Online real-time reconstruction of adaptive TSENSE with commodity CPU / GPU hardware

Sébastien Roujol, Baudouin Denis de Senneville, Erkki Vahalla, Thomas Sangild Sorensen, Chrit Moonen, Mario Ries

Online real-time reconstruction of adaptive TSENSE with commodity CPU / GPU hardware

Sébastien Roujol, M.Sc.[1], Baudouin Denis de Senneville, PhD[1], Erkki Vahalla, PhD [2], Thomas Sangild Sørensen, PhD[3], Chrit Moonen, PhD[1], Mario Ries, PhD[1]

[1]: Laboratory for Molecular and Functional Imaging: From Physiology to Therapy, UMR 5231 CNRS/Université Bordeaux 2, 146 rue Léo Saignat, F - 33076 Bordeaux, France.

[2]: Philips Medical Systems MR Finland, Äyritie 4, FI-01510 Vantaa, Finland.

[3]: Department of Computer Science and Institute of Clinical Medicine, University of Aarhus, Denmark.

Address correspondence to:
Mario Ries
UMR 5231, Imagerie Moléculaire et Fonctionnelle
Université « Victor Segalen » Bordeaux 2
146, rue Leo Saignat, case 117
33076 Bordeaux, France
E-Mail: mario.ries@imf.u-bordeaux2.fr
Tel: +33 5 57 57 45 89
Fax: +33 5 57 57 45 97

Acknowledgments: Ligue Nationale Contre le Cancer, Conseil Régional d'Aquitaine, Diagnostic Molecular Imaging EC-FP6-project LSHB-CT-2005-512146 Ministère de la Recherche, Philips Healthcare.

Key words: Real-time reconstruction, TSENSE, Interventional Imaging.

Word count: 3210.

ABSTRACT

Adaptive TSENSE has been suggested as a robust parallel imaging method suitable for MR-guidance of interventional procedures. However, in practice, the reconstruction of adaptive TSENSE images obtained with large coil arrays leads to long reconstruction times and latencies and thus hampers its use for applications such as MR-guided thermotherapy or cardiovascular catheterization.

Here, we demonstrate a real-time reconstruction pipeline for adaptive TSENSE with low image latencies and high frame-rates on affordable commodity PC hardware. For typical image sizes used in interventional imaging (128×96, 16-channels, SENSE-factor 2-4), the pipeline is able to reconstruct adaptive TSENSE images with image latencies below 90 ms at frame-rates of up to 40 images/s, rendering the MR-performance in practice limited by the constraints of the MR-acquisition. Its performance is demonstrated by the online reconstruction of in-vivo MR images for rapid temperature mapping of the kidney and for cardiac cathetirization.

INTRODUCTION

Real-time MRI for therapy control is used to provide retroactive feedback information for interventional applications such as catheter guidance or MRI-controlled tissue ablation. Such interventions require several minutes of continuous imaging and are preferably carried out under free-breathing conditions. This presents the following challenges to the MR-acquisition and reconstruction.

The data acquisition has to be rapid enough to resolve physiological motion such as respiratory or cardiac motion, and the signal changes from the interventional process itself. In addition, the employed data acquisition and reconstruction schemes must not introduce long image latencies, since this degrades the value of the image data for feedback control.

One of the significant developments in this field was the introduction of parallel imaging methods such as SMASH (1), SENSE (2), or GRAPPA (3), which allows, on one hand, to accelerate the data acquisition and, on the other hand, to shorten readout echo-trains, and thus reduces signal dropout and image distortions in regions of large susceptibility variations.

However, parallel imaging methods have also several drawbacks. SENSE, SMASH, and GRAPPA require coil sensitivity data for the image reconstruction, which depends on the coil position. Due to the long duration of interventional imaging procedures, displacements of

the receiver coils are common and lead either to increased image artifacts over time, or, in the cases where the receiver coils are embedded in the interventional device itself, make these approaches altogether unsuitable. Several ideas have been brought forward to overcome this problem:

AUTO-SMASH (4) embeds low-resolution *k*-space calibration data in the acquisition of undersampled high-resolution images. Alternative approaches, such as TSENSE (5) or TGRAPPA (6), collect high-resolution calibration data at a lower temporal resolution than what is used for image encoding.Common to all of these techniques is the fact that the additional computational overhead leads to reconstruction times which often exceed the MR-acquisition time, in particular if large coil arrays are used, and are therefore in practice hard to exploit for real-time therapy guidance.

Guttman et al. demonstrated the feasibility of a real-time reconstruction for adaptive TSENSE on a four channel receiver system with an image latency of ~0.3 s (7). Recently, Hansen et al. showed that a Cartesian SENSE reconstruction can be significantly accelerated - by up to two orders of magnitude - if the massively parallel architecture of commodity graphics hardware (GPU) is used (8).

The presented work combines both approaches to demonstrate the ability to reconstruct adaptive TSENSE data in real-time with low image latencies on affordable commodity hardware. For this, a detailed performance comparison of the TSENSE reconstruction steps for the GPU and CPU implementation is presented and the image latency and throughput of the full reconstruction under different typical interventional work-loads are benchmarked. Finally, the benefit of the proposed online reconstruction for therapy guidance is demonstrated with two in-vivo imaging experiments imitating conditions typical for MR-guided thermotherapy on abdominal regions and cardiovascular catheterization under real-time MRI guidance.

MATERIAL & METHODS

Reconstruction Architecture

A simplistic approach of the data handling and the reconstruction is illustrated in figure 1.a: upon completion of *k*-space sampling of the n^{th} image, the data is transported from the acquisition system to the reconstruction system. There the reconstruction process is started and a new image acquisition can begin. Although this approach has the advantage of a straight forward implementation of the reconstructor pipeline due to its serial nature, it leads to a sub-

stantial image latency. A more effective design to reduce image latency, which is to some extent standard on most commercially available MR-systems, is shown in figure 1.b. Here the acquired data of the n^{th} image is transported to the reconstruction system line by line and the first preparative reconstruction steps are carried out immediately. Upon arrival of the last required *k*-space data, all remaining reconstruction steps are completed. Ideally, the data acquisition of the $n+1^{th}$ image can be started as soon as the n^{th} acquisition cycle is terminated. In practice, this leads to temporal overlap of data transport and image reconstruction. Further reduction of the image latency is achievable by reducing the computational time of the main reconstruction steps (as shown in figure 1.c) with a highly parallel reconstruction system such as a GPU.

The above means that for an efficient implementation, the MR-data have to be reorganized at several points during the reconstruction. Within the scope of this paper, the following reconstruction-related denotations apply: The MR-signal S is referred to as $S(r,p,s,c)$. The order of the indices reflects the ordering of the data in the computer memory, r being the fastest varying component (r : index in read-out direction, p : index in phase direction, s : slice index, c : coil index). Let N_r, N_p, N_{psense}, N_s and N_c define the number of elements in the read-out direction, the number of phase encoding steps, the number of phase encoding steps reduced by SENSE acceleration, the number of slices and the number of coils, respectively.

Reconstruction pipeline

The detailed overview of the implementation of a pipeline is shown in figure 2. The acquisition system streams raw *k*-space data of dynamic image *n* to a communication thread (CPU-thread #1) on the reconstructor, which feeds - over a buffer queue - a second preparatory reconstruction thread. Here *k*-space ordering and Fourier-reconstruction in readout-direction are carried out and the resulting *x-ky-z*-space is buffered to a shared-memory, which is accessible by all further reconstruction processes. The shared-memory is implemented as a round-robin to allow CPU-threads #3 and #4 to complete the reconstruction of dynamic image *n*, while threads #1 and #2 are immediately ready to receive and process new incoming data of the subsequent image *n+1*. The main image reconstruction is carried out by CPU-thread #3 which applies the EPI-phase corrections and Fourier transforms the image data in phase encoding direction.

For non-SENSE encoded images the image reconstruction would be complete at this point. SENSE encoded images require an additional unfolding step. Furthermore, adaptive TSENSE requires a continuous recalculation of the sensitivity maps and subsequently the

SENSE matrix, which in itself is very time consuming. As Hansen et al. have shown, the highly linear nature of the reconstruction steps required for SENSE are well-suited for GPU offloading (8). The presented architecture takes advantage of the linearity by offloading all adaptive TSENSE calculations to a GPU. For this, the CPU-thread #4 sends a copy of the undersampled *k*-space to the GPU, which then partially updates a fully sampled *k*-space copy, as described by Kellman et al. (5). Subsequently, the GPU Fourier-transforms this data, calculates the corresponding sensitivity maps and recalculates the SENSE matrix.

Simultaneously, the CPU-thread #3 finalizes the Fourier-reconstruction of the folded image and sends the result to the GPU for the final SENSE reconstruction step.

Finally, an optional temporal filter used for residual artifact suppression can be applied. Upon completion of the reconstruction, the image is taken over by a CPU-thread #5 which dispatches the images to one (or several) viewing station(s).

GPU implementation

Since a detailed description of the GPU architecture can be found elsewhere (8) and the potential of GPU-hardware for SENSE reconstruction has been evaluated by Hansen et al. (9), the description of GPU implementation is limited in the scope of this paper to the minimum required to understand the presented TSENSE implementation.

The employed GPU can be seen as a massively parallel coprocessor which contains 128 basic arithmetic processing units, which are, in turn, organized into teams of eight multiprocessors containing fast shared on-chip memory and a slower global memory shared across multiprocessors. The CUDA programming API (8) of the GPU reflects this architecture in the following way: executable code is broken down into a sequentially executed set of kernels, which are organized into a grid of N_{block} blocks. Each block contains a set of N_{thread} threads executed in parallel by one multiprocessor.

Data is exchanged between the dynamic random access memory (DRAM) of the GPU and the RAM of the hosting computer via direct memory access (DMA) of the PCI-express bus, which is bandwidth limited to 3-4 GB/s. In addition the GPU DRAM memory access has a high latency (400-600 clock cycles). Therefore, in order to obtain a maximal processing performance on this particular GPU architecture, the TSENSE method was implemented according to the following guidelines:

- Excessive exchange of data between the RAM of the hosting computer and the GPU has to be avoided.

- The large memory latency of the global GPU DRAM can be hidden by designing the kernels to solve the mathematical problems with a high density of arithmetic operations per memory access and designing the execution configuration to exploit the dimensionality of the given problem to maximize N_{block} and N_{thread}.
- Memory intensive processing has to be implemented so that cooperation takes place between threads within the same block and not between different blocks.
- Multi-dimensional data in the DRAM should be organized so that memory access from the individual threads results in contiguous memory access by the memory controller.

Fourier Reconstruction in phase encoding direction

Since for the most of the GPU reconstruction steps $N_{block}=N_r$, and $N_{thread}=N_p$, the data is first reordered from data organisation $S(r,p,c,s)$ to $S(p,r,c,s)$ to allow coalesced memory access by the threads ($N_{thread}=16\times16$ in our implementation, $N_{block}=N_rN_pN_cN_s/N_{thread}$). This step also prepares the Fast Fourier Transformations (FFTs) in phase encoding direction by applying the required frequency shifts. Subsequently, the CUDA programming API (8) provided FFT is applied in the p-dimension (using $N_{thread}=N_p$, $N_{block}=N_rN_cN_s$,), resulting in an unaliased image for each receiver coil and slice.

Sensivity Map Update

First, the calculation of the synthetic reference image is performed. This can be, for the case of TSENSE, provided by the Sum-of-Square of the coil images (2) ($N_{thread}=N_p$, $N_{block}=N_rN_s$). Subsequently, the coil sensitivities are calculated ($N_{thread}=N_p$, $N_{block}=N_rN_s$) and optionally, a receiver coil noise decorrelation step (10) is performed. For the decorrelation step, the first N_c threads load each one element of the pre-calculated noise-decorrelation matrix $\psi_{noise}^{-1}\otimes I_d$. After a mutual barrier point each thread loads the reference image value for the position associated with the thread and calculates the spatial coil sensitivity map S for all coils. The noise decorrelated sensitivity maps S_{decorr} are obtained as follows:

$$S_{decorr}=(\psi_{noise}^{-1}\otimes I_d)S \qquad [1]$$

The result is stored in $S(c,p,r,s)$ order to prepare the SENSE-matrix recalculation.

Temporal Filtering

For residual artifact suppression, all sensitivity maps are temporally filtered with an infinite impulse response filter (IIR) ($N_{thread}=N_p$, $N_{block}=N_rN_cN_s$) using a SENSE factor dependent bandwidth of *1/SENSE_factor* as described in detail by Maclair et al. (11).

SENSE-matrix Recalculation

The unfolding matrix U is calculated for each slice position (using $N_{thread}=N_r$, $N_{block}=N_pN_s/SENSE\_factor$) as follows:

$$U=(S_{decorr}{}^{H}S_{decorr})^{-1}S_{decorr}{}^{H} \quad [2]$$

Each thread calculates the matrix inversion for one pixel using an adapted version of the LU-decomposition algorithm (12). Alternatively, for larger coil arrays, a Cholesky (12) decomposition can be used.

SENSE-Reconstruction

This step (performed using $N_{thread}=N_{psense}$, $N_{block}=N_rN_s$) begins optionally with a noise-decorrelation of folded image. For this, the same process as the one used in sensitivity map noise-decorrelation (see equation [1]) is applied. Then, the unfolding process is performed as follows:

$$I_{unaliased}=U.I_{aliased} \quad [3]$$

where $I_{unaliased}$ and $I_{aliased}$ are the unaliased and the aliased image, respectively. Note that superimposed voxel indices are precalculated in both data organisations $S(r,p,c,s)$ and $S(p,r,c,s)$ to speed up this step. In our implementation, $I_{aliased}$ is computed on the CPU in parallel with the SENSE-matrix recalculation step explained above.

MRI

All in-vivo experiments were performed on a Philips Achieva 3.0 T X-series scanner equipped with a Philips 16-channel torso coil, using a gradient-echo EPI (GE-EPI) sequence. The following acquisition parameters were used:

Abdominal thermometry imaging:

A single slice placed on the right kidney of a healthy volunteer was acquired using an echo-train of 67 echoes (33 for two-fold and 17 for four-fold acceleration, respectively), a fixed TE of 15 ms and TR of 50 ms for all acceleration factors, a flip angle of 35 º, a bandwidth in phase-encoding direction BW_{pe} of 4.1 kHz (1.5 kHz for two-fold and 0.7 kHz for four-fold acceleration), and a resolution of 3.13×3.13×5 mm^3. Fat suppression with spectral pre-saturation with inversion recovery (SPIR) was used.

Cardiac imaging:

A single slice placed on the right ventricular outflow tract and the aorta of a healthy volunteer was imaged using an echo-train of 63 echoes (31 for SENSE 2; 15 for SENSE 4), a *TR* of 50 ms, a *TE* of 15 ms (7.8 ms for SENSE 2, 4.5 ms for SENSE 4), a flip angle of 35 º, a BW_{pe}=3.9 kHz and an image resolution of 3.0×3.0×5 mm^3.

Reconstructor Hardware

The reconstructor was a dual processor (INTEL 3.1 GHz Penryn, four cores) workstation with 8 GB of RAM and dual 1 Gb/s network interface cards. The GPU was a NVIDIA 8800GTX card with 768 MB of DRAM connected over a PCIe x16 link.

Reconstructor Software

The reconstructor was running under Linux (SMP-Kernel 2.6.18) which was stripped from all non-essential system processes. For the data transport from the MR-acquisition system to the reconstructor and from the reconstructor to the viewing station(s) a real-time implementation of the common object request broker architecture (CORBA) known as The Ace Orb (TAO) (13,14) was used. For Fourier reconstruction and all linear algebra operations on the CPU, the ACML math package (15) was employed. The package contains linear algebra packages BLAS (16) and LAPACK (17). The GPU implementation of the adaptive TSENSE method was realized using CUDA (8), with in-house developed linear algebra routines based on methods presented in Numerical Recipes in C++ (18).

RESULTS

Detailed performance comparison between the CPU only and the CPU/GPU implementation

Table 1 compares the computation time for each TSENSE reconstruction step for a two-fold accelerated image (resolution 128×128, six receiver channels) between the implementation using the CPU only and the CPU/GPU implementation. The two most time consuming tasks, the temporal filtering of sensitivity maps and the SENSE matrix recalculation, were accelerated by a factor of 14.7 and 16.3, respectively and lead to an overall eight-fold increase of the reconstruction speed despite the additional data transfer to the GPU.

Figure 3 compares the computation time measured for various TSENSE acceleration factors (2, 3 and 4) and different image resolutions (128×128, 256×256 and 512×512 pixels)

using the CPU only (3.a) and using the CPU/GPU approach (3.b).

Real-time benchmarking of throughput and latency

Table 2 reports the data transfer time, the reconstruction time and the resulting image latency (as defined in figure 1) under real-time conditions for coil setups with 4, 6, 8, and 16 elements, and TSENSE acceleration factors of 2, 3, and 4 (resolution 128×128, two-fold read-out oversampling). Since computation times for multi-slice acquisitions are almost linear with the slice number, only results measured with a single slice are reported.

Data transport varies between 17 ms (four-fold accelerated, four channels, ~135 kB per image) to 76 ms (two-fold accelerated, 16 channels, ~1 MB per image) depending on the data size. The reconstruction itself has a theoretical peak performance between 75 images/s (SENSE 4, 16-channels) to 330 images/s (SENSE 2, 4-channels). However, in practice the achievable peak data-throughput was found to be limited by the I/O subsystem of the acquisition system to ~2.1 MB/s, which corresponds to an image frame-rate of 20 images/s for a two-fold TSENSE accelerated data set (128×128, two-fold read-out oversampling, 16 receiver channels) with a overall image latency of 90 ms, or 40 images/s and an image latency of 60 ms for a four-fold acceleration.

In-vivo imaging

Figures 4 shows a fully sampled (a), two-fold accelerated (b) and four-fold accelerated (c) abdominal image of the right kidney of a healthy volunteer obtained under free-breathing conditions. An imaging frame-rate of 20 images/s was maintained online for 300 seconds of MR-imaging. The thorax and the lower abdomen show residual folding artifacts due to limited coil coverage. The center of the image depicting the kidney was artifact-free over the entire imaging duration and is hence suitable for continuous MR-thermometry.

Similarly, figures 4.d, 4.e, and 4.f show a cross-section through the right ventricular outflow tract and the aorta as it would typically be used for catheterization guidance, obtained with 20 images/s. Besides the residual folding artifacts in areas of limited coil coverage, the target area was found to be artifact-free during the entire experiment.

DISCUSSION

The usefulness of a reconstruction pipeline for real-time MR-guidance of

interventional procedures can be characterized by the reconstruction speed and the achievable image latency. Fundamentally, the reconstruction time of the pipeline has to be comparable to, or preferably shorter as, the acquisition time of the MR-system, in order to maintain the real-time condition for sustained imaging. Table 2 shows that a purely CPU-based reconstruction limits MR-acquisition to only 4-6 coils elements and SENSE acceleration factor to 2-3. Since the proposed CPU/GPU implementation of the TSENSE reconstruction achieves in average an eight-fold acceleration, these limits are virtually removed. In practice, this means that imaging speed is now limited by the boundaries imposed by the MR-sequence, the specific absorption rate, the desired signal-to-noise ratio and the I/O-bandwidth of the acquisition system.

For the case of MR-guided surgical interventions, large image latencies introduce a significant lag between the action and its observable consequence and thus limit the accuracy and speed of the procedure. The proposed CPU/GPU reconstruction achieves image latencies from 20 to 90 ms for all coil configurations and acceleration factors, and is thus well-suited for manual feedback.

For the case of MR-image guidance of semi- or fully-automated interventions, such as high intensity focused ultrasound (HIFU) ablations, short latencies are of even greater importance. In this type of application, the position and target temperature are ascertained with the help of MR-imaging/thermometry and further used to retroactively adjust the beam position and power-levels of a HIFU device. De Senneville et al. (19) demonstrated that large image latencies have to be compensated with help of extensive modeling which anticipates displacement and temperature evolution; and that the robustness of such algorithms greatly increases with short image latencies.

The presented work shows that a reconstruction pipeline for real-time reconstructed adaptive TSENSE imaging, which is suitable for interventional MR-guidance, is realizable on affordable commodity hardware.

REFERENCES

(1) Sodickson D. K., Manning W. J., Simultaneous acquisition of spatial harmonics (SMASH): Fast imaging with radiofrequency coil arrays, Magnetic Resonance in Medicine, 1997;38:591-603.

(2) Pruessmann K. P, Weiger M., Scheidegger M. B., Boesiger P., SENSE: Sensitivity encoding for fast MRI, Magnetic Resonance in Medicine, 1999; 42:952-962.

(3) Griswold M. A., Jakob P. M., Heidemann R. M., Nittka M., Jellus V., Wang J., Kiefer B., Haase A., Generalized autocalibrating partially parallel acquisitions (GRAPPA). Magnetic Resonance in Medicine, 2002;47:1202–1210.

(4) Jakob P. M., Griswold M. A., Edelman R. R., Sodickson D. K., AUTO-SMASH: a selfcalibrating technique for SMASH imaging: Simultaneous Acquisition of Spatial Harmonics. MAGMA, 1998;7:42–54.

(5) Kellman P., Epstein F. H., McVeigh E. R., Adaptive Sensitivity Encoding Incorporating Temporal Filtering (TSENSE), Magnetic Resonance in Medicine 2001; 45:846–852.

(6) Breuer F. A., Kellman P., Griswold M. A., Jakob P. M., Dynamic autocalibrated parallel imaging using temporal GRAPPA (TGRAPPA), Magnetic Resonance in Medicine, 2005;53(4): 981–985.

(7) Guttman M. A., Kellman P., Dick A. J., Lederman R. J., McVeigh E. R., Real-Time Accelerated Interactive MRI With Adaptive TSENSE and UNFOLD, Magnetic Resonance in Medicine, 2003;50(2):315-321.

(8) NVIDIA Corporation, NVIDIA CUDA: Compute Unified Device Architecture – Programming Guide, 2.0 ed 2008, pp. 1-107.

(9) Hansen M. S., Atkinson D., Sorensen T. S., Cartesian SENSE and *k*-t SENSE Reconstruction using Commodity Graphics Hardware, Magnetic Resonance in Medicine, 2008;59(3):463-8.

(10) Pruessmann K. P., Weiger M., Börnert P., Boesiger P., Advances in Sensitivity Encoding With Arbitrary *k*-Space Trajectories, Magnetic Resonance in Medicine 2001;46:638–651.

(11) Maclair G., Ries M., Moonen C.T.W., Low latency vs. non linear phase: A balanced approach for UNFOLD filter design in quantitative Real-Time MRI, Proceedings of International Society of Magnetic Resonance in Medicine, Seattle, USA, 2006.

(12) Press W., Teukolsky S., Vetterling W., Flannery B., Numerical Recipes in C, Cambridge University Press.

(13) Schmidt D. C. , Levine D. L., and Mungee S., "The Design and Performance of Real-Time Object Request Brokers," Computer Communications, vol. 21, pp. 294–324, Apr. 1998.

(14) Pyrali I, Schmidt DC, Cytron R. Techniques for enhancing real-time CORBA quality of service. In: Proceedings of IEEE, RTSS 2003; Cancun, Mexico, vol. 91, 1070-1085.

(15) Numerical Algorithms Group Ltd. AMD Core Math Library User Guide, 2006.

(16) Dongarra J.J., DuCroz J., Duff I., Hammarling S., Algorithm 679: A Set of Level 3 Basic Linear Algebra Subprograms: Model Implementation and Test Programs. ACM Trans. Math. Software, 16(1):18-28, March 1990.

(17) Anderson E., Bai Z., Bischof C., Demmel J., Dongarra J., DuCroz J., Greenbaum A., Hammarling S., McKenny A., Ostrrouchov S., Sorensen D., LAPACK Users Guide. SIAM Publications, 1992, ISBN 0-89871-294-7.

(18) Press W. H., Teukolsky S. T., Vetterling W. A., Flannery B. P., Numerical recipes in C++: The art of scientific computing, Cambridge University Press, 2nd edition, 2002.

(19) Denis de Senneville B., Mougenot C., Moonen C. T. W., Real-Time Adaptive Methods for Treatment of Mobile Organs by MRI-Controlled High-Intensity Focused Ultrasound, Magnetic Resonance in Medicine, 2007;57:319–330.

TABLE CAPTIONS

Table 1. Computation time for all TSENSE reconstruction steps for a single slice of resolution 128×128 pixels with an acceleration factor of 2 using six receiver coils.

Table 2. Reconstruction time and image latency for a single slice reconstruction of resolution 128×128 with TSENSE factor 2, 3, 4 for 4, 6, 8, 16 coil channels. With the proposed CPU/GPU implementation, total computation times are far below the TR on all tests demonstrating that real-time reconstruction is feasible.

FIGURE CAPTIONS

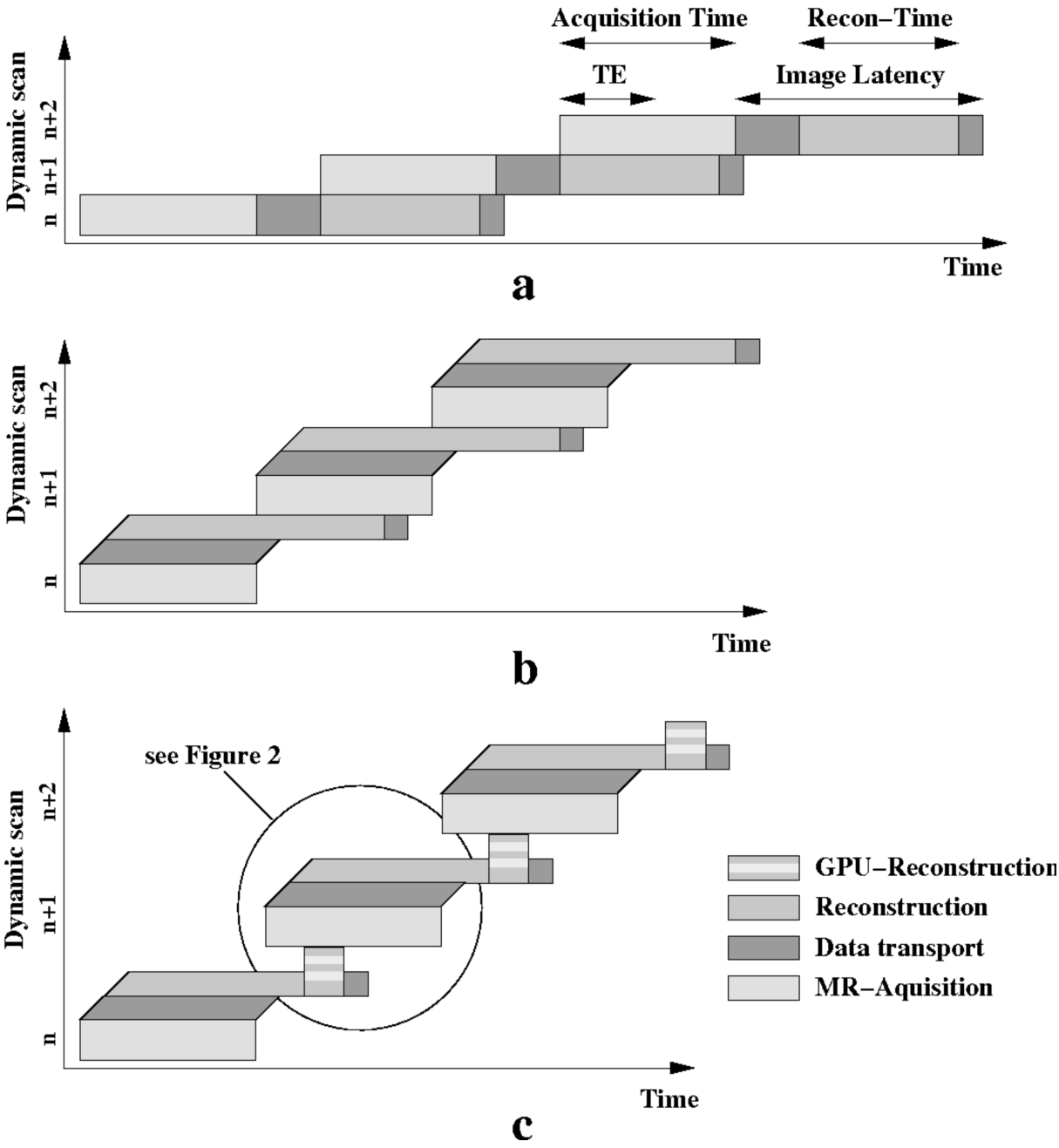

Figure 1. Overview of the principal optimization steps of the reconstruction pipeline in order to achieve short image latencies. The timing diagram on the top shows a simple design which has a completely serialized data handling and reconstruction and thus only allows the reconstruction and the data acquisition to overlap in time. In the middle an optimized design is shown, which transports data off the acquisition system to the reconstructor immediately, where the first reconstruction steps are carried out upon data arrival. At the bottom the final design is shown, which employs in addition a highly parallelized main reconstruction step based on GPU hardware to shorten the reconstruction time.

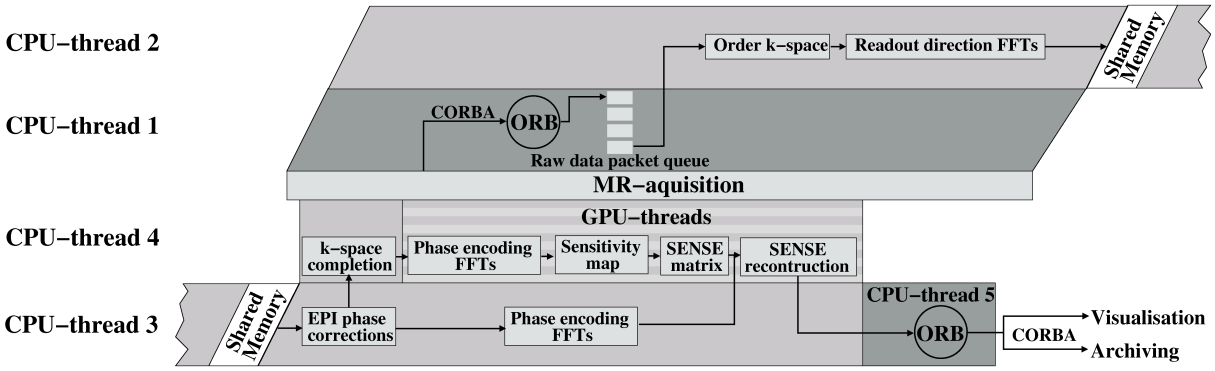

Figure 2. Overview of the thread architecture of the reconstruction pipeline.

In order to achieve high throughput and short image latencies, as many independent data handling/reconstruction steps as possible are carried out in parallel: CPU-threads one and two handle/reconstruct *k*-space data from the dynamic image *n+1*, while CPU-threads three and four finalize simultaneously the reconstruction of dynamic image *n*. The time-consuming processing steps for the adaptive TSENSE reconstruction are offloaded to GPU-hardware which in itself uses up to 128 threads in parallel. A separate thread for dispatching the data to a visualization and an archiving system is used.

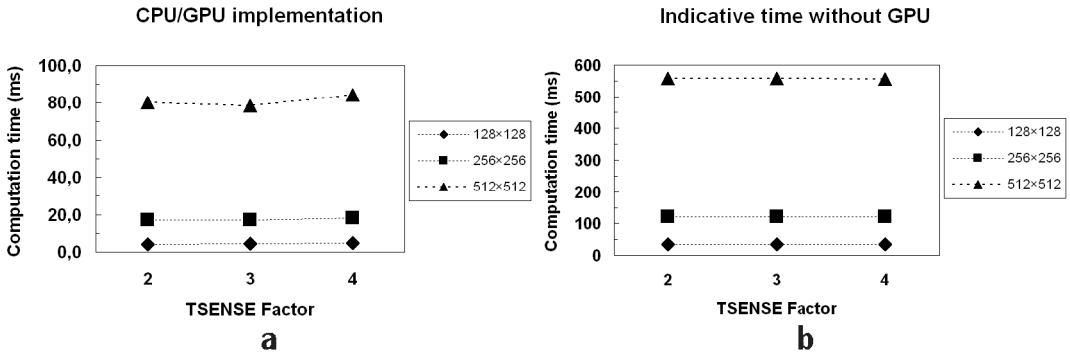

Figure 3. Computation time measured for various TSENSE acceleration factors (2, 3 and 4) and different image resolutions (128×128, 256×256 and 512×512 pixels) using the CPU only **(a)** and using the CPU/GPU approach **(b)**. Acceleration factors around 7-8 were measured with the proposed CPU/GPU approach for each tested resolution. Computation times remain identical using the proposed CPU/GPU approach for each tested TSENSE acceleration factor.

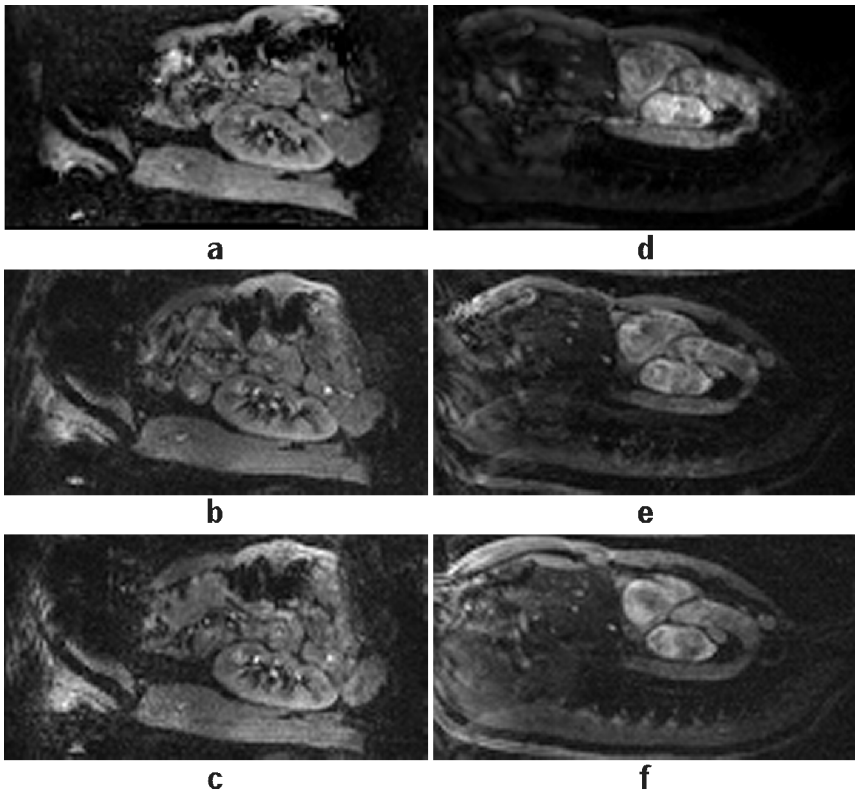

Figure 4. Real-time reconstructed MR-images using a combined CPU/GPU reconstruction (left-right: foot-head, up-down: anterior-posterior direction):

(a-c) : Fully sampled abdominal image centered on the kidney **(a)** and TSENSE accelerated by factor of 2 **(b)** and 4 **(c)**.

(d-f) : Right ventricular outflow tract and the aorta, with full sampling **(d)** and with a TSENSE acceleration factor of 2 **(e)** and 4 **(f)**.