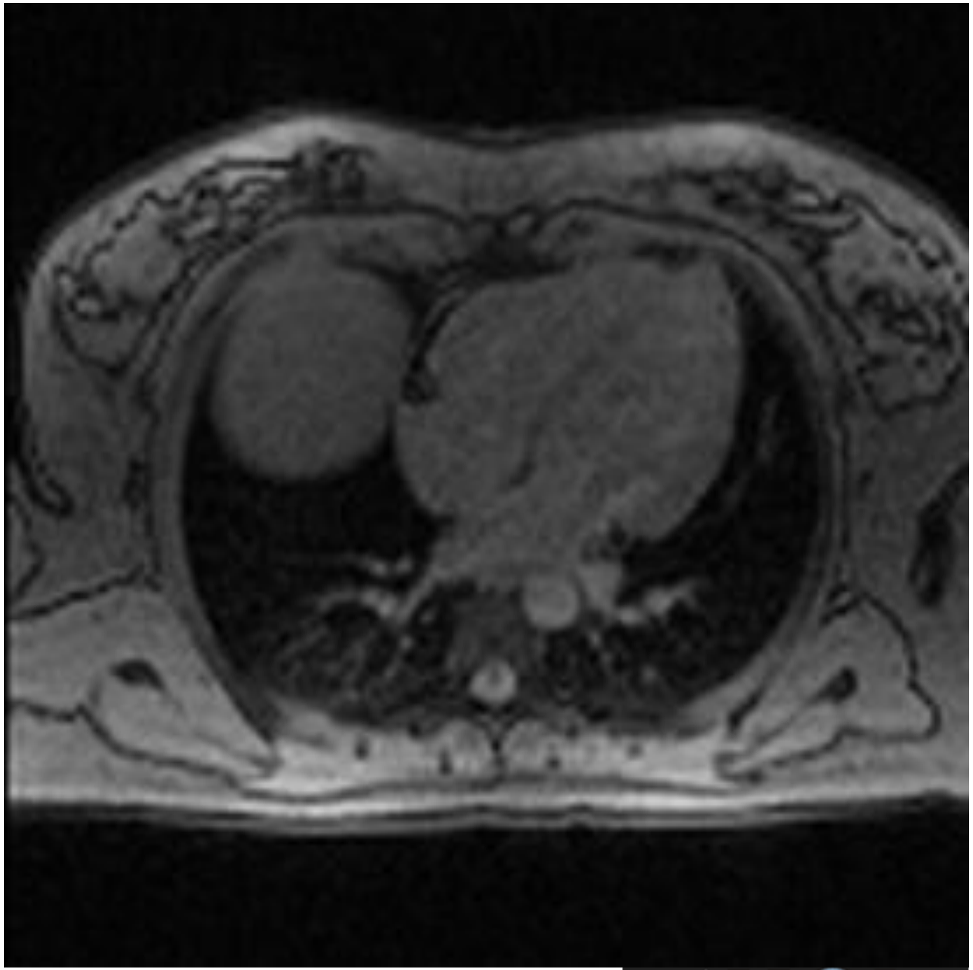
# A heterogeneous image reconstruction system for clinical Magnetic Resonance

Grzegorz Tomasz Kowalik[1], Jennifer Anne Steeden[1], Bejal Pandya[1], David Atkinson[2], Andrew Taylor[1], and Vivek Muthurangu[1]

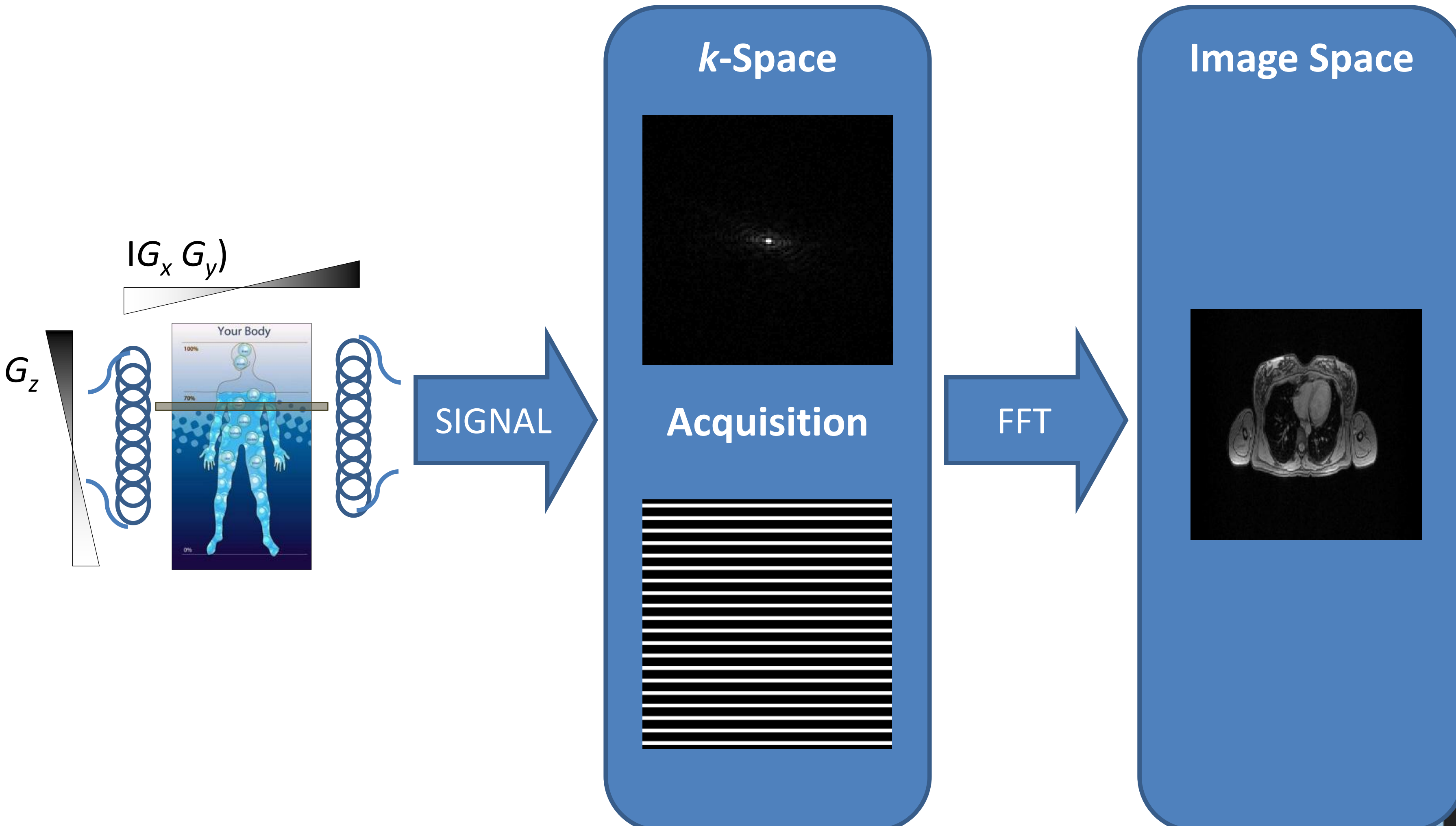
[1]*Institute of Cardiovascular Science, UCL Centre for Cardiovascular Imaging, London, United Kingdom,* [2]*Centre for Medical Imaging, UCL Division of Medicine, London, United Kingdom*

# MRI – general overview

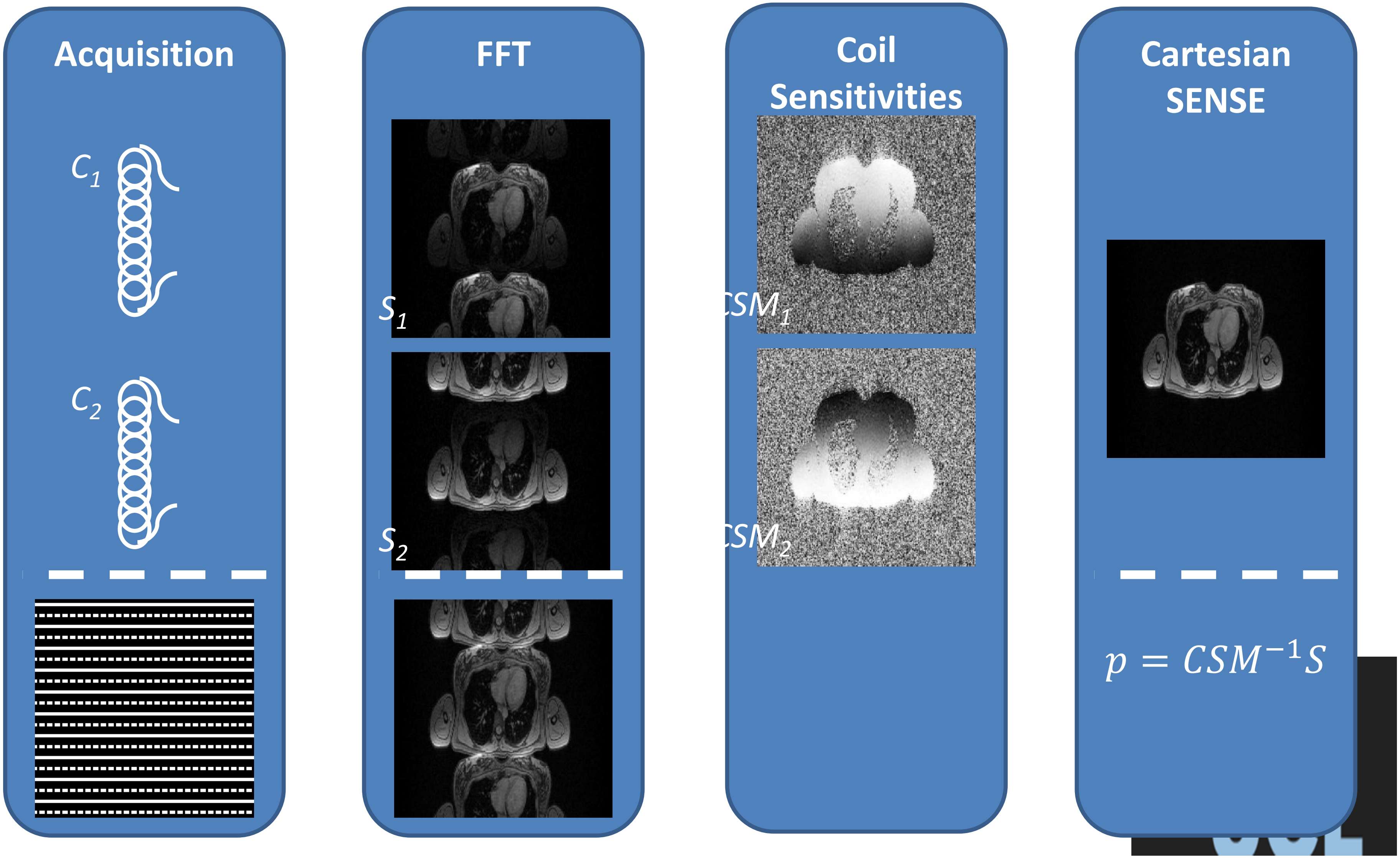
- Imaging by exploiting tissue magnetic properties
- Non-ionising procedure
- Good contrast between soft tissues
- Applications
  - Brain imaging
  - Cancer diagnosis
  - Fat quantification
  - Cardiac imaging
  - and many more



# Imaging process

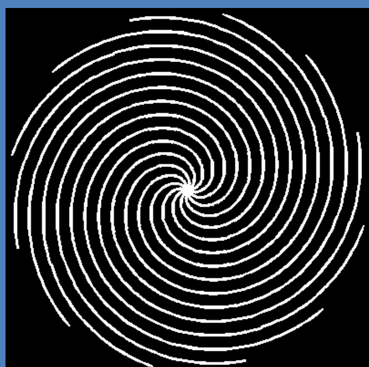


# Undersampling

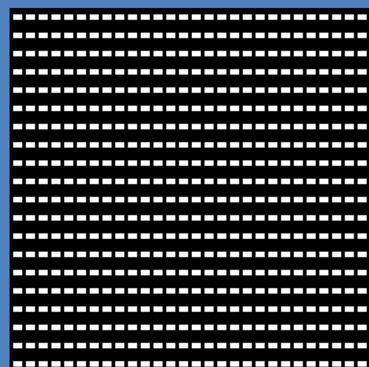


# Arbitrary trajectory

**Acquisition**

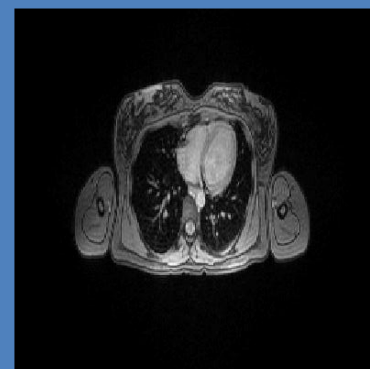


**Gridding**



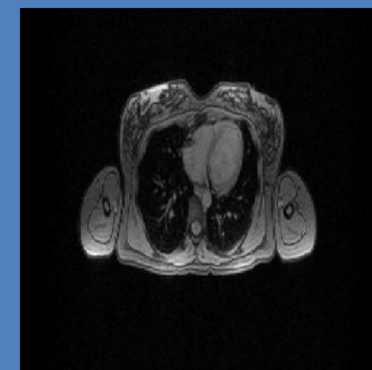
**Convolution onto Cartesian grid**

**FFT**



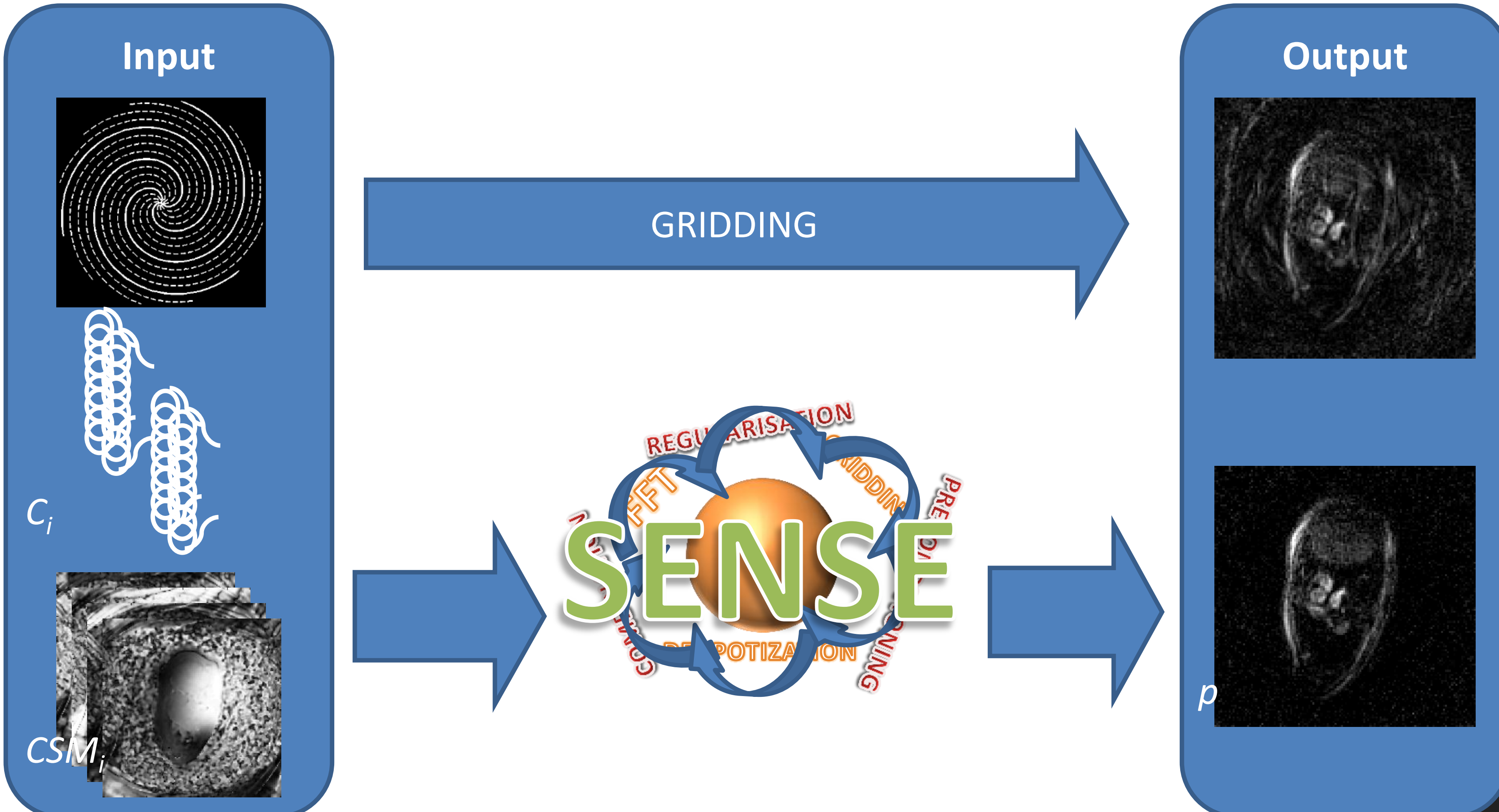
**Weighted image**

**De-apodization**



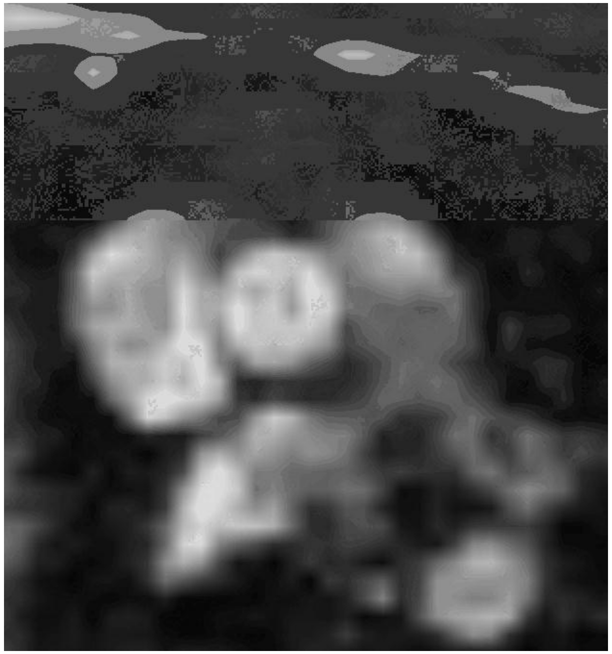
$$\hat{f}(\omega) * \hat{g}(\omega) \Leftrightarrow f(x) \cdot g(x)$$

# SENSE for arbitrary trajectories



# Cardiac MR – flow quantification

- Cardiac imaging requires fast acquisition
- Requires 2 sets of *k*-space data
- Real-time spiral PCMR sequence
  - High spatiotemporal resolution (~44ms/Flow-frame)
  - Slow image reconstruction



# Project goals

- Challenges
  - Use of advanced MRI sequences is limited by their reconstruction time
  - GPU implementations exist but run in offline mode
- Proposed solution
  - GPU based image reconstructor integrated into scanner’s system
- Benefits
  - Seamless reconstruction process
  - Translation of advanced MRI algorithms into clinical environment

# Continuous cardiac output assessment

- Complete overview of cardiovascular system
- Theoretically possible but practically not feasible

  ~10min of acquisition ➔ ~1.5h of reconstruction
- Reconstruction has to be faster than acquisition
- We need a super-computer under a desk

# Implementation

- Sensitivity Encoding for arbitrary trajectories
  - Algorithm’s description
  - Reconstruction bottleneck
  - Our GPU implementation
- Networking scheme
  - Remote execution
  - Data transfer management

# SENSE - mathematical formulation

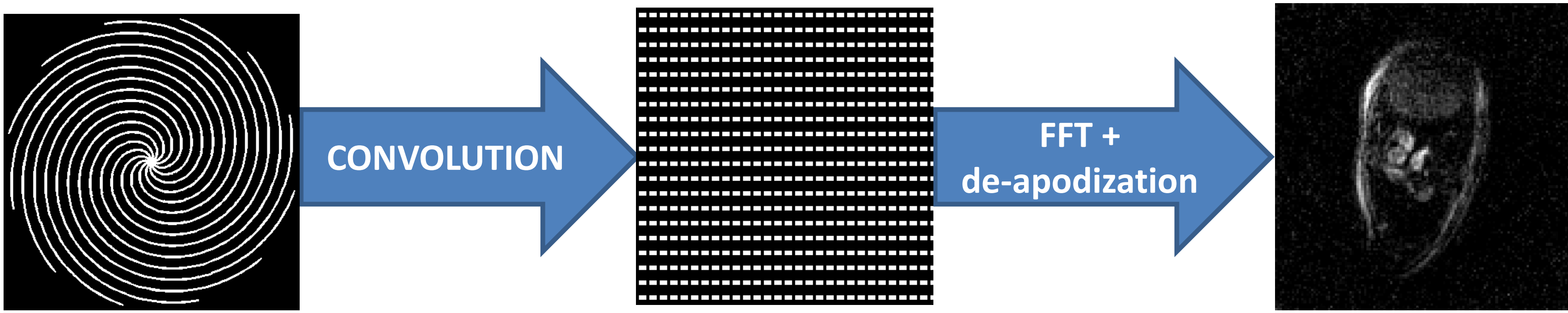
- Imaging process is based on a set of linear equations

$$E\rho = s$$

$$\rho = (E^H E)^{-1} E^H s$$

- *E*, $E^H$ evaluate into
  - Gridding
  - FFT
  - Matrix combinations with *coil sensitivity maps*

# Reconstruction bottleneck - Gridding

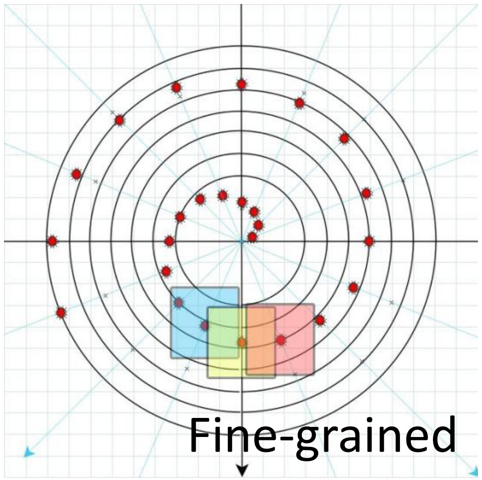


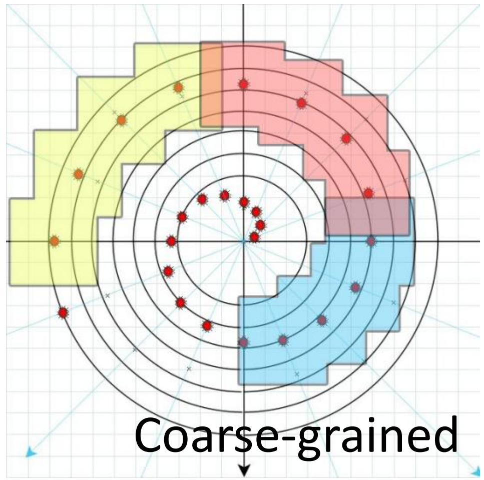

- Convolution onto Cartesian grid

$$\hat{f}(\omega) * \hat{g}(\omega) \Leftrightarrow f(x) \cdot g(x)$$

- No data dependency
- CPU implementation
  - Task is split across multiple threads
  - Gridding kernel values are stored in shared array
  - Up to 89% of reconstruction time

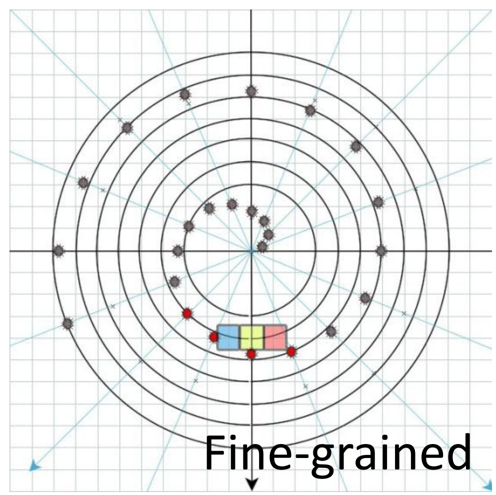
# Gridding – GPU implementation

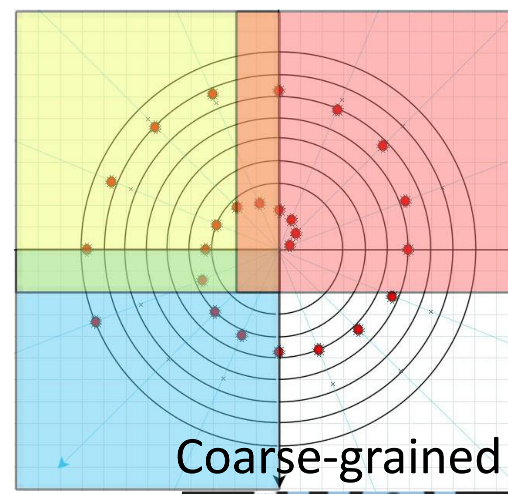



- **Input-driven assignment**
  - Optimal number of memory reads
  - Non-coalesced memory writes
  - Accumulative global memory writes
  - Shared memory as software managed cache

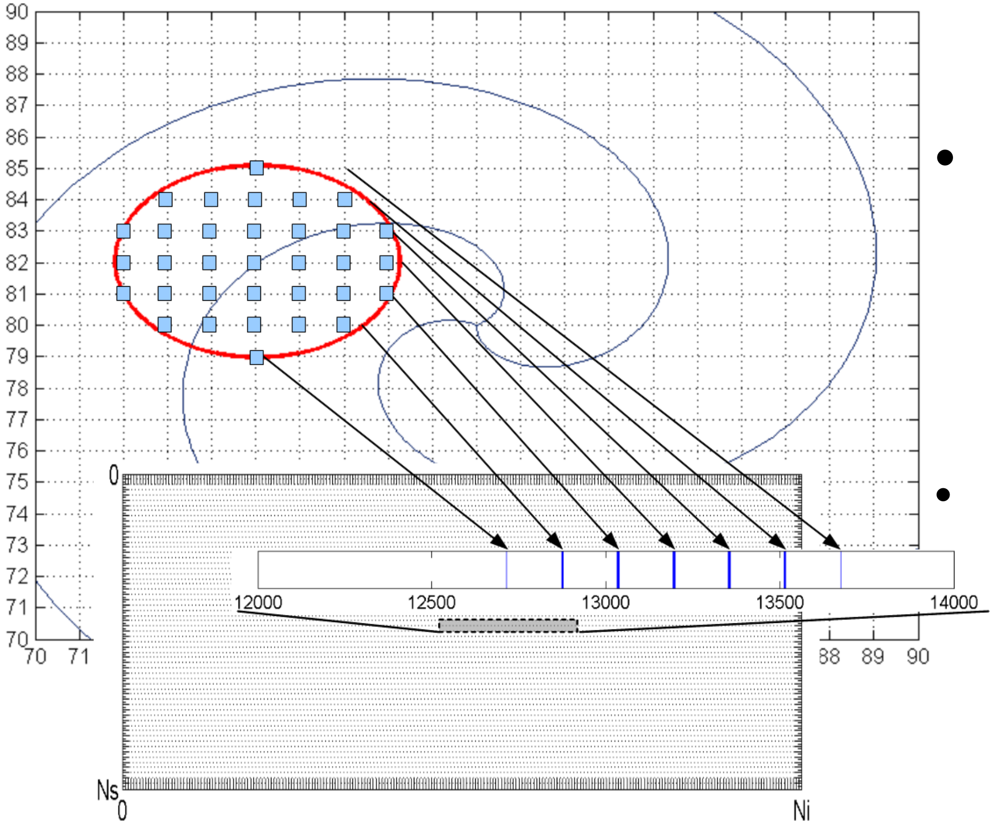
- **Output-driven assignment**
  - Requires pre-sorting of input data
  - Optimal number of memory writes requires sub-optimal number of memory reads
  - Optimal number of memory reads requires padding
  - Shared memory as software managed cache




A. Gregerson, *Memory* (2007) (available at http://www.nvidia.com/docs/IO/47905/ECE757_Project_Report_Gregerson.pdf).

# Gridding – GPU implementation




- Convolution as matrix multiplication

$$S_{N_s,N_c} = G_{N_s,N_i}\rho_{N_i,N_c}$$

$$\rho_{N_i,N_c} = G^H_{N_s,N_i}S_{N_s,N_c}$$

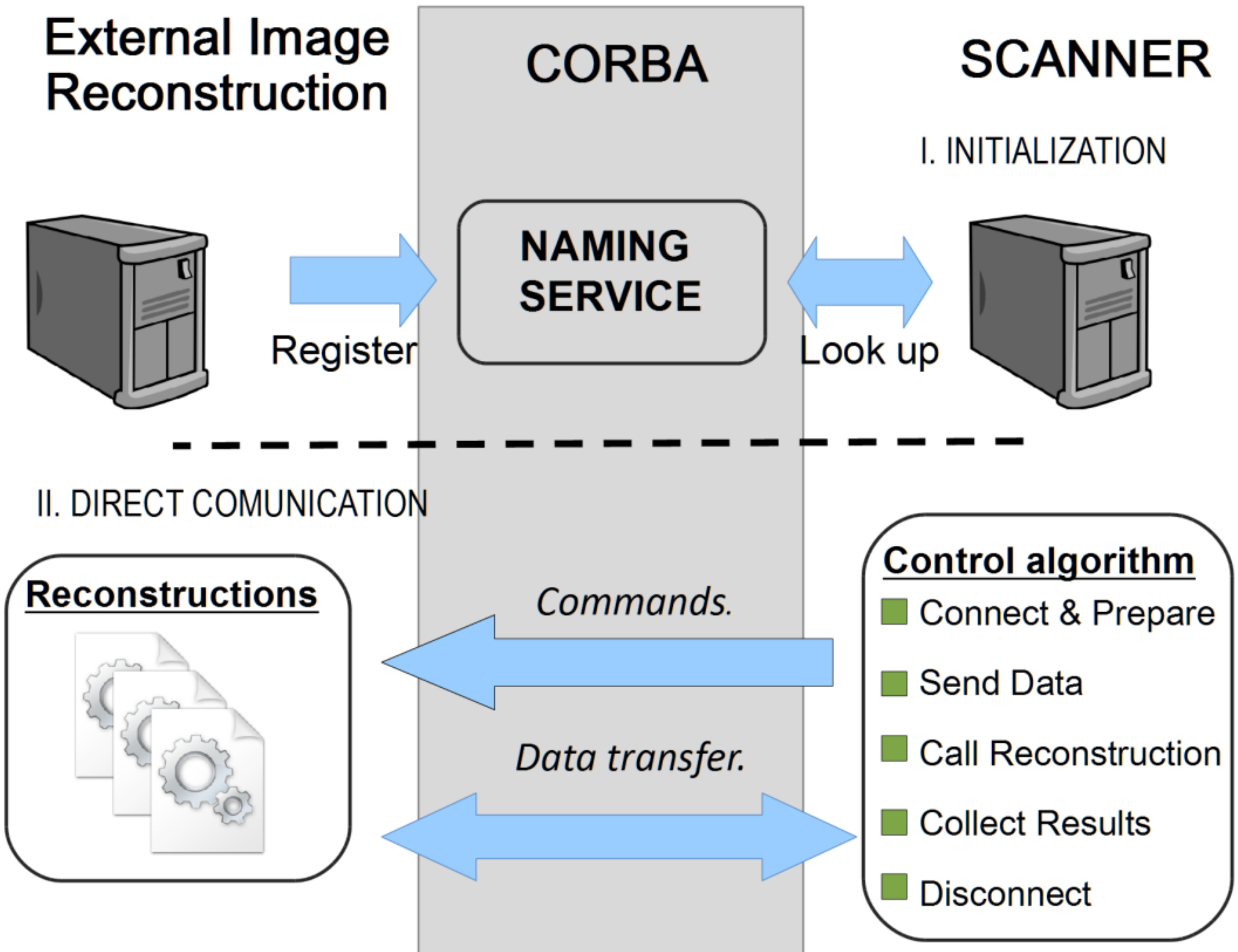
- Heavily sparse matrix
  - $N_i$ = 128x128 $N_s$ = 2205x3 Kernel size = 8x8
  - Full-size → 108380160
  - Non-zero values → 423360

  **Less than 0.4%**
- Compressed Sparse Row (CSR) Format

# GPU implementation

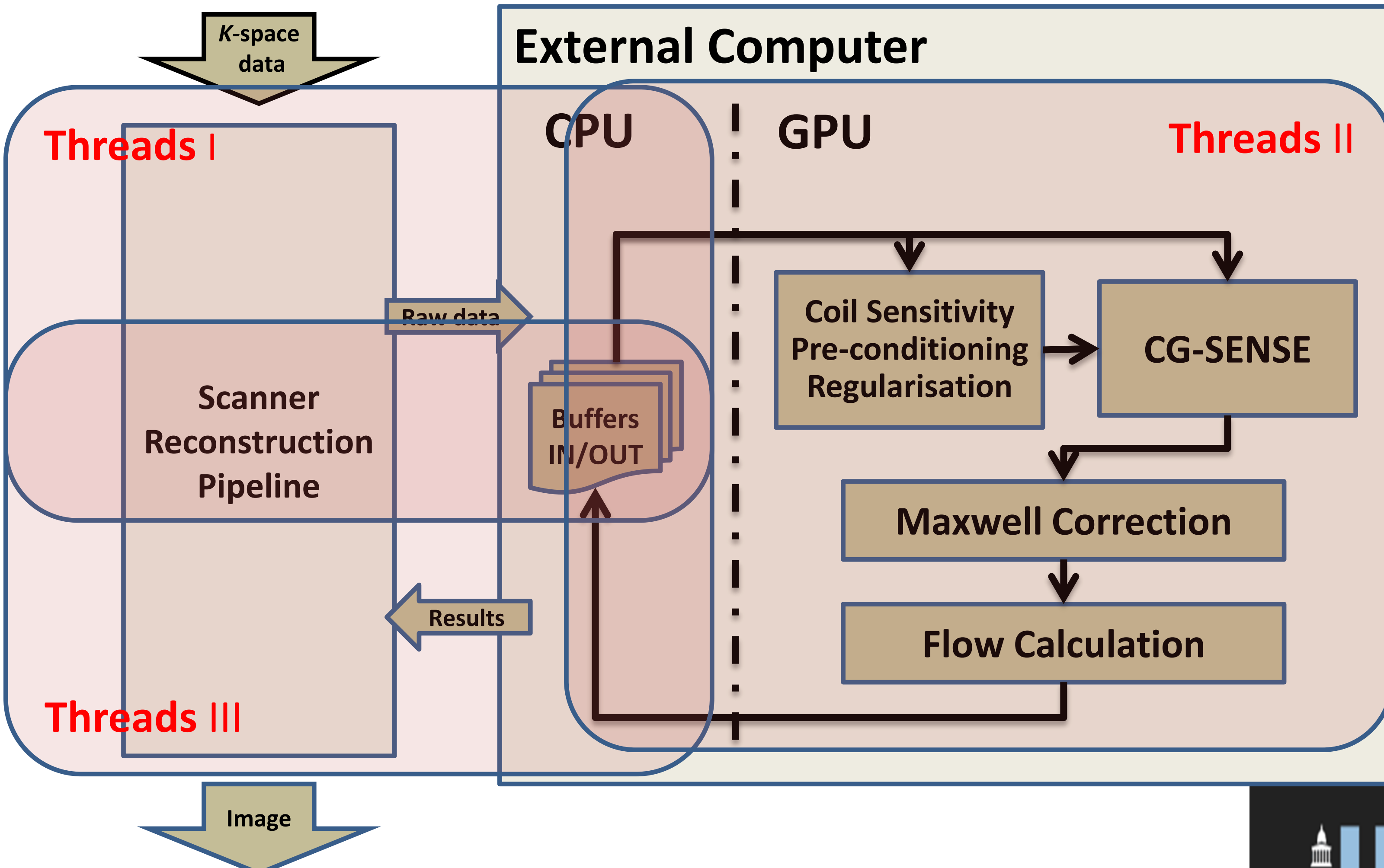
- CUDA 4.0 toolkit
  - cublas library – linear algebra
  - cusparse library – sparse matrix operations
  - cufft library – fast Fourier Transformations
- The algorithm was fully implemented on GPU
  - Synchronisation is needed only to check iteration process convergence

# Networking (1)




JCL

# Networking (2)



# Methods (1)

- Native reconstruction system
  - 8 Intel Xeon E5440 2.83 GHz (4 cores)
  - 16 GB RAM
  - C++ based, multi-threaded programing environment
- Network connection
  - Half-duplex, 1Gbps, ethernet connection
- External computer
  - Intel i7 920, 2.66 GHz (4 cores)
  - 9GB RAM
  - GTX 480 1.5GB, 480 CUDA cores

UCL

# Methods (2)

- Validation study
  - 60 frames (retrospective reconstruction)
- Continuous flow measurement
  - 20 healthy volunteers
  - Aortic flow quantification during an exercise
  - 13980 frames (online)
  - ~10 minutes acquisition
  - ~6GB of raw data

# Results - CPU vs GPU implementation

- Validation: -0.4 ± 0.8ml
- Timing

| | | CPU [ms] | GPU [ms] | CPU / GPU |
|---|---|---|---|---|
| **Per iteration** | FFT | 287.67 | 73.95 | 4 |
| | Gridding | 2674.75 | 58.48 | 46 |
| | Matrix combination | 245.69 | 4.24 | 58 |
| | Preconditioning | 52.95 | 1.00 | 53 |
| | Total | 3288.32 | 144.83 | 23 |
| **Per 60 frames (test data set)** | Total | 24115.24 | 1581.67 | 15 |

Tab. 1. Comparison of execution times of the PCMR reconstruction implementations.

# Results - system workload

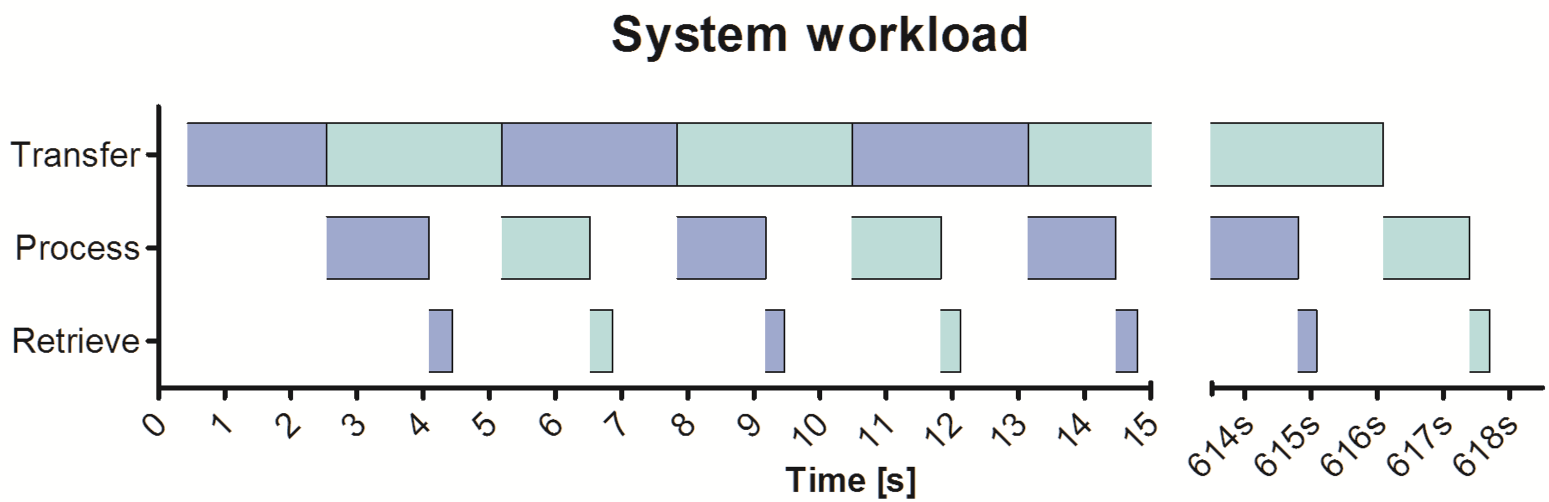



Fig. 1. Alternating colours represent two different processing buffers.

| | | CPU [ms] | GPU [ms] | CPU / GPU |
|---|---|---|---|---|
| Per 13980 frames (buffered recon.) | Total | 5618.85* | 629.24 | 9 |
| | From acquisition end | 4998.61* | 9.00 | 556 |

Tab. 2. Timing for the exercise study.

* CPU times for the entire dataset of 13980 frames were estimated from the CPU reconstruction time required for a subset of 60 frames.

# Results - continuous cardiac output

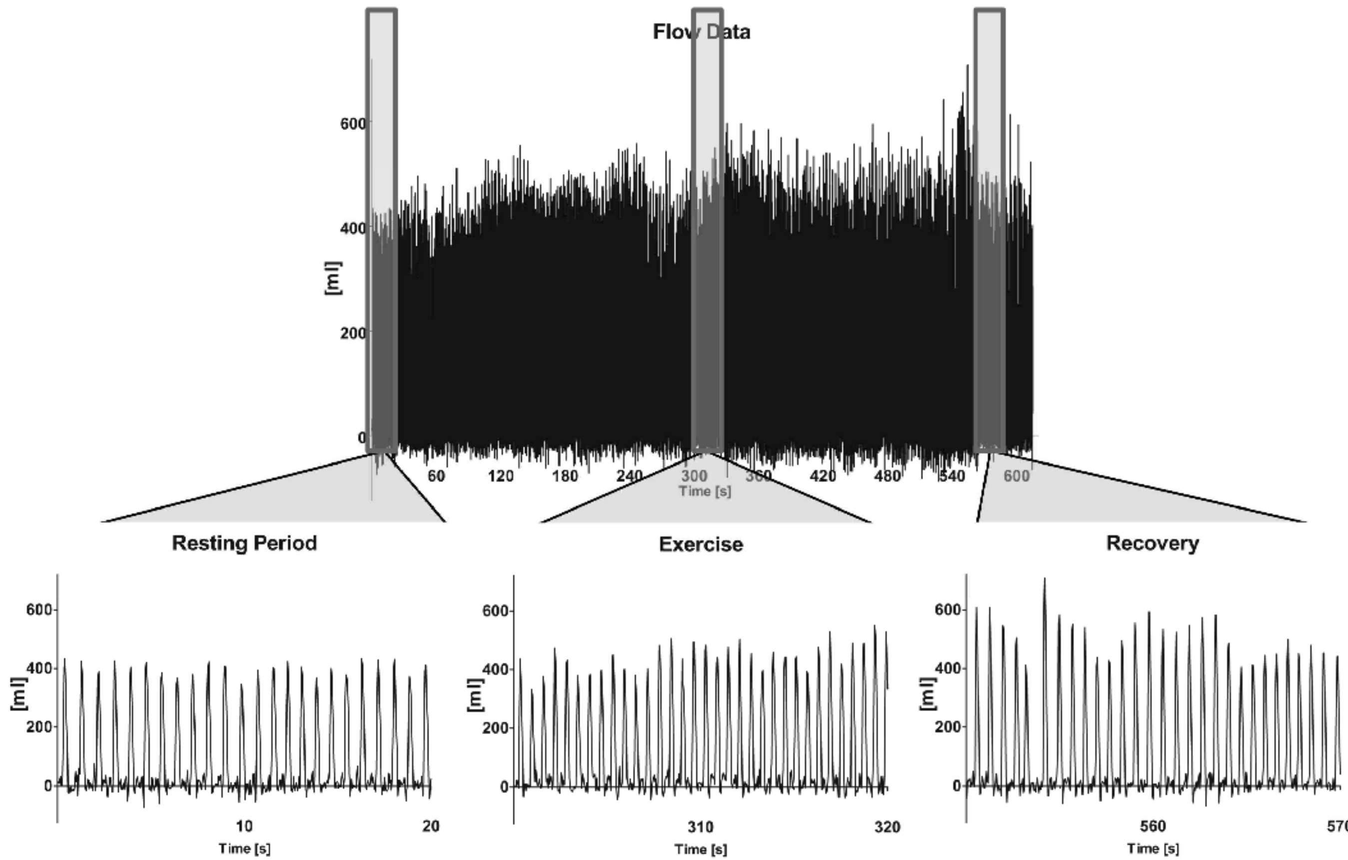



Fig. 2. An example of flow data acquired during continuous real-time PCMR during exercise. Bottom graphs show 20s section of the total data taken at rest, mid exercise and recovery.

# Results - continuous cardiac output

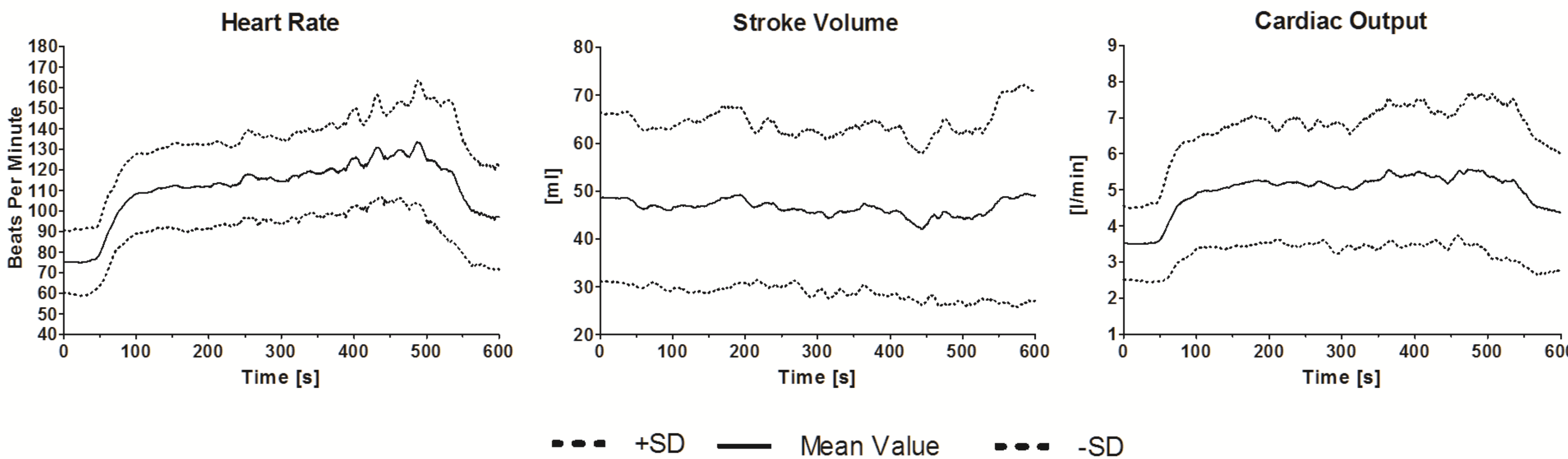



Fig. 3. Averaged exercise data based on continuous flow measurements from the population of 20 volunteers.

# Conclusion

- Online reconstruction of real-time MR data is feasible using integrated GPU reconstructor
- Advanced MRI techniques become feasible within clinical environment
- Data available within seconds to clinicians
- Full integration provides seamless reconstruction

**We believe, integrated GPU reconstruction has the potential to revolutionize the type of MRI sequences are performed on patients and could improve patient diagnosis and management**

UCL

# Thank you

Any questions ?

Great Ormond Street NHS
Hospital for Children
NHS Trust

British Heart Foundation

SIEMENS

UCL