Stack Transition Artifact Removal for Cardiac CT using a Symmetric Demons Algorithm

Sergej Lebedev[1,2,3], Eric Fournie[1], Karl Stierstorfer[1] and Marc Kachelrieß[2,3]

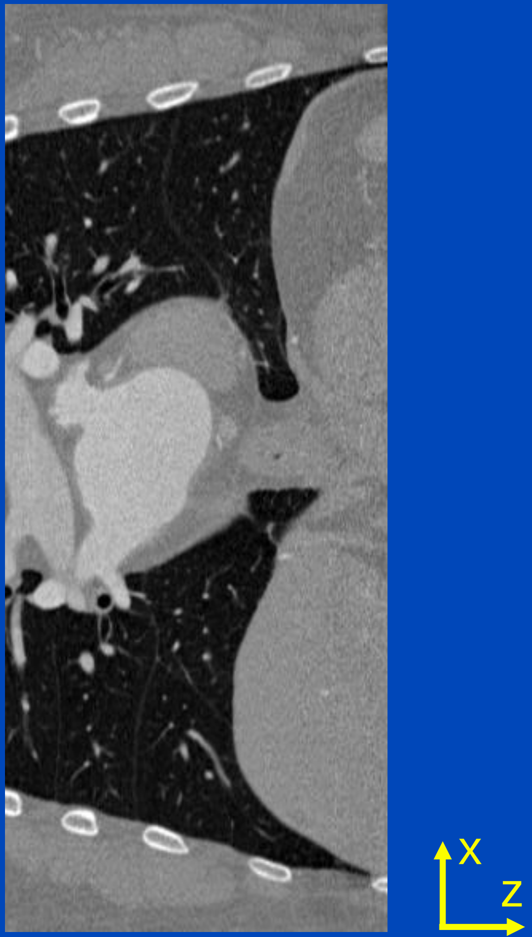
[1]Siemens Healthineers, Forchheim, Germany
[2]German Cancer Research Center (DKFZ), Heidelberg, Germany
[3]University of Heidelberg, Germany

SIEMENS

dkfz. DEUTSCHES KREBSFORSCHUNGSZENTRUM IN DER HELMHOLTZ-GEMEINSCHAFT

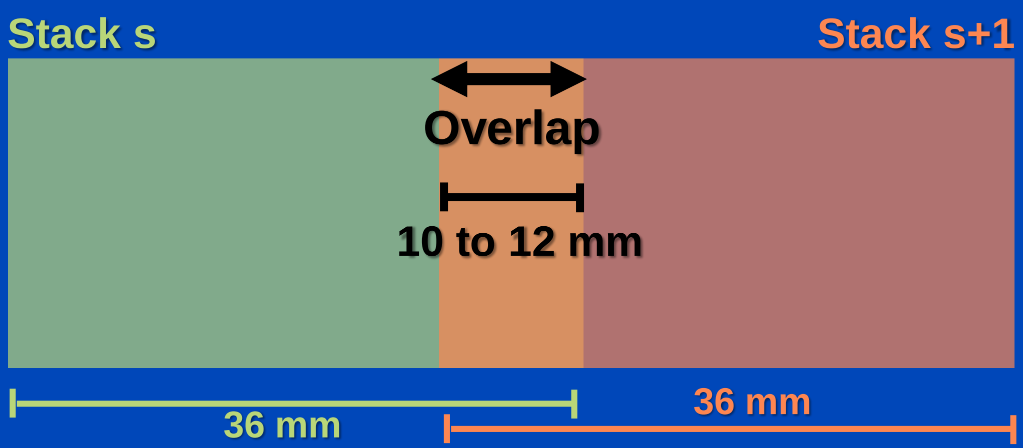
Introduction
Cardiac Imaging

- Data of from one cardiac phase can be acquired via prospective ECG-gating or extracted from a retrospectively gated data set.
- Cardiac reconstructions can yield sub volumes (stacks) corresponding to different times and, ideally, to the same heart phase.



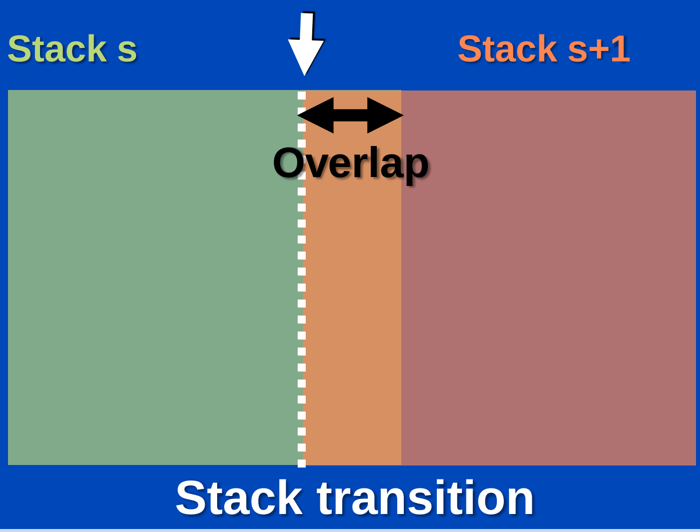
Introduction
Cardiac Imaging

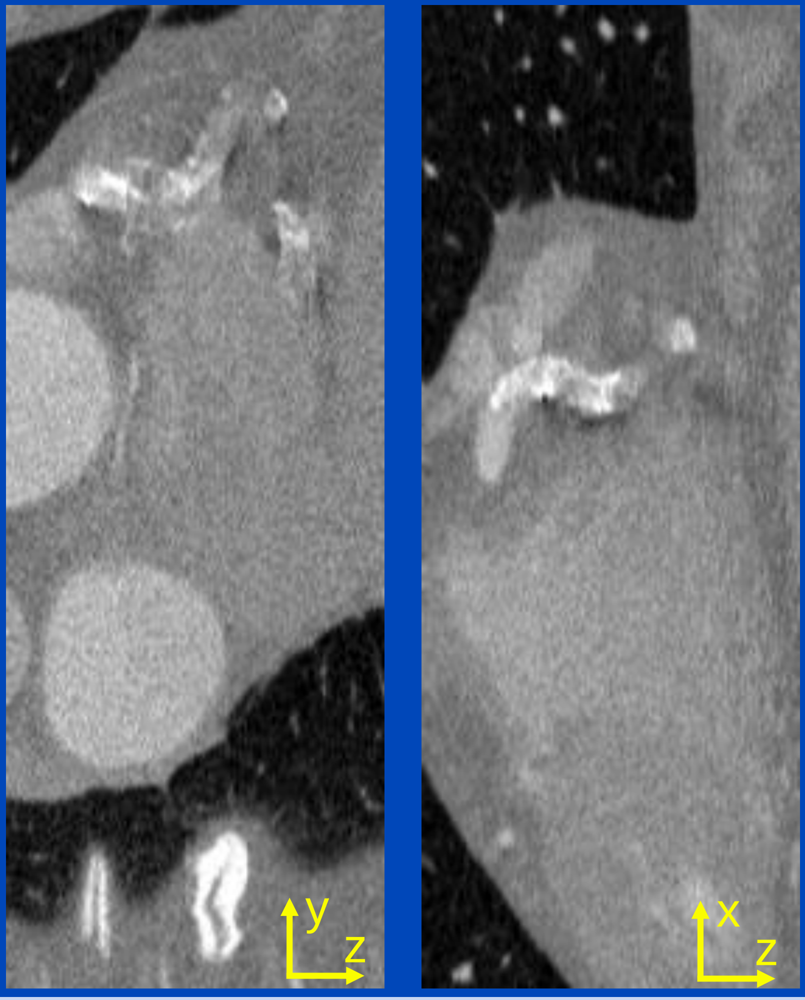
- Data of from one cardiac phase can be acquired via prospective ECG-gating or extracted from a retrospectively gated data set.
- Cardiac reconstructions can yield sub volumes (stacks) corresponding to different times and, ideally, to the same heart phase.
- The depth of the stacks depends on the longitudinal collimation of the CT scanner.
- The stacks generally have a longitudinal overlap.



Introduction
Stacks

- The final CT volume is assembled from the stacks.
- The stack transition, from which the next stack is used, can theoretically be set to any position within the stack overlap.
- A blending between the stacks can also be performed.

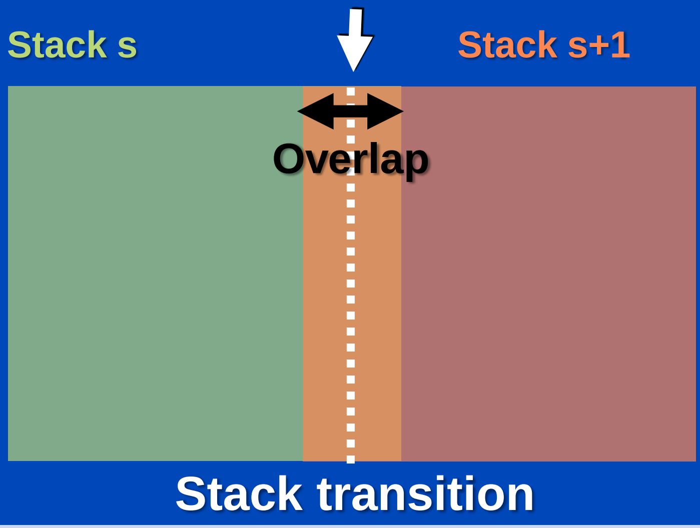


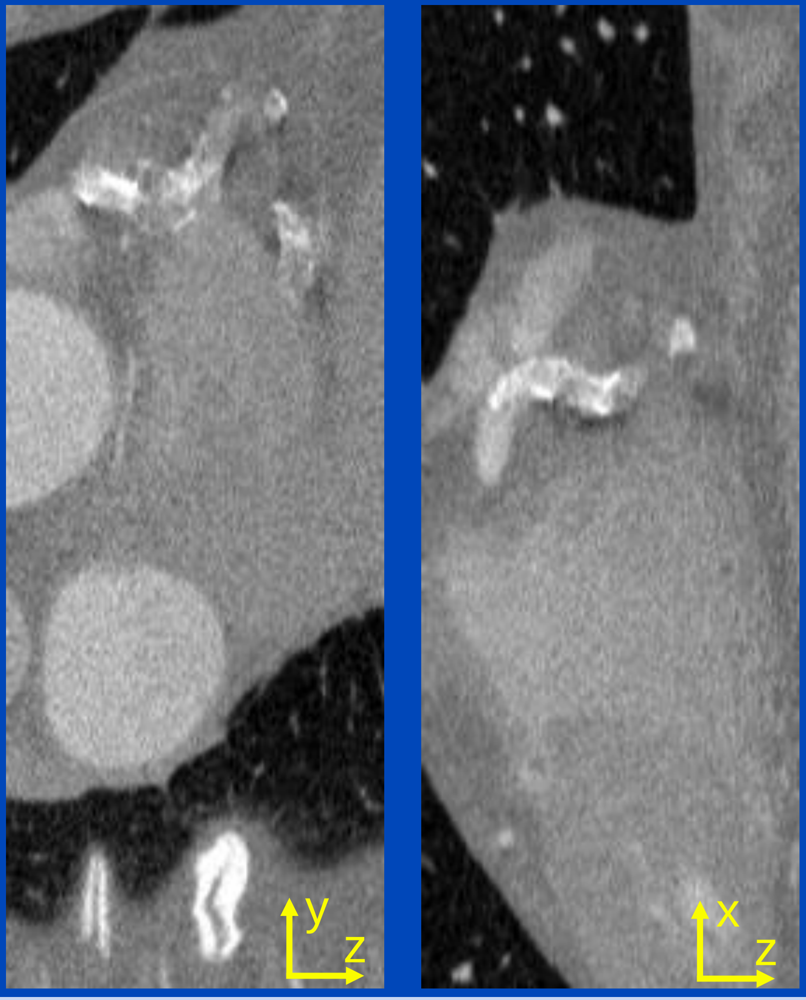


Introduction

Stacks

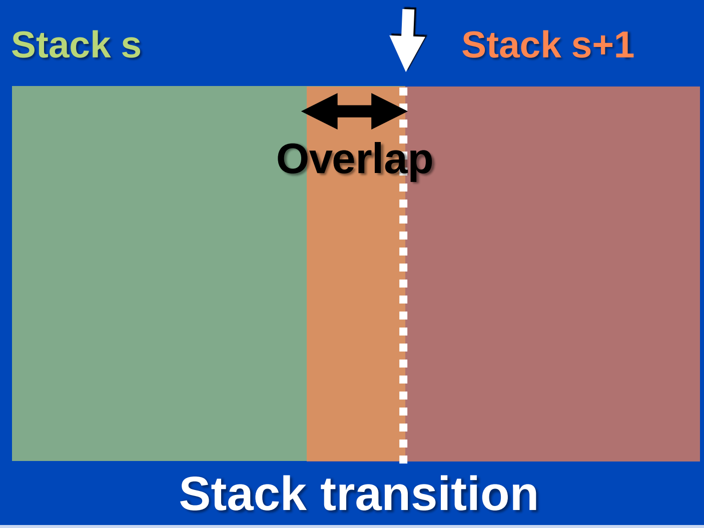
- The final CT volume is assembled from the stacks.
- The stack transition, from which the next stack is used, can theoretically be set to any position within the stack overlap.
- A blending between the stacks can also be performed.





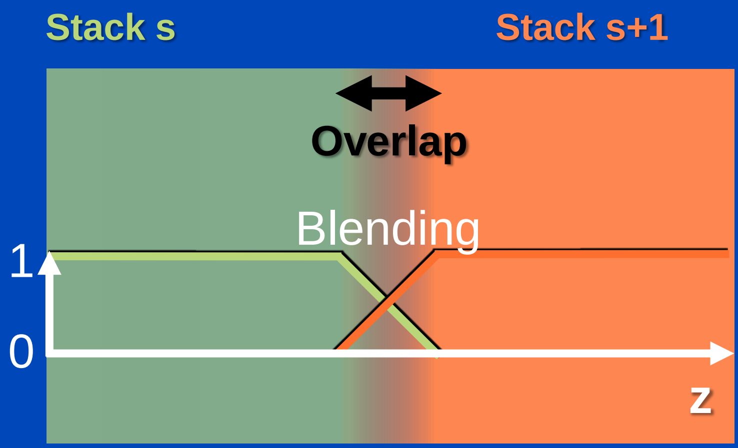
Introduction
Stacks

- The final CT volume is assembled from the stacks.
- The stack transition, from which the next stack is used, can theoretically be set to any position within the stack overlap.
- A blending between the stacks can also be performed.



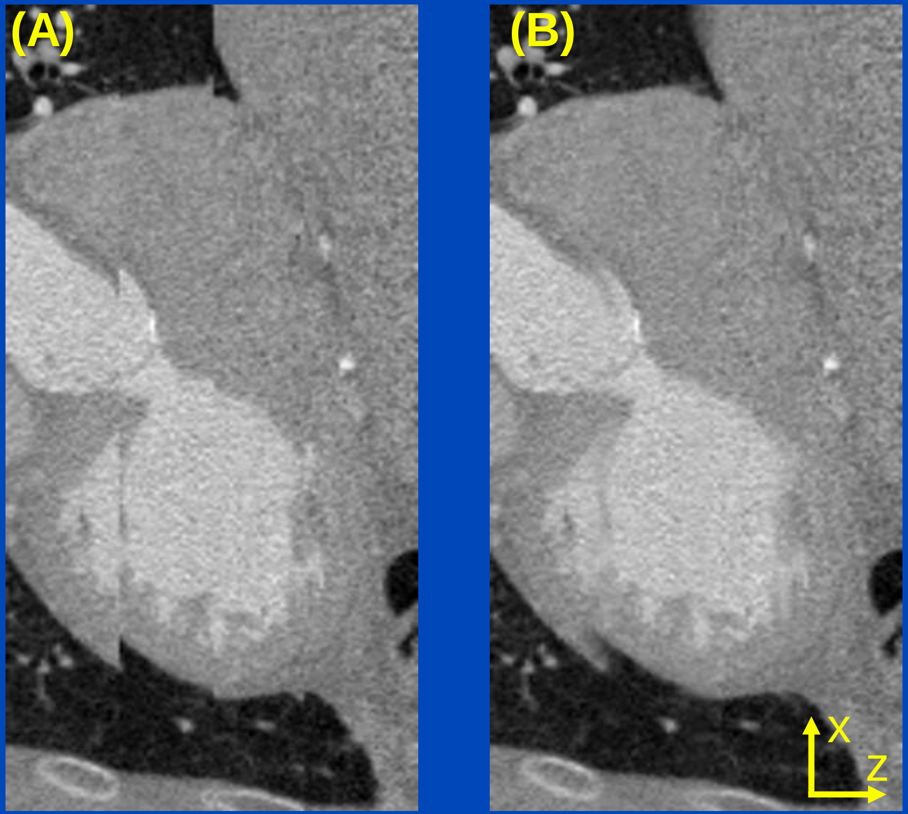
Introduction
Stacks

- The final CT volume is assembled from the stacks.
- The stack transition, from which the next stack is used, can theoretically be set to any position within the stack overlap.
- A blending between the stacks can also be performed.



Introduction
Stack transition artifacts

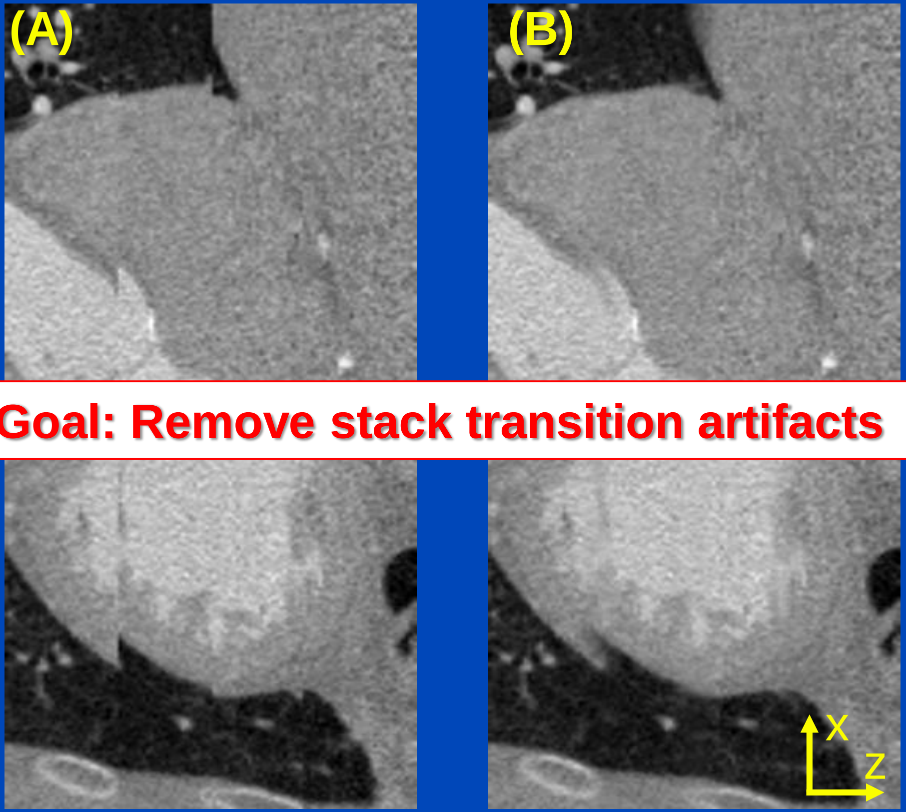
- **Irregular motion leads to stacks that do not represent exactly the same volume.**
- **Discontinuities (misalignment) at stack transitions arise when stitching the stacks together to yield the complete CT volume.**



Two coronal slices from a cardiac data set with strong stack transition artifacts. (A) Sharp stack transition. (B) Blending between stacks.

Introduction
Stack transition artifacts

- **Irregular motion leads to stacks that do not represent exactly the same volume.**
- **Discontinuities (misalignment) at stack transitions arise when stitching the stacks together to yield the complete CT volume.**



Goal: Remove stack transition artifacts

Two coronal slices from a cardiac data set with strong stack transition artifacts. (A) Sharp stack transition. (B) Blending between stacks.

Methods
Symmetric registration

- **Many registration approaches assume one volume that is registered onto a target volume.**
- **Given two volumes $f_1(\boldsymbol{r})$, $f_2(\boldsymbol{r})$, compute a DVF $\boldsymbol{d}$(r) that will match the two.**
- **Herein, symmetric means that a method is symmetric in terms of the deformations that are applied to both volumes so that the transformed volumes $\hat{f}_1$(r) and $\hat{f}_2$(r) match:**

$$\hat{f}_1(\boldsymbol{r}) = f_1(\boldsymbol{r} + \boldsymbol{d}(\boldsymbol{r}))$$
$$\hat{f}_2(\boldsymbol{r}) = f_2(\boldsymbol{r} - \boldsymbol{d}(\boldsymbol{r})).$$

DVF applied in opposing directions

- **Other symmetry definitions have been used, e.g. symmetry w.r.t. image input order.**

Methods
Symmetric Demons algorithm

- **Optimize cost function to find DVF d:**

$$C(\boldsymbol{e},\boldsymbol{d}) = ||\frac{1}{\sigma_i(\boldsymbol{r})}(T_{+e}\,\boldsymbol{f} - T_{-e}\,\boldsymbol{g})(\boldsymbol{r})||_2^2 + \frac{1}{\sigma_x^2}||\boldsymbol{e}-\boldsymbol{d}||_2^2 + \frac{1}{\sigma_T^2}||\nabla \boldsymbol{d}||_2^2$$

Methods

Symmetric Demons algorithm

- **Optimize cost function to find DVF d:**

$$C(\boldsymbol{e},\boldsymbol{d}) = \underbrace{||\frac{1}{\sigma_i(\boldsymbol{r})}(T_{+e}\,\boldsymbol{f} - T_{-e}\,\boldsymbol{g})(\boldsymbol{r})||_2^2 + \frac{1}{\sigma_x^2}||\overbrace{\boldsymbol{e}-\boldsymbol{d}}^{\Delta d}||_2^2} + \frac{1}{\sigma_T^2}||\nabla\boldsymbol{d}||_2^2$$

<u>1st optimization step</u>

- Ensure similarity
- Intermediate DVF: $\boldsymbol{e} = \boldsymbol{d} + \Delta\boldsymbol{d}$
- Find optimal $\Delta\boldsymbol{d}$

Methods
Symmetric Demons algorithm

- **Optimize cost function to find DVF d:**

$$C(\boldsymbol{e},\boldsymbol{d}) = ||\frac{1}{\sigma_i(\boldsymbol{r})}(T_{+e}\,\boldsymbol{f} - T_{-e}\,\boldsymbol{g})(\boldsymbol{r})||_2^2 + \frac{1}{\sigma_x^2}||\overbrace{\boldsymbol{e}-\boldsymbol{d}}^{\Delta d}||_2^2 + \frac{1}{\sigma_T^2}||\nabla \boldsymbol{d}||_2^2$$

1st optimization step
- Ensure similarity
- Intermediate DVF: $\boldsymbol{e} = \boldsymbol{d} + \Delta\boldsymbol{d}$
- Find optimal $\Delta\boldsymbol{d}$

Analytical solutions can be found for both steps

→ No computationally expensive iterative searches within the main iteration

2nd optimization step
- Ensure smoothness
- Find new, optimal $\boldsymbol{d}$

Methods

Symmetric Demons algorithm

- **Optimize cost function to find DVF d:**

$$C(\boldsymbol{e},\boldsymbol{d}) = \left\|\frac{1}{\sigma_i(\boldsymbol{r})}(T_{+\boldsymbol{e}}\boldsymbol{f} - T_{-\boldsymbol{e}}\boldsymbol{g})(\boldsymbol{r})\right\|_2^2 + \frac{1}{\sigma_x^2}\|\underbrace{\boldsymbol{e}-\boldsymbol{d}}_{\Delta\boldsymbol{d}}\|_2^2 + \frac{1}{\sigma_T^2}\|\nabla\boldsymbol{d}\|_2^2$$

1st optimization step

- Ensure similarity
- Intermediate DVF: $\boldsymbol{e} = \boldsymbol{d} + \Delta\boldsymbol{d}$
- Find optimal $\Delta\boldsymbol{d}$

Analytical solutions can be found for both steps

→ No computationally expensive iterative searches within the main iteration

→ Individual updates at each voxel position r_n

2nd optimization step

- Ensure smoothness
- Find new, optimal $\boldsymbol{d}$

1st update

- Compute additive update vectors Δd to acquire the intermediate DVF e

$$\Delta\boldsymbol{d} = \frac{(T_{+\boldsymbol{d}}\boldsymbol{f}_1 - T_{-\boldsymbol{d}}\boldsymbol{f}_2)(\boldsymbol{r}_n)(T_{+\boldsymbol{d}}\nabla\boldsymbol{f}_1 + T_{-\boldsymbol{d}}\nabla\boldsymbol{f}_2)(\boldsymbol{r}_n)}{(\sigma_i(\boldsymbol{r}_n)/\sigma_x)^2 + (T_{+\boldsymbol{d}}\nabla\boldsymbol{f}_1 - T_{-\boldsymbol{d}}\nabla\boldsymbol{f}_2)^2(\boldsymbol{r}_n)}$$

2nd update (convolution)

- Gaussian kernel used

$$\boldsymbol{d}(\boldsymbol{r}) = \boldsymbol{e}(\boldsymbol{r}) * \frac{(\sigma_x/\sigma_T)^{-1}}{\sqrt{2\pi}} \exp\left(-\boldsymbol{r}^2(\sigma_x/\sigma_T)^{-2}/2\right)$$

Methods

Parameterization

$$C(\boldsymbol{e},\boldsymbol{d}) = ||\frac{1}{\sigma_i(\boldsymbol{r})}(T_{+e}\,\boldsymbol{f} - T_{-e}\,\boldsymbol{g})(\boldsymbol{r})||_2^2 + \frac{1}{\sigma_x^2}||\boldsymbol{e}-\boldsymbol{d}||_2^2 + \frac{1}{\sigma_T^2}||\nabla\boldsymbol{d}||_2^2$$

- **σ_i accounts for image noise, set by local noise estimator.**
- **σ_x limits the update length.**
- **σ_T and σ_x define the standard deviation (SD)/convolution kernel width.**

Redefine the independent parameters into task-specific, more intuitive input parameters

- **Max. allowed update length:** $\Delta = \sigma_x/2$ **with** $||\Delta\boldsymbol{d}_n|| \leq \frac{\sigma_x}{2}$**. Limits the update in the 1st update step.**
- **SD for Gaussian kernel:** $\sigma = 2\Delta/\sigma_T$**. Affects the smoothness of the final DVF.**

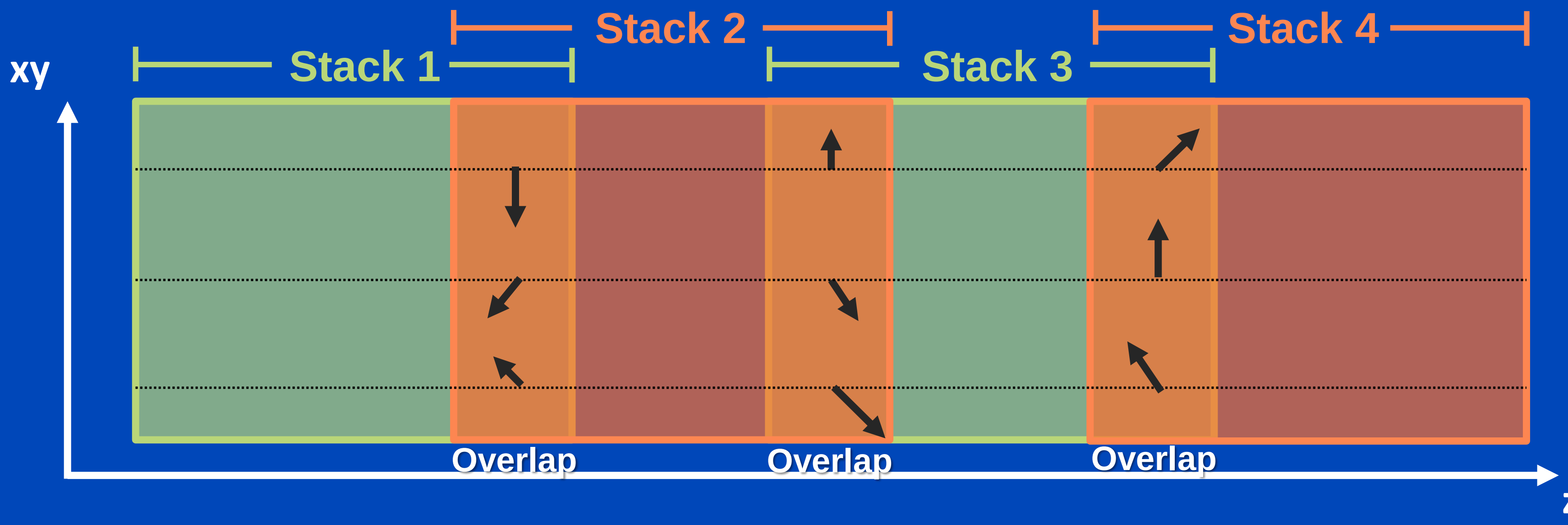
- **Parameter configuration:** $\sigma = 1\,\mathrm{mm}$, $\Delta = 2\,\mathrm{mm}$

Methods
Multi resolution approach

- Our symmetric Demons algorithm is applied at different resolutions to improve performance and stability.
- Intermediate results used to initialize registration with next best resolution.
- Used resolutions: Start at ~$3 \times 3 \times 3$ mm^3. Improve by a factor of two every step until achieving original resolution (Here: $0.3 \times 0.3 \times 0.6$ mm^3).

Methods
DVF interpolation

- Given S stacks, S-1 registrations are performed yielding as many DVFs defined in the respective overlap
- DVFs are extended on to the non-overlapping/non-redundant regions via interpolation to achieve smooth transformations

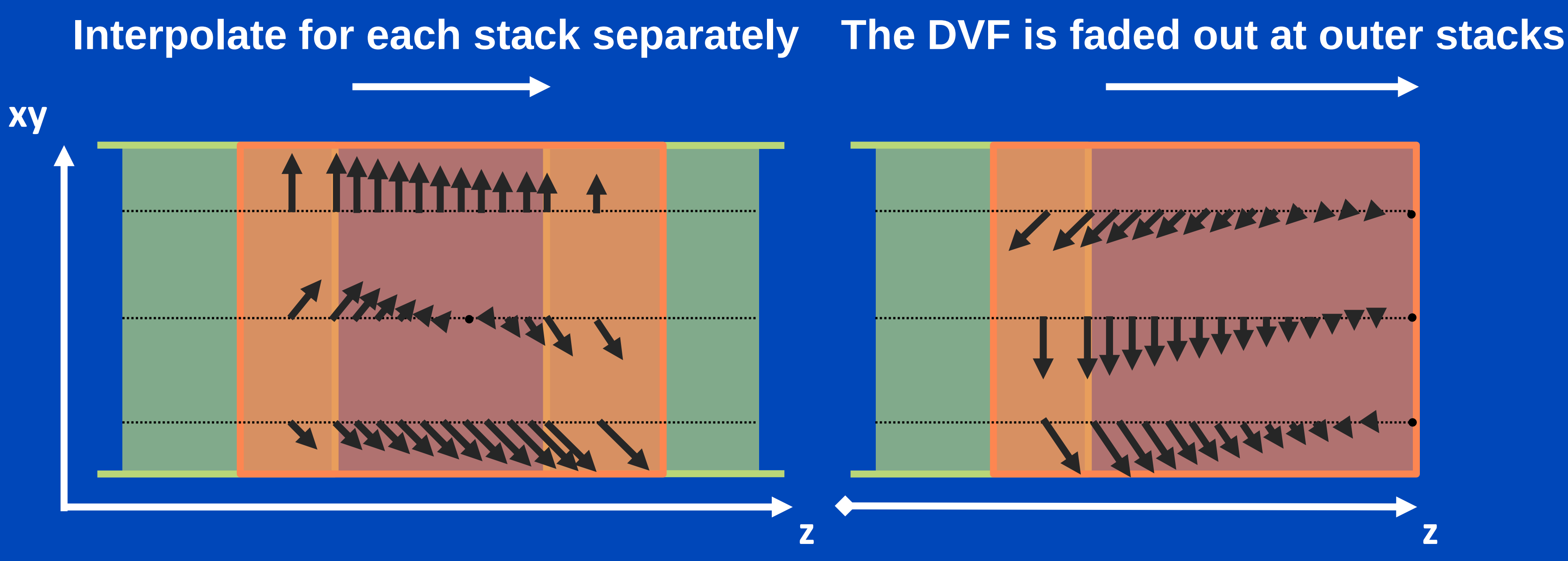


2D illustration for four stacks. One vector per overlap represents the respective DVF

Methods

DVF interpolation

- For each stack a linear interpolation between the upper edge of the lower overlap and the lower edge of the upper overlap is performed



2D illustration for a stack and its neighbors. Three vectors per overlap represent the DVF.

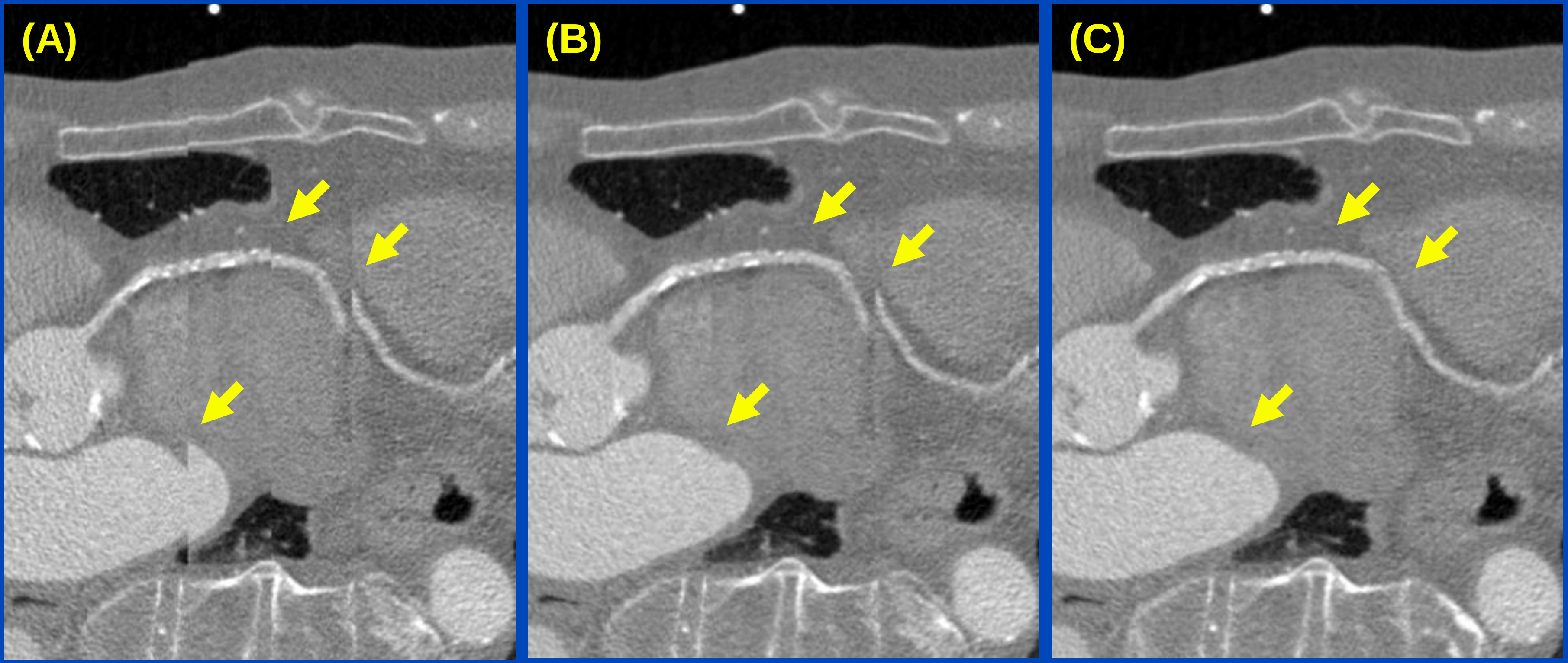
Materials

- Patient data acquired with a Somatom Definition AS+ (Siemens Healthineers, Forchheim Germany).
- 3 patient data sets with stack transition artifacts.
- Standard partial scan WFBP reconstructions and reconstructions using motion compensation for coronary arteries (PAMoCo*).
- t_{rot} = 285 ms
- eff. mAs = 61 - 125 mAs
- Tube voltage = 80 - 125 kV
- CTDI vol = 4 - 34 mGy
- DLP = 59 - 560 mGy cm

* Hahn, Bruder, Rohkohl, Allmendinger, Stierstorfer, Flohr, Kachelrieß, Motion compensation in the region of the coronary arteries based on partial angle reconstructions from short scan CT data", Medical Physics, 2017.

Results
Patient A

Standard recon. | **Demons** | **Multi resolution Demons**

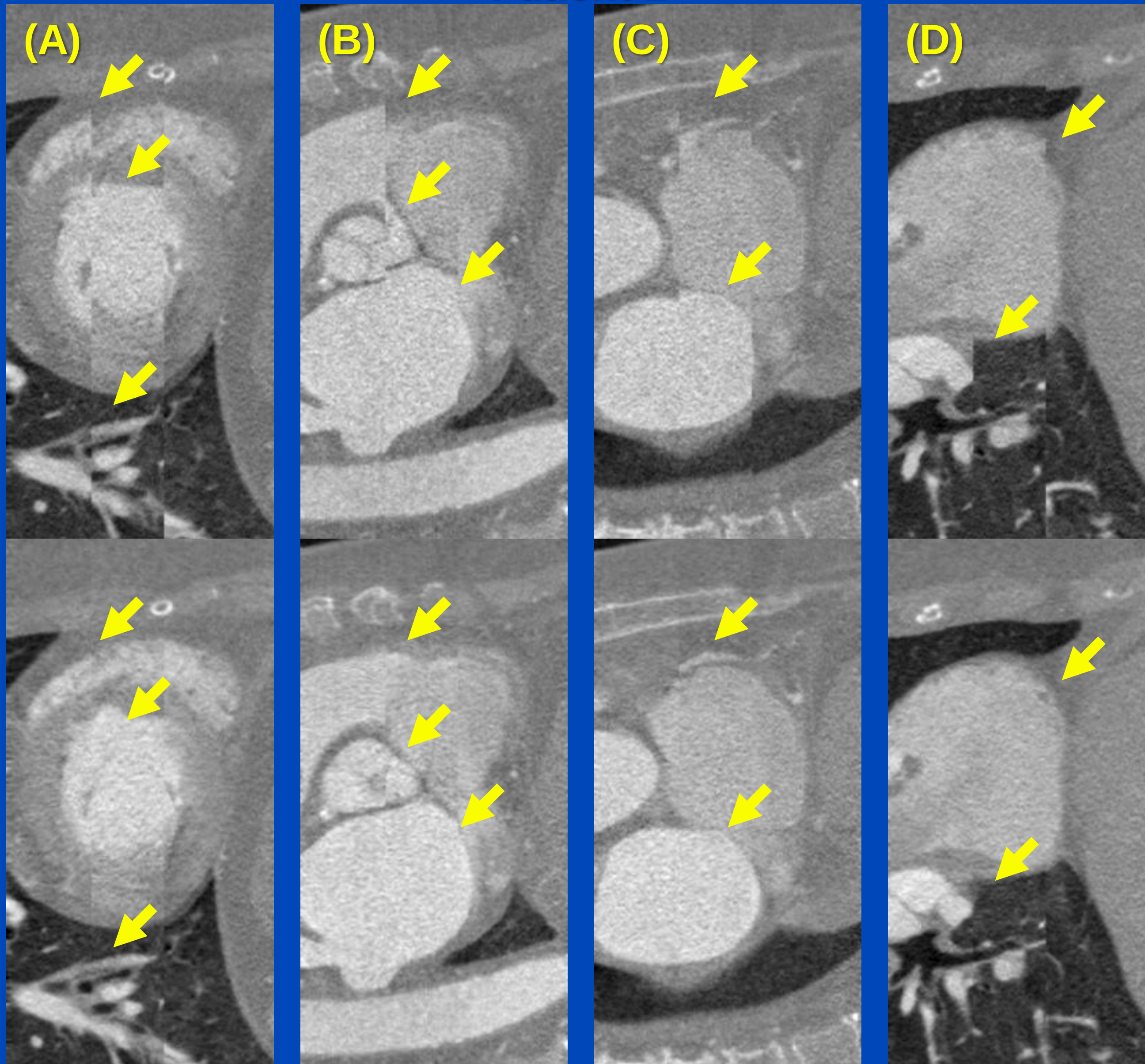


Curved MPR with coronary artery, extracted from a standard partial scan reconstruction (A) and the volume processed with the symmetric Demons algorithm using single (B) and multi resolution (C).

Results
Patient B

Standard recon.

Multi resolution Demons



C = 0 HU, W = 2000 HU

SIEMENS

dkfz.

Results

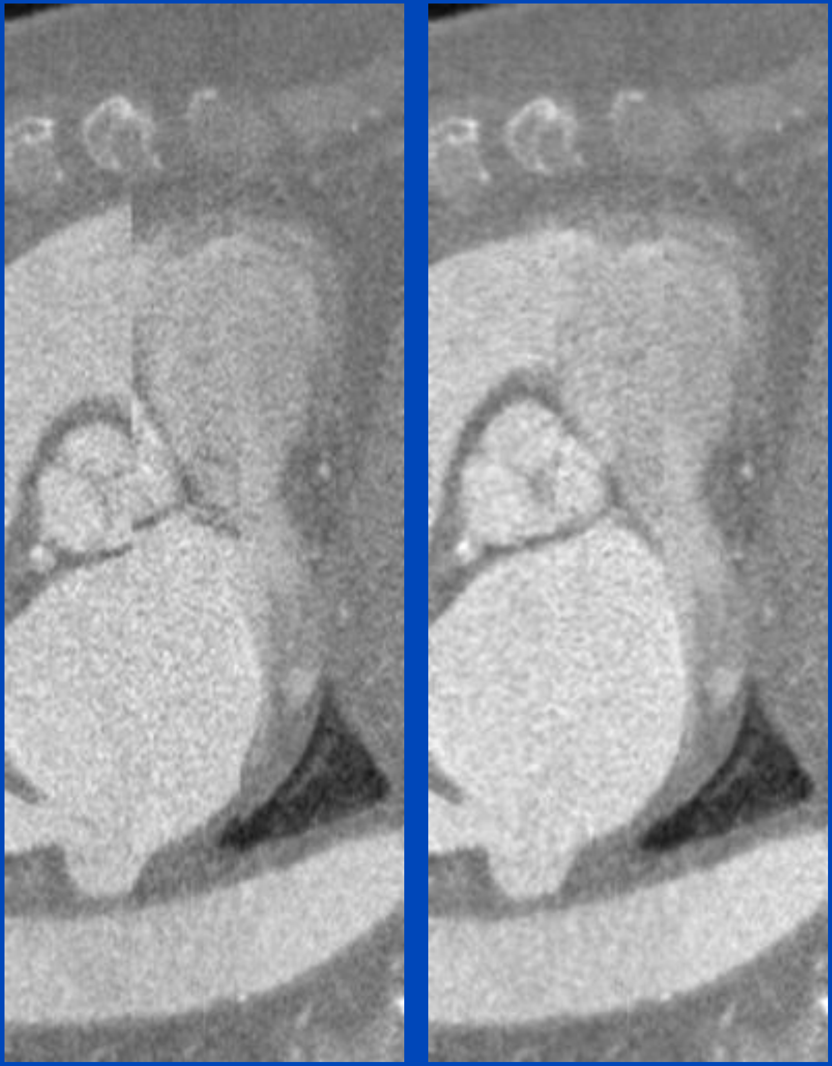
Patient B

Standard recon.

Multi res. symmetric Demons

C = 0 HU, W = 2000 HU

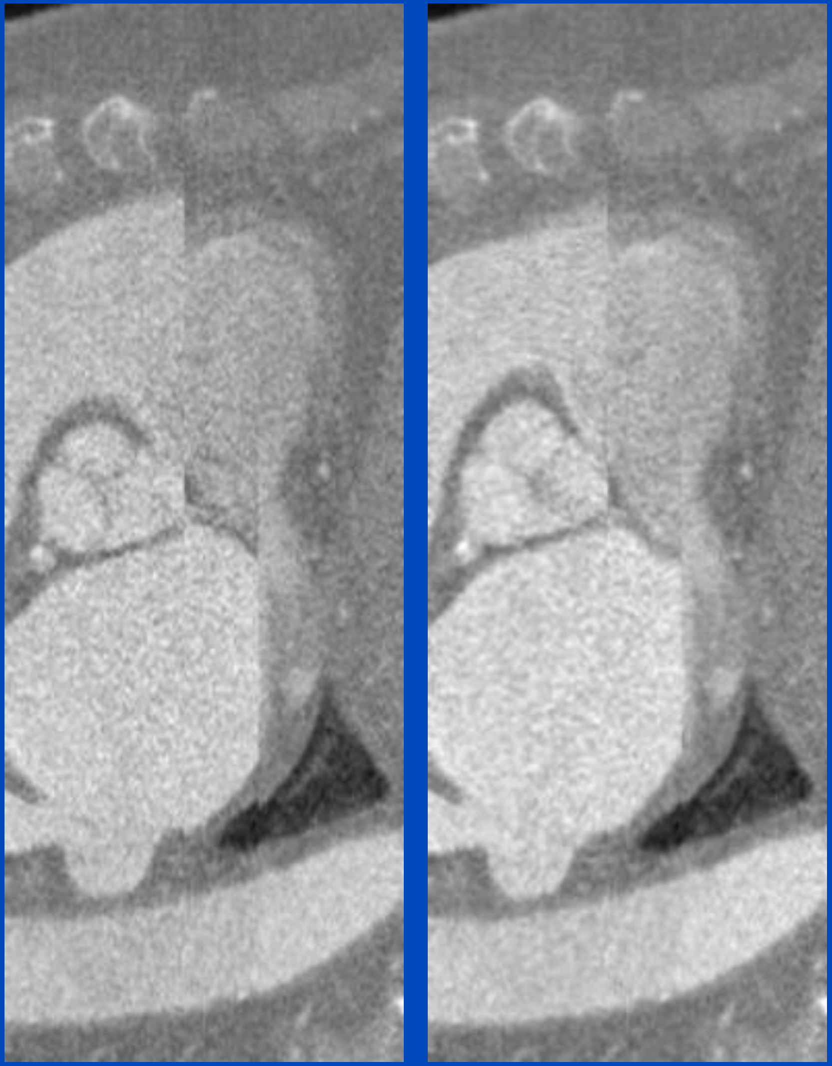
Results
Patient B



C = 0 HU, W = 2000 HU

Results
Patient B

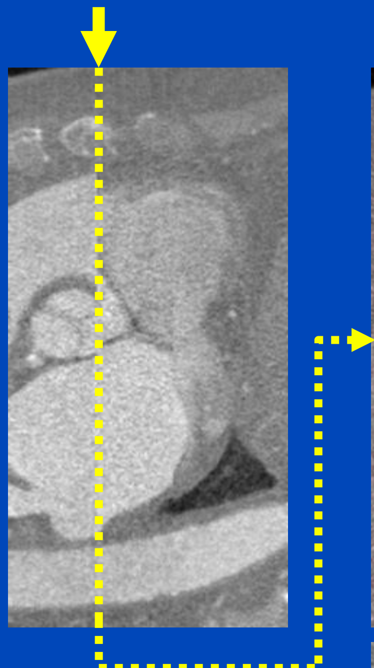
Standard recon.	Multi res. symmetric Demons



C = 0 HU, W = 2000 HU

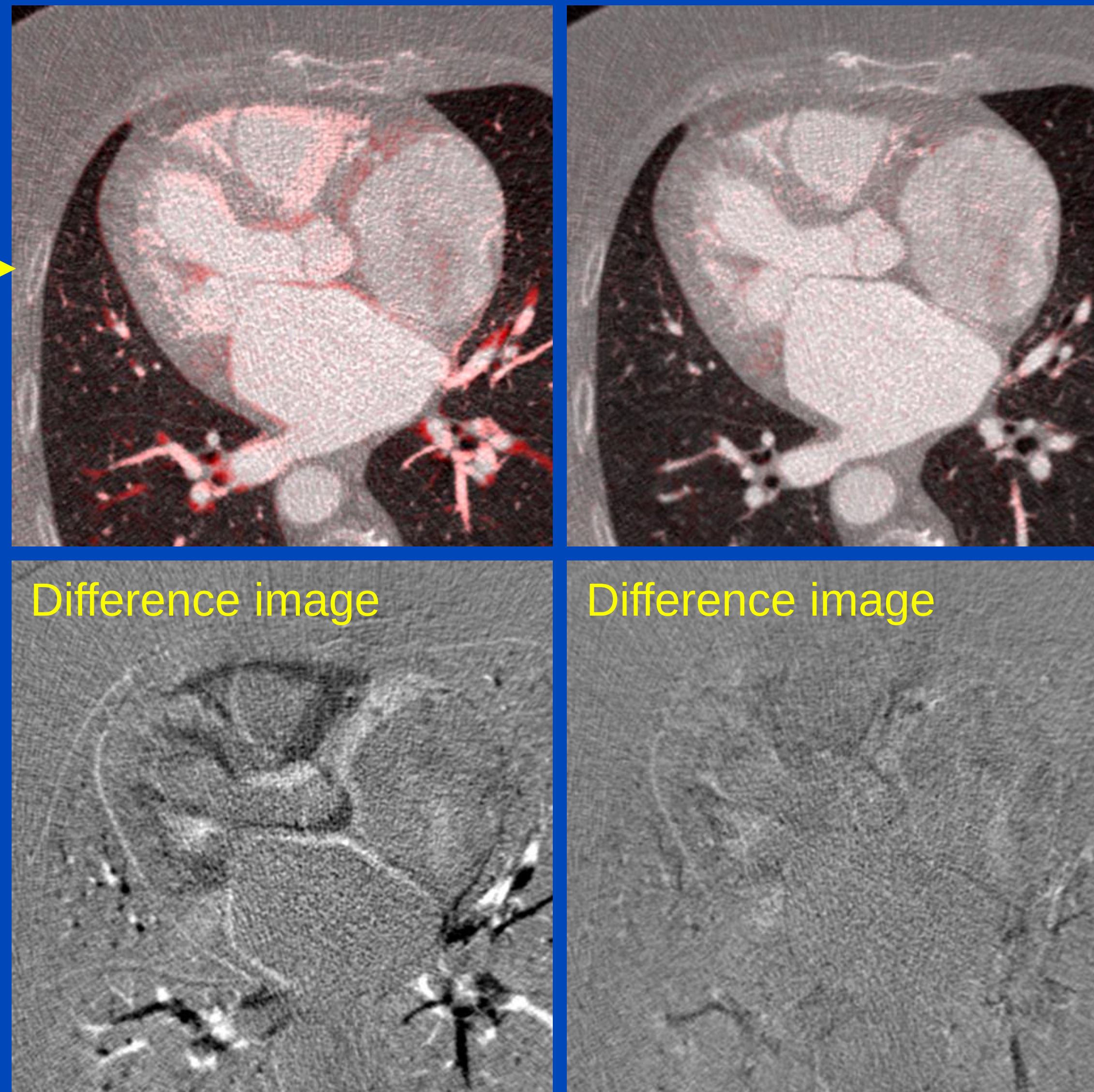
Results

Patient B



Standard recon.

Multi res. symmetric Demons



1st row: Slices (lower stack) at a stack transition. Colored overlay represents the absolute difference to the respective slice in the upper stack.
2nd row: Regular difference images. Upper – lower stack.

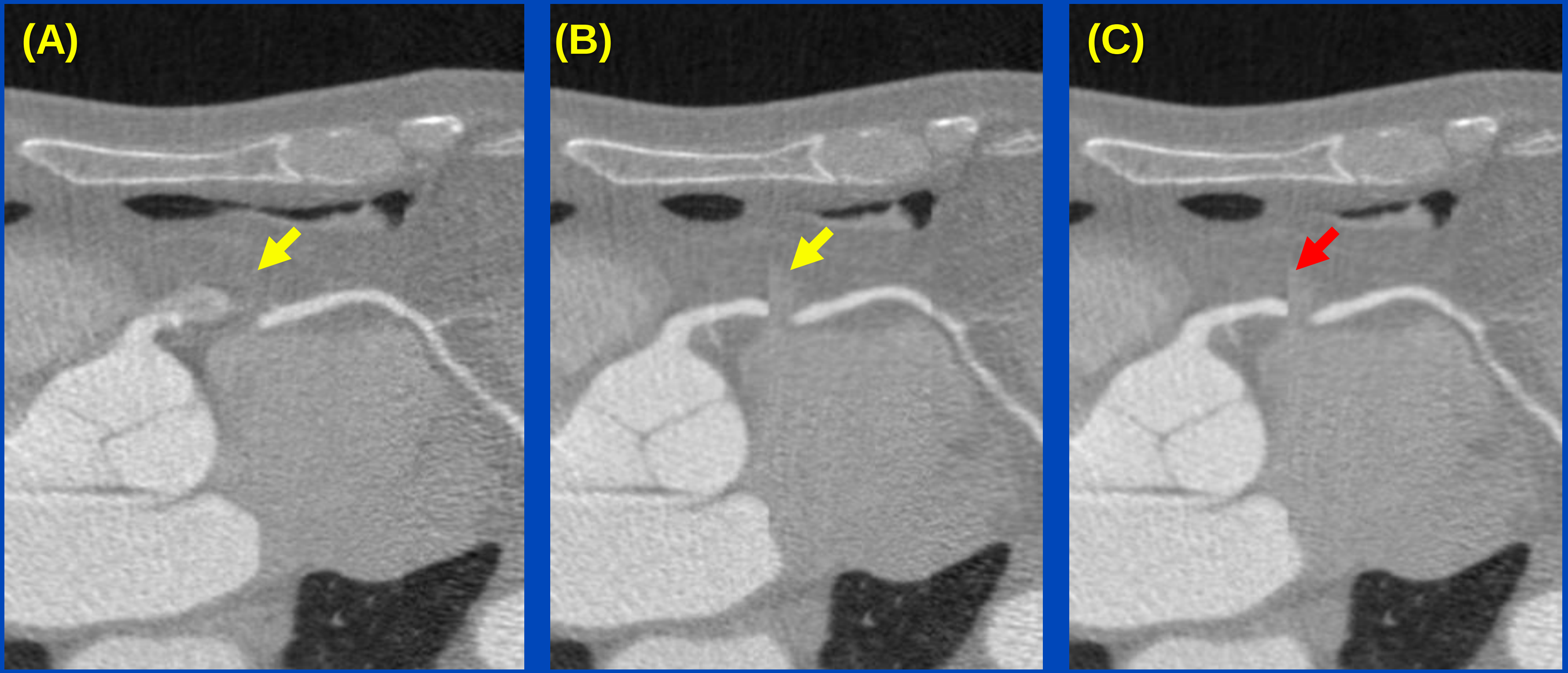
SIEMENS

C = 0 HU, W = 2000 HU

dkfz.

Results
Patient C

Standard recon. | **PAMoCo*** | **Multi resolution Demons**



Curved MPR with coronary artery, extracted from a standard partial scan reconstruction (A) a PAMoCo reconstruction (B) and the volume processed with multi resolution symmetric Demons algorithm (C).

SIEMENS

* Hahn, Bruder, Rohkohl, Allmendinger, Stierstorfer, Flohr, Kachelrieß, Motion compensation in the region of the coronary arteries based on partial angle reconstructions from short scan CT data”, Medical Physics, 2017.

dkfz.

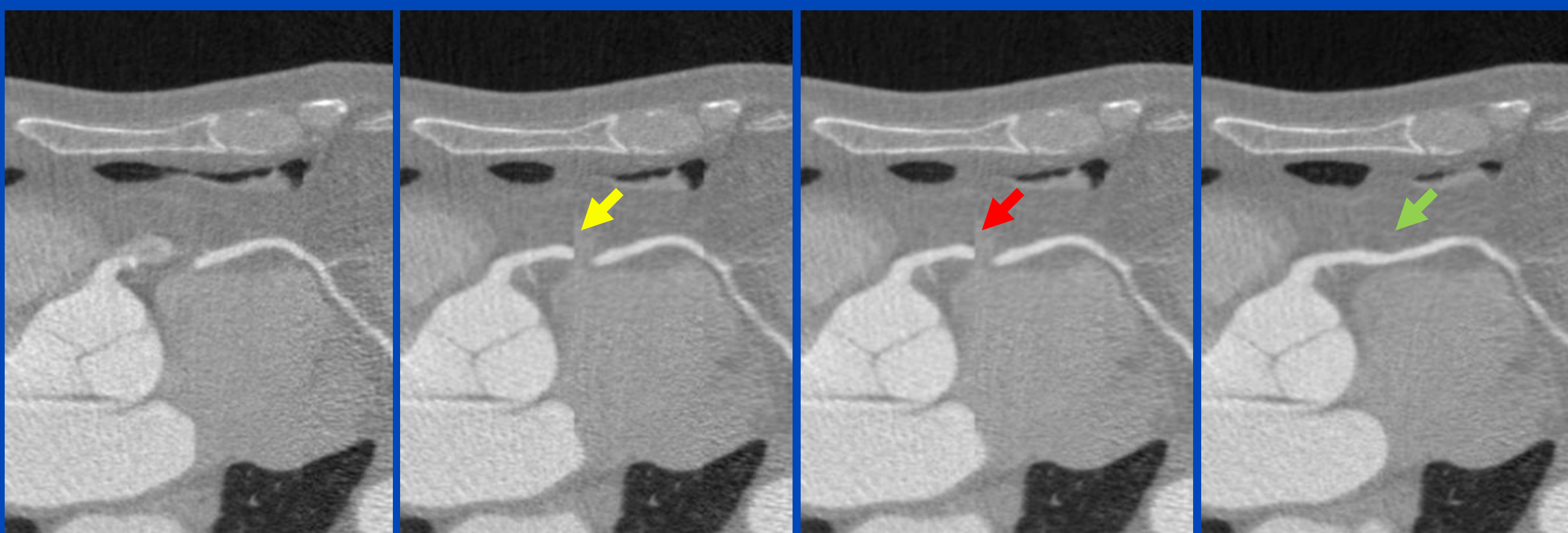
Conclusions and Outlook

- The symmetric Demons algorithm removes most stack transition artifacts.
- The algorithm computes smooth DVFs that transform the volume in a realistic way
- In case of strong displacements (multiple mm) where there is little or no overlap between the to be registered structures, some artifacts may obviously remain.
- Outlook: The Demons algorithm may be initialized with a different algorithm that manages large deformations better.

Conclusions and Outlook

- **Outlook: The Demons algorithm may be initialized with a different algorithm that manages large deformations better.**

Standard recon. | **PAMoCo** | **Multi resolution Demons** | **Patch-based* initialized multi resolution Demons**



SIEMENS

*Lebedev, Fournie, Stierstorfer, Kachelrieß et al. Stack Transition Artifact Removal for Cardiac CT using Patch-Based Similarities, Proceedings SPIE Medical Imaging, 2018

dkfz.

Thank You!

This presentation will soon be available at www.dkfz.de/ct.

Job opportunities through DKFZ's international PhD or Postdoctoral Fellowship programs (www.dkfz.de), or through Marc Kachelriess (marc.kachelriess@dkfz.de).

Parts of the reconstruction software were provided by RayConStruct® GmbH, Nürnberg, Germany.