta u, \delta c)$

$$\min_{\delta x} \frac{1}{2}\|F'(x^k)\delta x + F(x^k) - g\|^2 + \frac{\alpha_k}{2}\|W(c^k + \delta c)\|^2 + \beta_k TGV^2(u^k + \delta u)$$

4: Set $x^{k+1} = x^k + \delta x$, $\alpha_{k+1} = \alpha_k q$, $\beta_{k+1} = \beta_k q$, $k = k + 1$
5: **until** $\|F(x^k) - g\| < tol$
6: **return** u, c

Solution of IRGNTGV subproblems

Convex duality:

$$\beta TGV^2(u) = \inf_v \beta\|\nabla u - v\| + 2\beta\|\mathcal{E} v\|$$

Here: $v \in \mathcal{C}^1(\Omega, \mathbb{C}^d)$, $\mathcal{E} v = \frac{1}{2}(\nabla v + \nabla v^T) = (-\text{div}^{\,2})^* v$

$\rightsquigarrow$ **Interpretation**: TGV balances first and second derivative

Saddle point problem

$$\min_{\delta u, \delta c, v} \max_{\substack{p \in C_{\beta_k} \\ q \in C_{2\beta_k}}} J(\delta u, \delta c) + \langle \nabla u^k + \delta u - v, p\rangle + \langle \mathcal{E} v, q\rangle$$

Primal-dual extragradient method

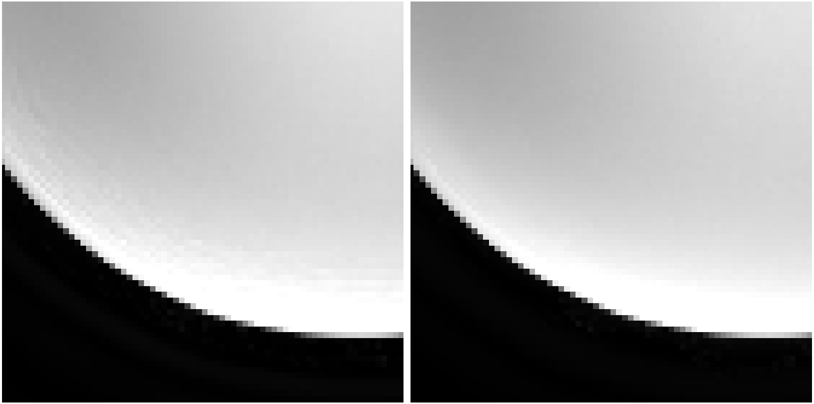
function TGVSOLVE$(u, c, \alpha, \beta, \sigma_u, \sigma_c, \sigma_v, \tau)$
    $\delta u, \overline{\delta u}, \delta c, \overline{\delta c}, v, \overline{v}, p, q \leftarrow 0$
    repeat
        $p \leftarrow \text{proj}_\beta(p + \tau(\nabla(u + \overline{\delta u}) - v)$
        $q \leftarrow \text{proj}_{2\beta}(q + \tau(\mathcal{E} v))$
        $\delta u_{old} \leftarrow \delta u,\ \delta c_{old} \leftarrow \delta c,\ v_{old} \leftarrow v$
        $\delta u \leftarrow \delta u - \sigma_u(\partial_u J(u, c)(\overline{\delta u}, \overline{\delta c}) - \text{div}\, p)$
        $\delta c \leftarrow \delta c - \sigma_c(\partial_c J(u, c)(\overline{\delta u}, \overline{\delta c}))$
        $v \leftarrow v - \sigma_v(-p - \text{div}^2 q)$
        $\overline{\delta u} \leftarrow 2\delta u - \delta u_{old}$
        $\overline{\delta c} \leftarrow 2\delta c - \delta c_{old}$
        $\overline{v} \leftarrow 2v - v_{old}$
    until convergence
end function

Effect of TGV: Random ($R = 4$)

(a) IRGNTV

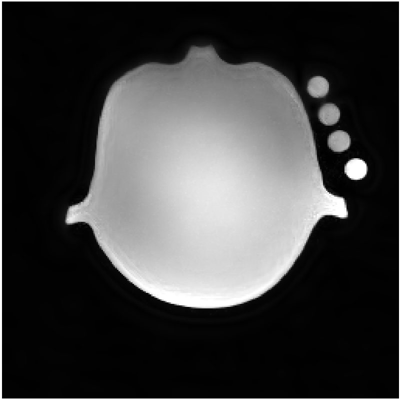
(b) IRGNTGV

Effect of TGV: Random ($R = 4$)

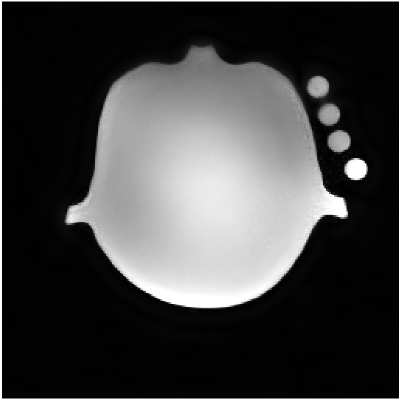


(a) IRGNTV (detail) (b) IRGNTGV (detail)

Effect of TGV: Random ($R = 10$)

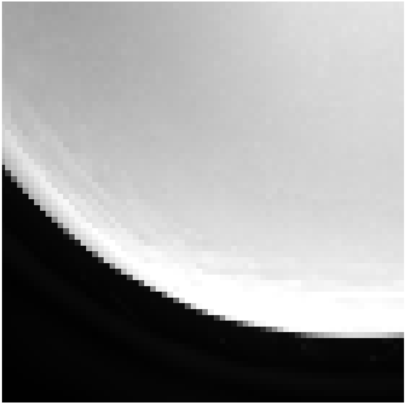


(a) IRGNTV

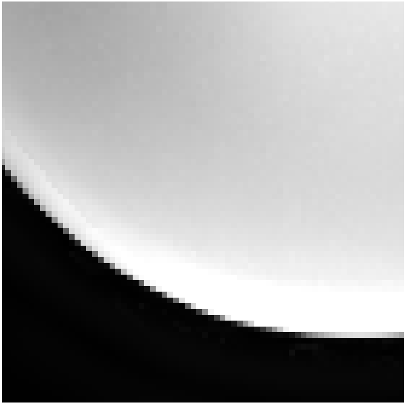


(b) IRGNTGV

Effect of TGV: Random ($R = 10$)

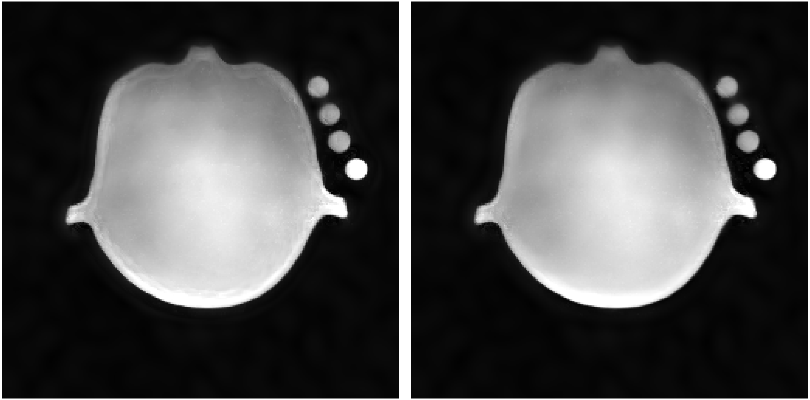


(a) IRGNTV (detail)



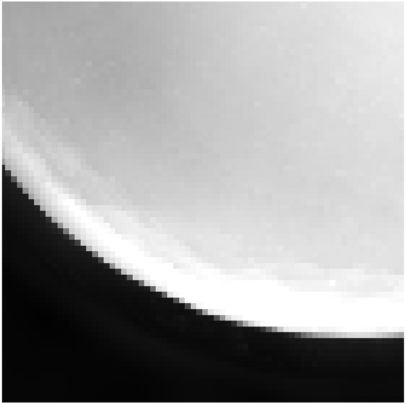
(b) IRGNTGV (detail)

Effect of TGV: Random ($R = 18$)

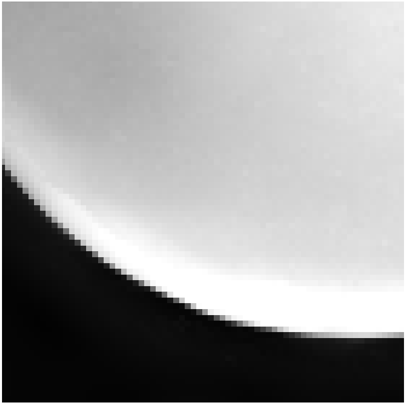


(a) IRGNTV (b) IRGNTGV

Effect of TGV: Random ($R = 18$)

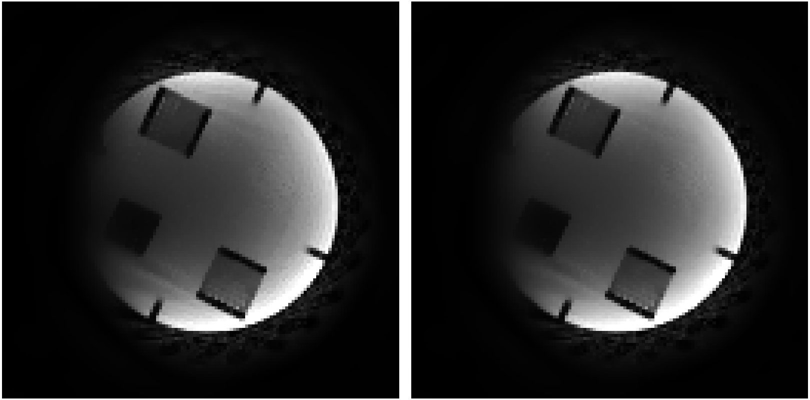


(a) IRGNTV (detail)



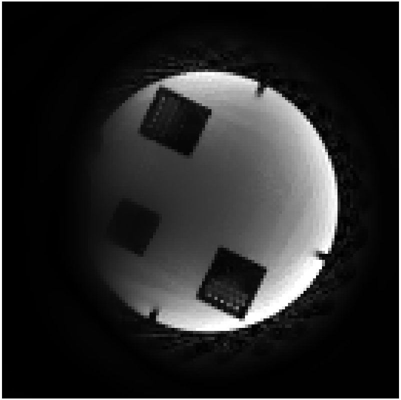
(b) IRGNTGV (detail)

Radial sampling: phantom (25 proj)

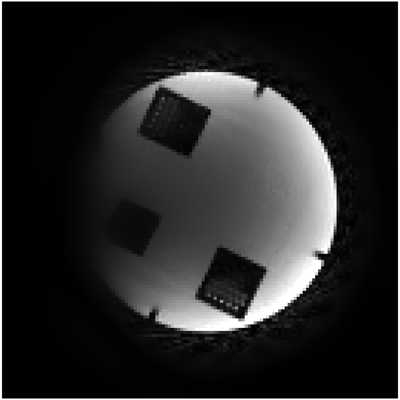


(a) IRGNTV (b) IRGNTGV

Radial sampling: phantom (25 proj)

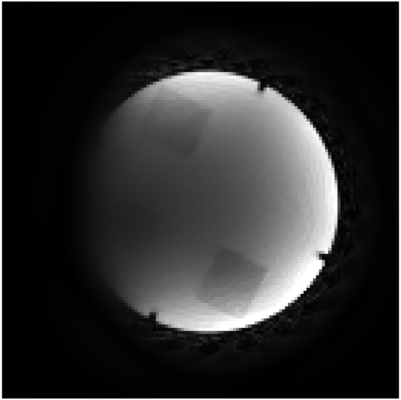


(a) IRGNTV

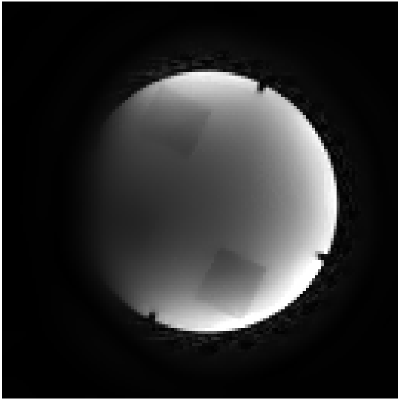


(b) IRGNTGV

Radial sampling: phantom (25 proj)

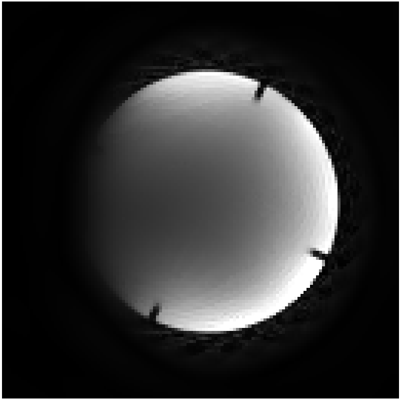


(a) IRGNTV

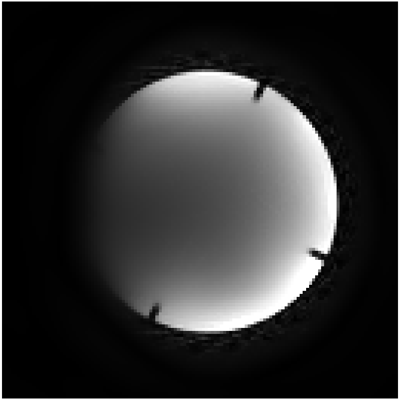


(b) IRGNTGV

Radial sampling: phantom (25 proj)

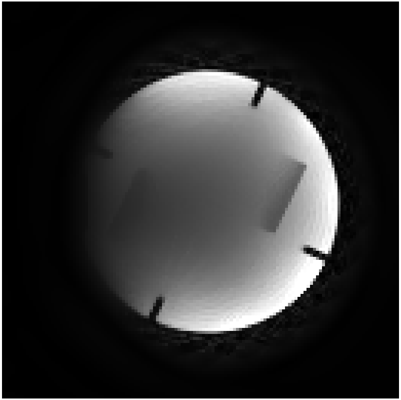


(a) IRGNTV

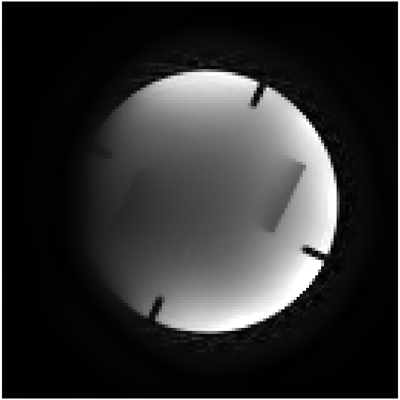


(b) IRGNTGV

Radial sampling: phantom (25 proj)

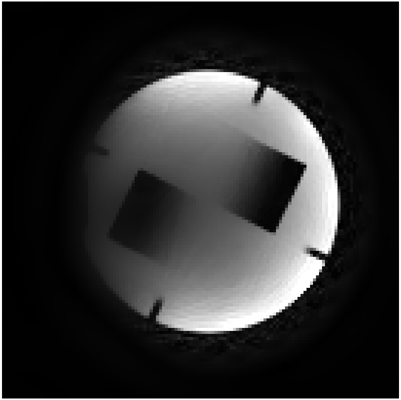


(a) IRGNTV

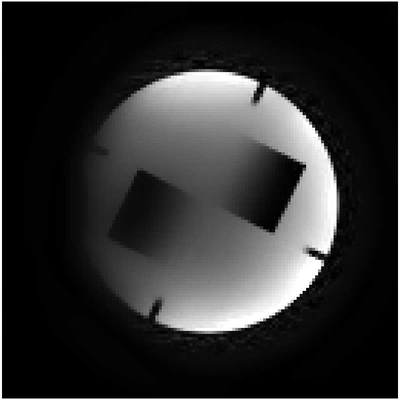


(b) IRGNTGV

Radial sampling: phantom (25 proj)

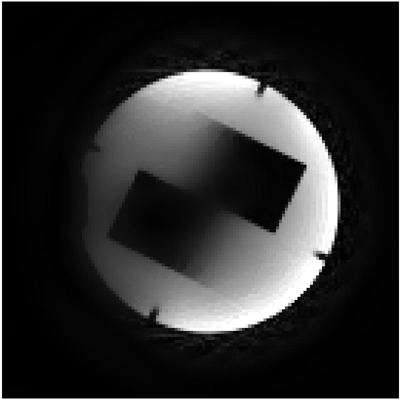


(a) IRGNTV

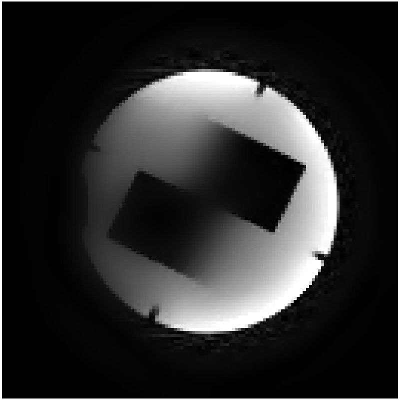


(b) IRGNTGV

Radial sampling: phantom (25 proj)

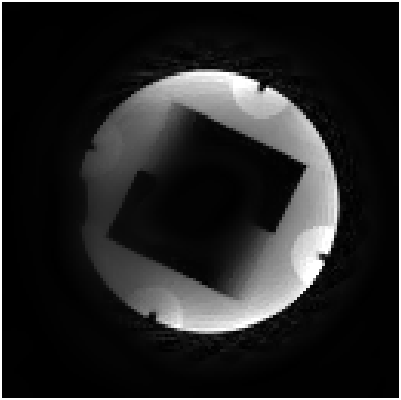


(a) IRGNTV

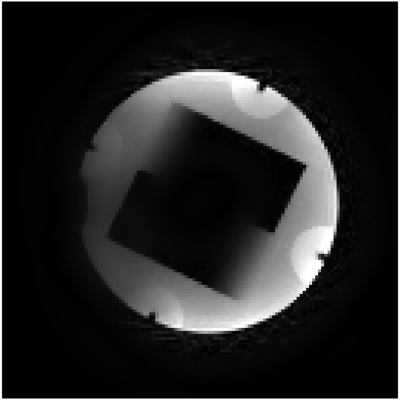


(b) IRGNTGV

Radial sampling: phantom (25 proj)

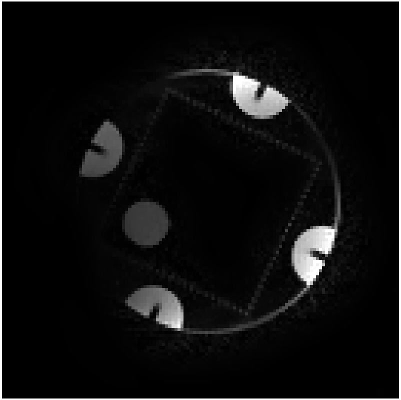


(a) IRGNTV

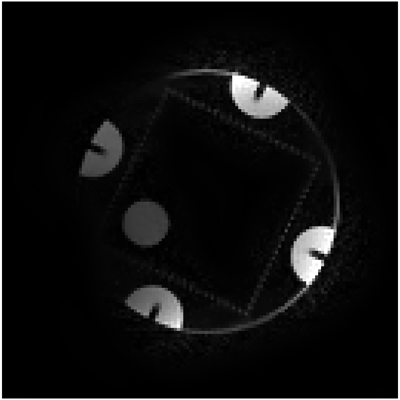


(b) IRGNTGV

Radial sampling: phantom (25 proj)

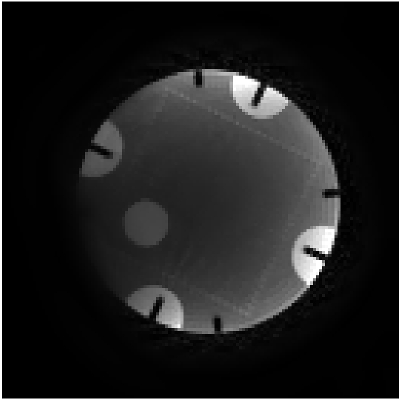


(a) IRGNTV

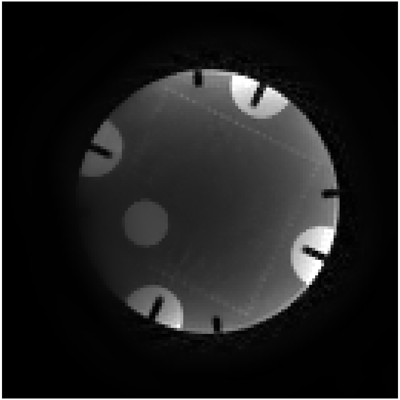


(b) IRGNTGV

Radial sampling: phantom (25 proj)

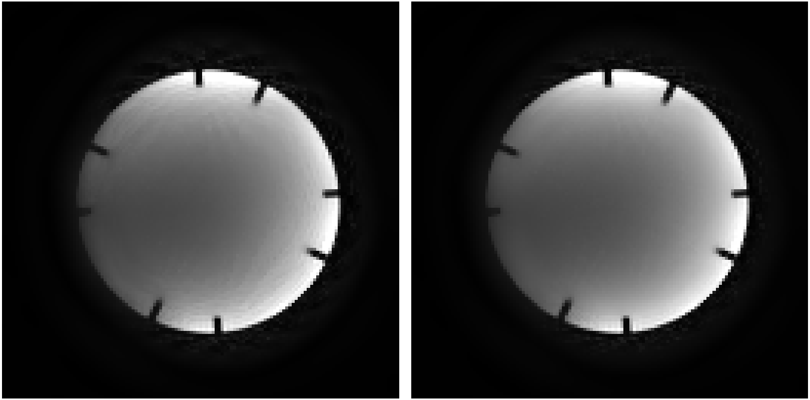


(a) IRGNTV



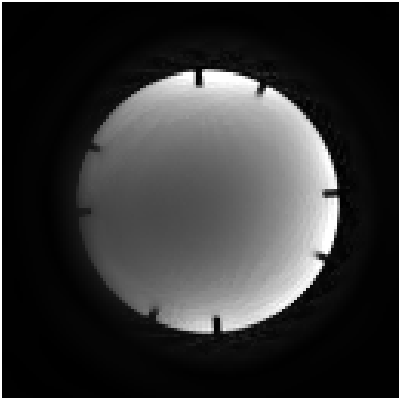
(b) IRGNTGV

Radial sampling: phantom (25 proj)

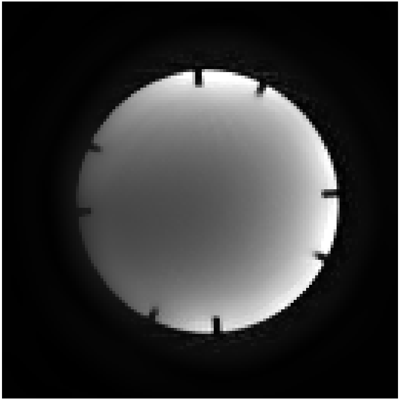


(a) IRGNTV (b) IRGNTGV

Radial sampling: phantom (25 proj)

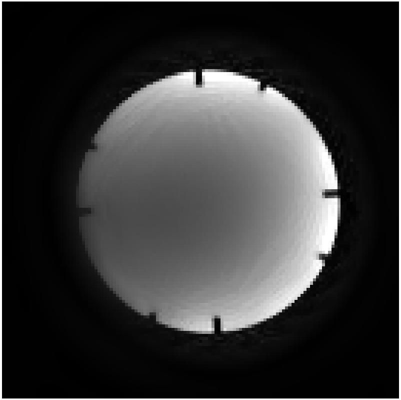


(a) IRGNTV

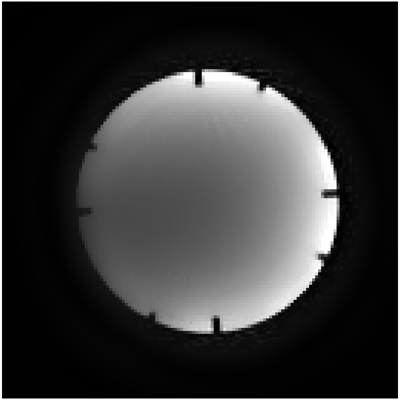


(b) IRGNTGV

Radial sampling: phantom (25 proj)

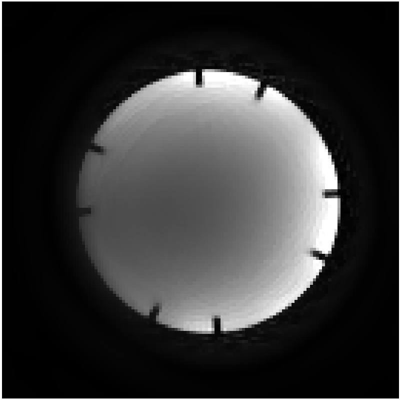


(a) IRGNTV

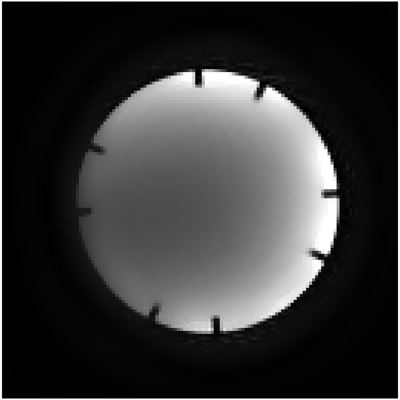


(b) IRGNTGV

Radial sampling: phantom (25 proj)

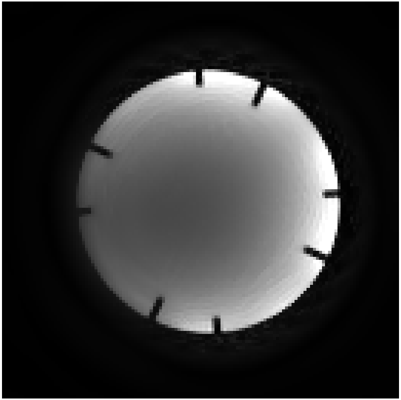


(a) IRGNTV

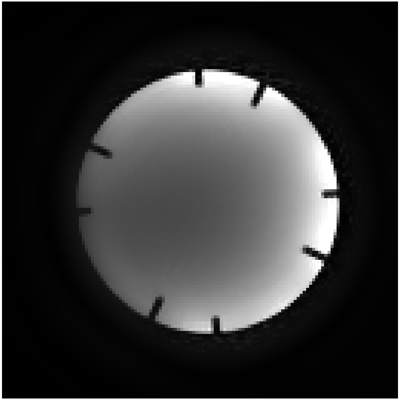


(b) IRGNTGV

Radial sampling: phantom (25 proj)

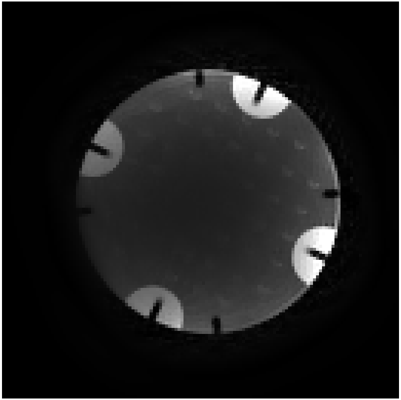


(a) IRGNTV

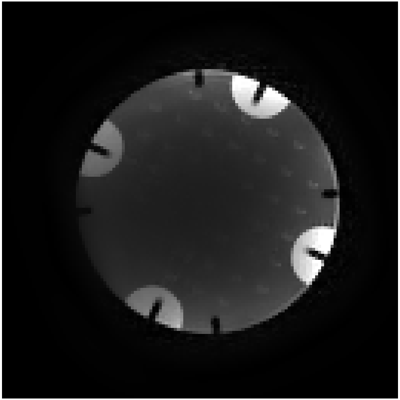


(b) IRGNTGV

Radial sampling: phantom (25 proj)

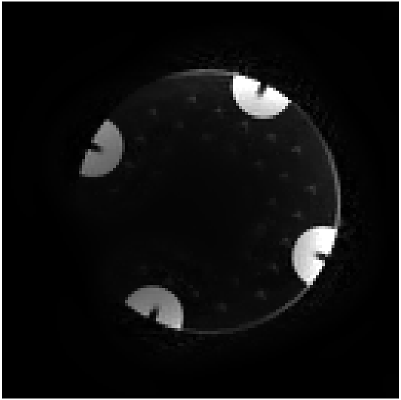


(a) IRGNTV

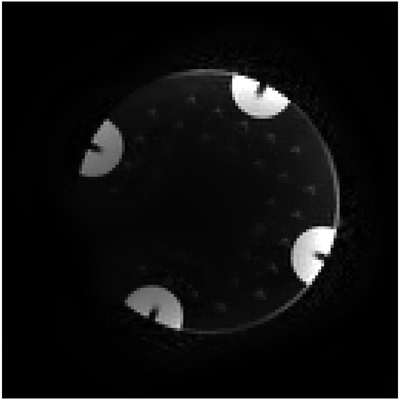


(b) IRGNTGV

Radial sampling: phantom (25 proj)

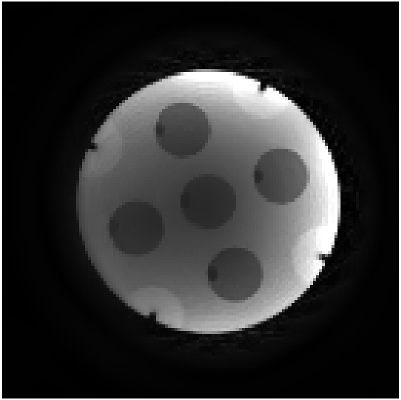


(a) IRGNTV

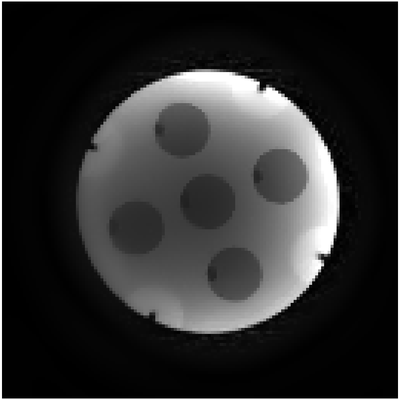


(b) IRGNTGV

Radial sampling: phantom (25 proj)

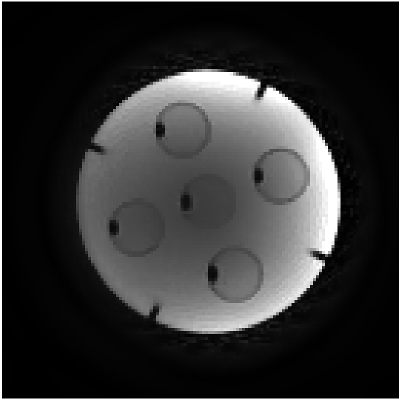


(a) IRGNTV

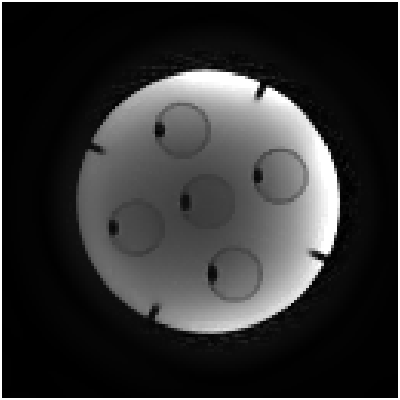


(b) IRGNTGV

Radial sampling: phantom (25 proj)

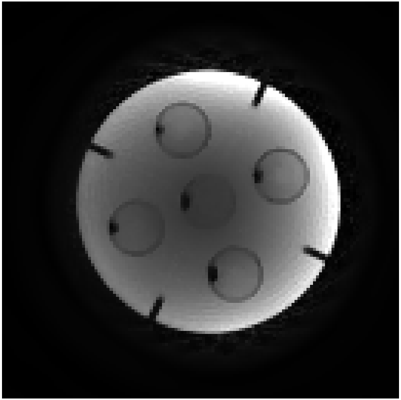


(a) IRGNTV

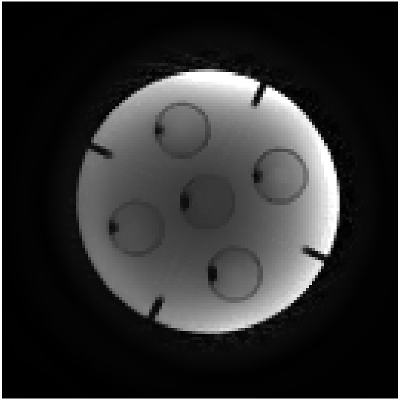


(b) IRGNTGV

Radial sampling: phantom (25 proj)



(a) IRGNTV



(b) IRGNTGV

Conclusion

Summary:

- Nonlinear inverse approach gives flexibility
- IRGNTV more stable, same complexity as IRGN
- IRGNTGV better for modulated images

Outlook:

- Add constraint on slice/frame differences; 3DT(G)V
- Include parameter identification in IRGN

Thanks to Martin Uecker (FLASH data), Kristian Bredies (TGV)

Computation of gradients

$$J(\delta u, \delta c) = \frac{1}{2}\|F'(x)\delta x + F(x) - g\|^2 + \frac{\alpha}{2}\|W(c+\delta c)\|^2$$

$$\partial_u J(u,c)(\delta u, \delta c) = \sum_{i=1}^{N} c_i^* \cdot \mathcal{F}_s^*(\mathcal{F}_s(u \cdot \delta c_i + c_i \cdot \delta u) + F(u,c) - g)$$

$$(\partial_c J(u,c)(\delta u, \delta c))_i = u^* \cdot \mathcal{F}_s^*(\mathcal{F}_s(u \cdot \delta c_i + c_i \cdot \delta u) + F(u,c) - g)) + \alpha W^* W(c_i + \delta c_i)$$

$\rightsquigarrow$ only (N)FFT, pointwise multiplication required

back