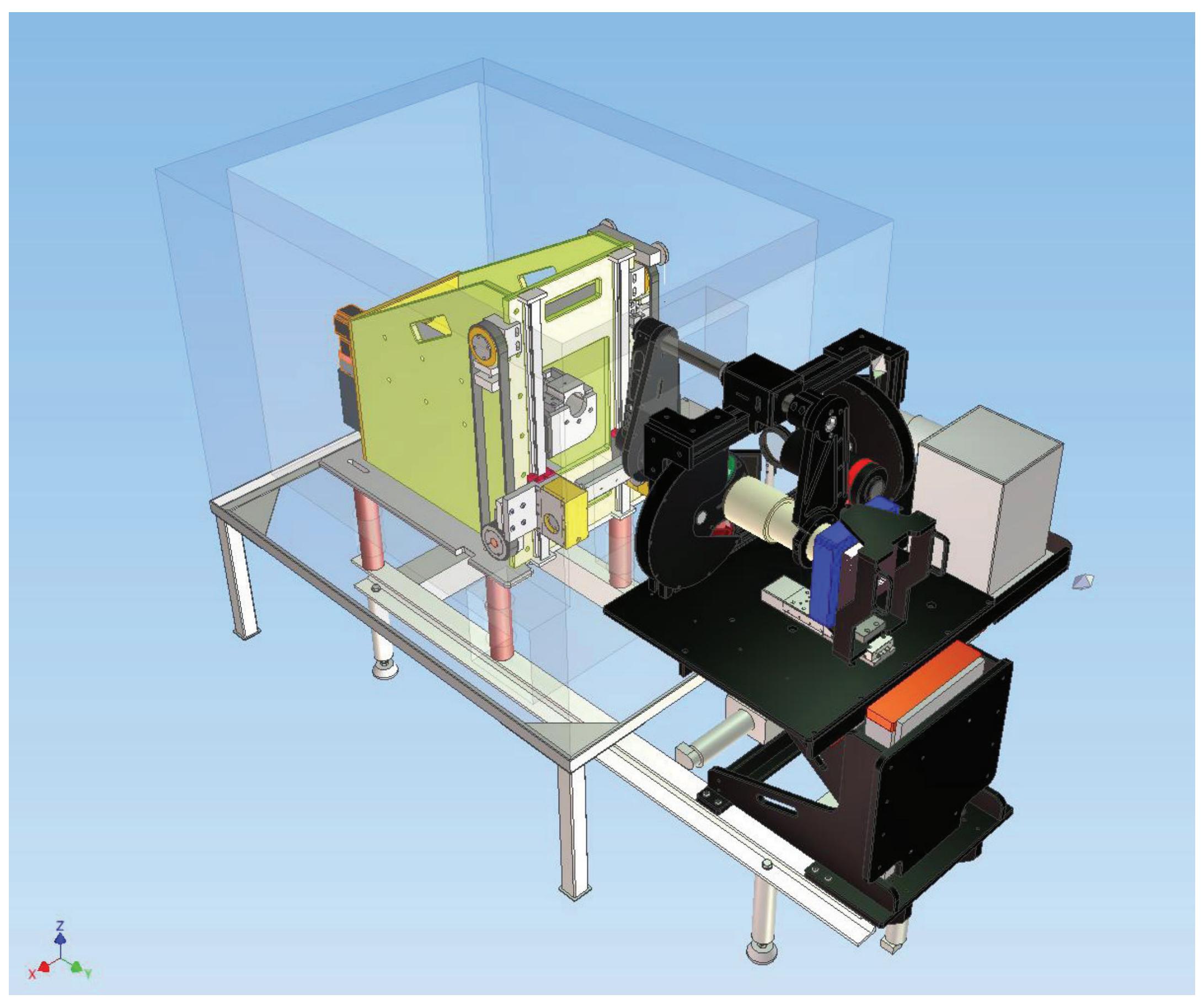
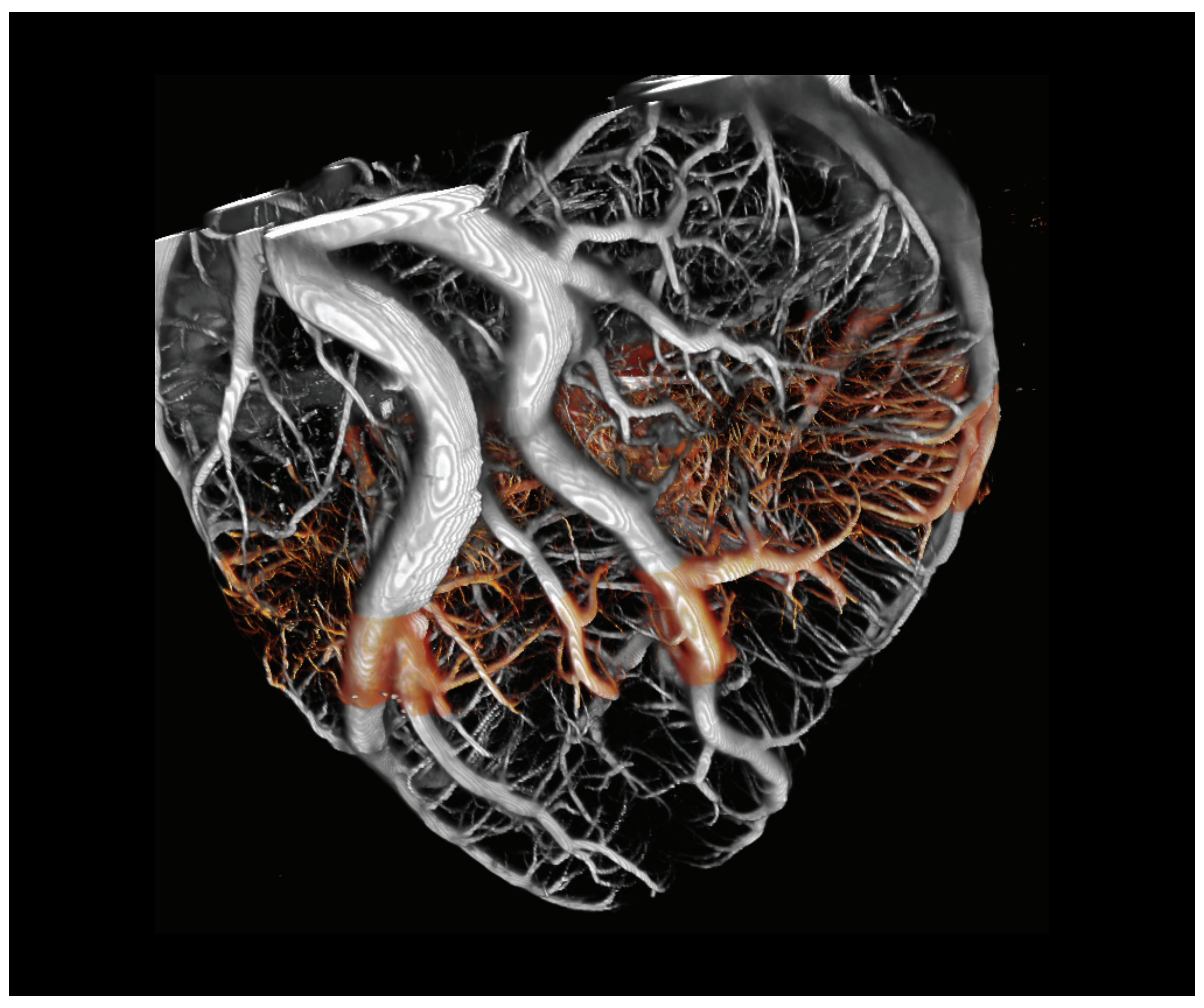
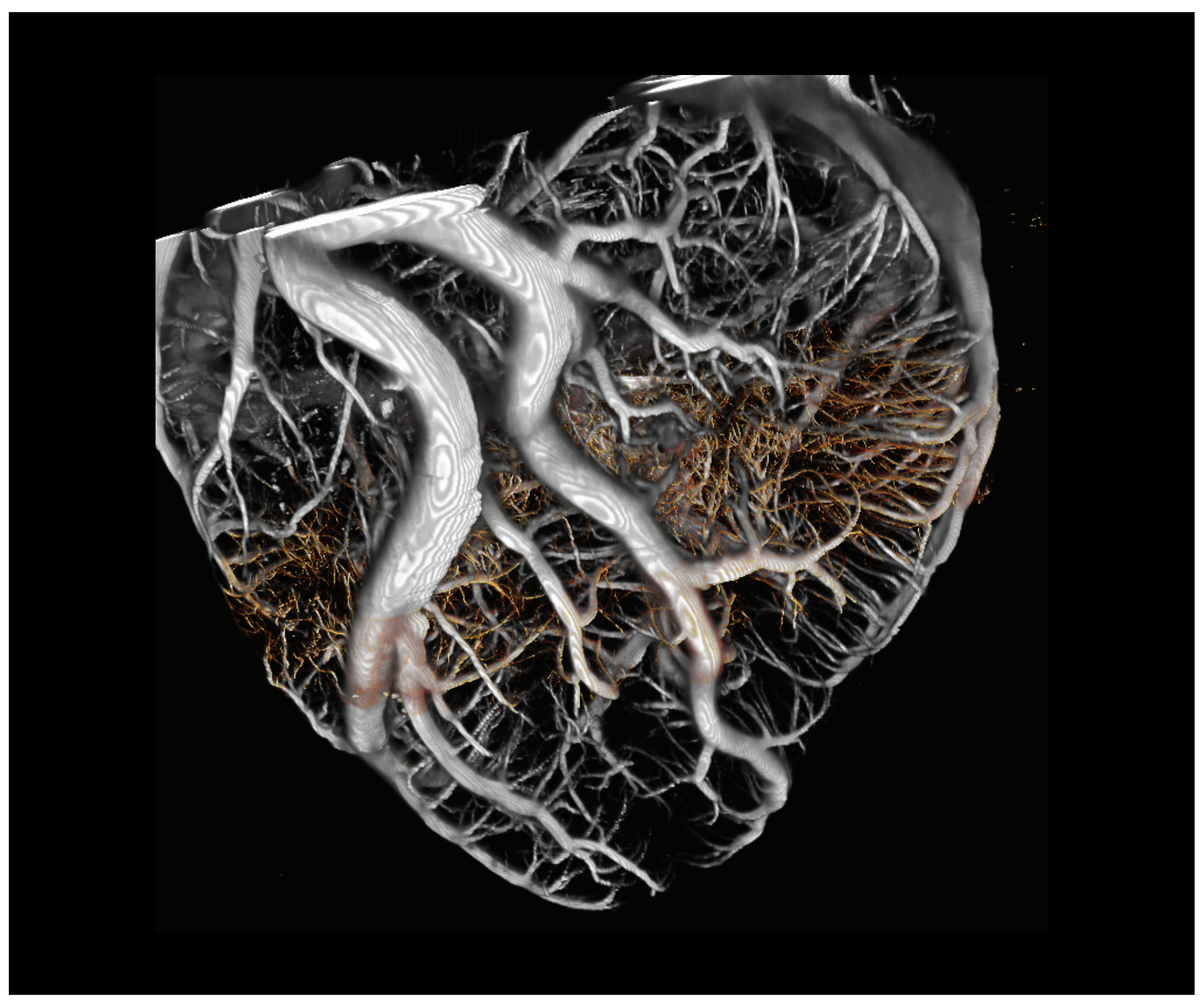
Iterative deblurring of large 3D datasets from Cryomicrotome imaging using an array of GPUs

The aim was to deblur large 3D datasets (4000x4000x4000 pixels) resulting from cryomicrotome images. A parallel iterative spatial convolution strategy for GPU arrays was developed. Our implementation reduced computational time up to 350 times on four GPU's in parallel compared to a single core CPU implementation and outperforms FFT based filtering strategies on GPU's.


Artist impression of the cryomicrotome. The arterial vasculature of the myocardium under investigation is cannulated, flushed with buffered saline and filled with a fluorescent plastic penetrating the arterial vessels up to approximately 10 µm. Subsequently a thin slice of preset distance, e.g. 25 µm, is cut from the frozen sample after which a high resolution image is obtained from the remaining bulk material.


A pig hart is cut in thin slices by the cryomicrotome imaged and recombined as a 3D stack to represent the coronary arterial architecture. The unfiltered data are shown in gray. The brown color depicts a selection of the unfiltered data that will be subjected to the iterative deblurring filter, shown on the right.

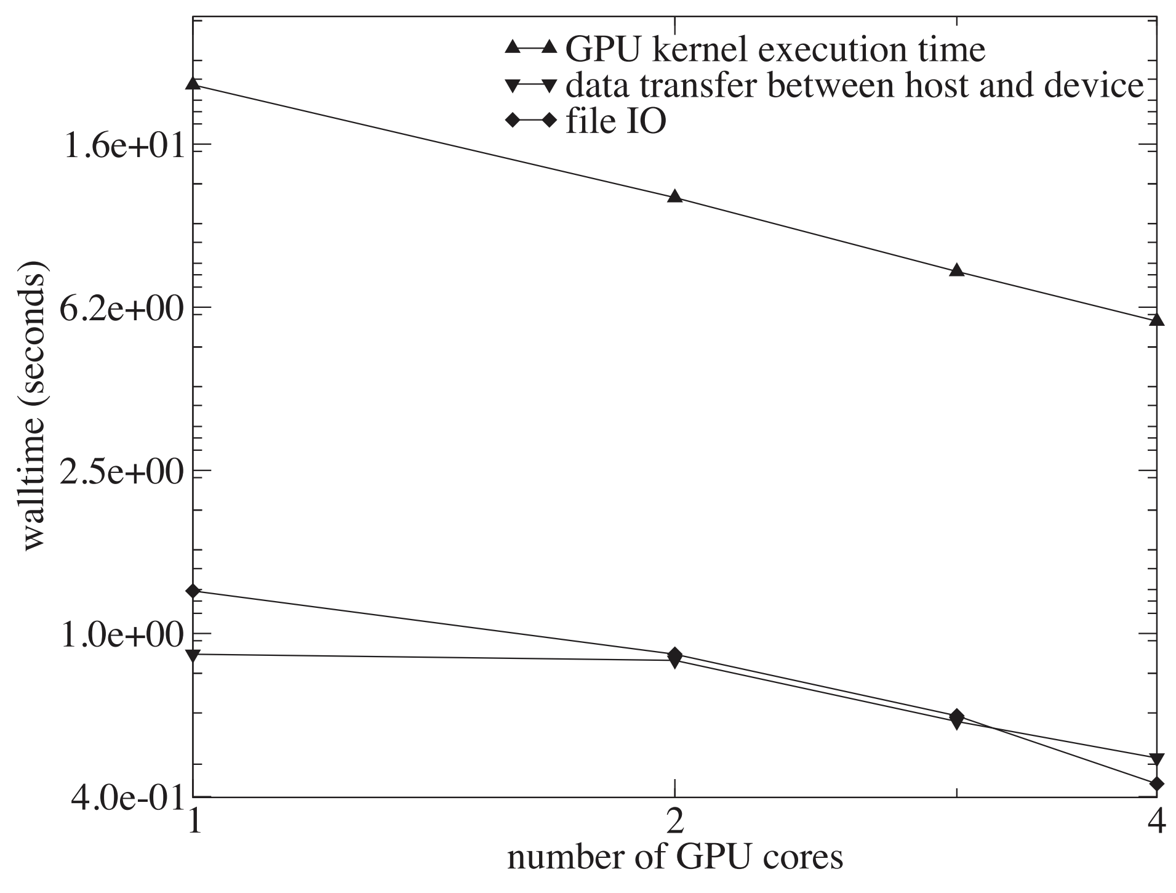

The unfiltered data are shown in gray. In brown a selection of the filtered data are shown. Clearly the arterial structure is enhanced and blurring and halo artifacts suppressed. The enhanced arterial structure is now ready to be analyzed or subjected to further data processing algorithms such as skeletonization or lineness enhancement.

To study coronary arterial tree organization an Imaging Cryomicrotome was developed previously at the AMC. Since the Cryomicrotome has become available, it is possible to study the coronary circulation with increased resolution and across a variety of scales yielding high resolution 3D datasets. Important clinical applications involve the study of the growth of coronary collaterals in the diseased or failing myocardium as well as the perfusion heterogeneity illustrated by the distribution of fluorescent labeled microspheres.

Table 1: Speedup

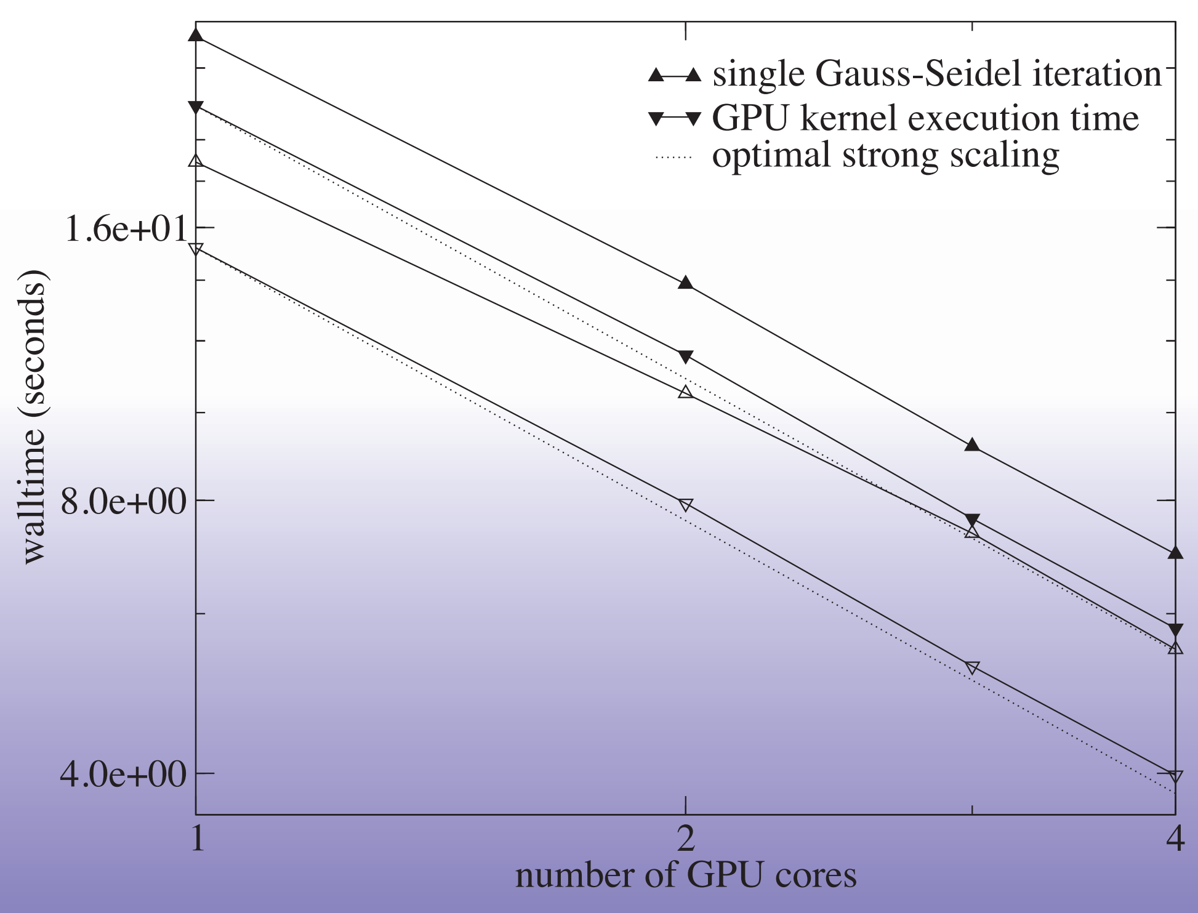
Image size	CPU spatial	CPU spectral	GPU (Tesla M1060)	GPU (gtx 285)	speedup
100^2	0.01	0.14	0.629×10^{-4}		159
800^2	0.52	0.86	0.22×10^{-2}	0.16×10^{-2}	325
2000^2	3.32	4.8	0.13×10^{-1}	0.925×10^{-2}	359
4000^2	13.33		0.578×10^{-1}	0.398×10^{-1}	334

Speedup results for our GPU implementation relative to a CPU based spatial and spectral filter implementation. The CPU experiments were performed on a AMD Phenom II X4 995 CPU at 3.2 GHz. The spatial convolution was implemented in fortran and compiled with gfortran with optimization flags: -O3 -mtune=athlon64-sse3 -fast-math -unroll-all-loops



Relative walltime for GPU kernel execution, data transfer between GPU and motherboard and file IO, for a stack of 100x4000x4000 pixel images.

A parallel iterative (Gauss Seidel) spatial convolution strategy for GPU arrays was developed, using a system specific point spread function (PSF), to enhance the arterial structure. The PSF is small and spatially invariant and resides in fast constant memory of the GPU while the unfiltered data reside in slower global memory but are prefetched by blocks of threads in shared GPU memory. Filtering is achieved by a series of unrolled loops in shared memory. Between iterations the filtered data is stored to disk using asynchronous MPI-IO effectively hiding the IO overhead with the kernel execution time.



Strong scaling relation for a stack of 4000x4000 pixel images. Closed symbols represent experiments on a TeslaM106, open symbols experiments on gtx-285.

Netherlands Organisation for Scientific Research

AUTHORS THOMAS GEENEN | PEPIJN VAN HORSSEN | JOS SPAAN | MARIA SIEBES | JEROEN VAN DEN WIJNGAARD