

Implementation of an MRI Reconstruction Framework with a System-Integrated Acquisition Database

Philip J Beatty¹, Chad T Harris¹, Curtis N Weins¹, David Kennedy¹, Andrew T Curtis¹, and Jeff A Stainsby¹

¹Research and Development, Synaptive Medical, Toronto, ON, Canada

Synopsis

A reconstruction framework is introduced that adds an acquisition database between the system receive chain and the reconstruction processors. This additional system service opens up a number of dataflow possibilities. The framework is demonstrated on a mid-field superconducting head-only MRI system by simultaneously running three reconstruction programs on the system reconstruction platform and comparing the resulting images.

Introduction

We have developed a reconstruction framework that adds an acquisition database in between the system receive chain and the reconstruction processors (Fig. 1). The addition of this database enables reconstruction methods to *fetch data in the order and timeframe that is most convenient for them*, while still enabling reconstruction processing to happen concomitant with data acquisition.

This architecture opens up a number of dataflow possibilities, a few of which are illustrated in Fig. 2. The decision to export acquired scan data to a file can be made after the scan has completed and the images have been viewed. Additional data consumption tools can be added to the system, for example to support further export formats such as the ISMRMRD format¹. Inclusion of an acquisition database makes it straightforward to run multiple client reconstruction programs that consume data from the same scan, and client programs that require only a subset of the acquired data can limit data transfer to their needs.

Methods

The proposed MRI reconstruction architecture was developed and deployed onto our mid-field (0.5T) superconducting head-only system. The acquisition database was implemented as a standalone service that receives “raw” data from multiple receiver channels at a rate of 781 kHz and filters/downsamples (decimates) the data to a scan-specified bandwidth. Both the raw and decimated data are stored and available for retrieval. All acquisitions are tagged with unique identifiers, abstracting out any scan-specific structure; a tailored TCP/IP communication protocol is used to accept asynchronous requests for acquisition data by multiple clients.

In this work, we explore the use of our system by performing an EPI acquisition and connecting three different reconstruction programs (Fig. 3): 1) A standard Cartesian 2D reconstruction with no bipolar EPI calibration; 2) an EPI reconstruction program with bipolar EPI calibration²; and 3) An EPI reconstruction program that processes even and odd lines separately using parallel imaging^{3,4}.

The three reconstruction programs were attached to a spin-echo EPI scan application and an in-house built phantom was imaged. Imaging was performed using an echo train=160, acquisition matrix=160x160, bandwidth=250 kHz, number of slices=10, slice width=5mm, field-of-view=24 cm, TE/TR=138 ms/3s, number of averages=4. The progress of the reconstruction programs was monitored on a system status monitor UI service and the reconstructed images were compared after being converted to DICOM and inserted into the scanner image database.

Results and Discussion

Figure 4 shows screen captures of a status monitor UI service both during and after the scan acquisition, demonstrating the ability of the system to run and monitor multiple reconstruction programs in parallel and for the reconstruction programs to simultaneously fetch data from the acquisition database. Selected imaging results are shown in Fig. 5. As expected, the two reconstruction programs that use methods designed to reconstruct bipolar EPI acquisitions effectively diminished N/2 ghost and image shift issues. The reconstruction framework provided a straightforward and natural way to compare candidate reconstruction methods during a scan session.

In this work we focused on use cases centered around scan and reconstruction development and evaluation. However, this architecture could present advantages beyond the cases considered. For example, it has previously been proposed to use raw (less decimated) data during the beginning of acquisition windows to improve image quality in ultra-short T2/T2* imaging⁵. The proposed framework could be used to keep data transfer manageable in this application by implementing a reconstruction program that requests only the needed portions of the raw (non-decimated) and decimated data. A great deal of flexibility is achieved by allowing data flow to be tailored by reconstruction programs, which can be modified and made application specific more easily than the receive chain.

We have implemented and demonstrated a reconstruction framework that includes a system-integrated acquisition database as an intermediary between the receive chain and reconstruction processors. The proposed architecture supports a number of useful dataflows that can be tailored by application and objective.

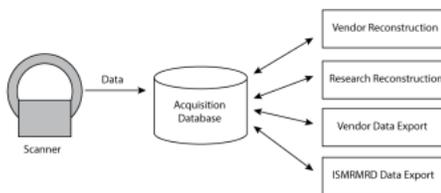
Acknowledgements

No acknowledgement found.

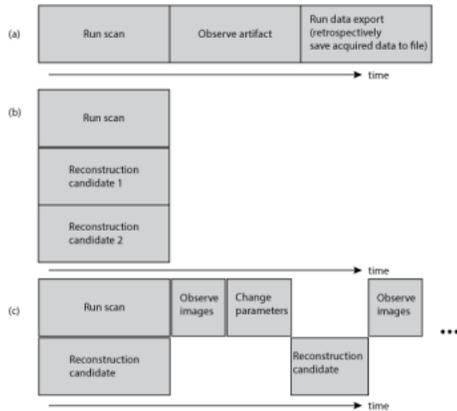
References

1. Inati et al., 2017 MRM 77:411-421.
2. Ahn and Cho, 1987, IEEE TMI, 6:32-36.
3. Kuhara et al. ISMRM 2000, p154.
4. Kellman and McVeigh, 2001 MRM 46:335-343.
5. Marjanovic et al., ISMRM 2014, p927.

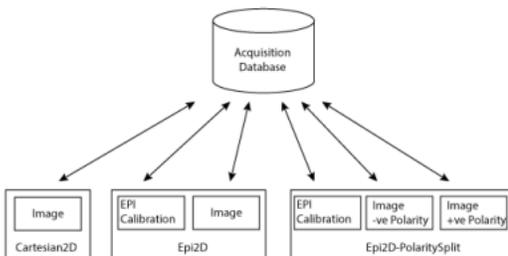
Figures



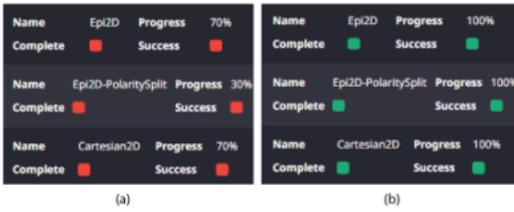
Acquired data is streamed to the acquisition database as it is produced. Vendor and research reconstructions, as well as other tools that consume data (such as data export tools) are all treated as peer clients that interact with the database buffer through a common API. Database status can be observed, and data can be requested as needed by each client. Data is then delivered in an asynchronous manner when available.



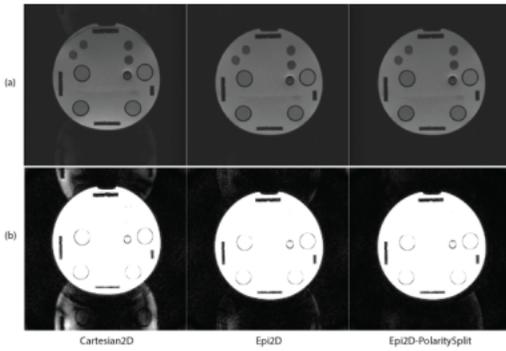
Example use cases. (a) After observing an artifact, the user can retrospectively decide to export the scan data to file for further offline study. This is very useful in cases where the observed artifact is hard to reproduce. (b) Multiple reconstruction programs can run concomitant with the scan acquisition. This can be useful in comparing candidate reconstruction methods. (c) After observing the resultant images from an initial reconstruction, the user can change parameters that alter the behavior of the reconstruction and then re-run the reconstruction on the same data. This enables iterative reconstruction parameter tuning on the MR system without having to reacquire data.



Our experimental setup. Three reconstruction programs were run on the “online” reconstruction system: Cartesian2D, Epi2D and Epi2D-PolaritySplit. Internally, Epi2D and Epi2D-PolaritySplit are architected to have multiple parallel tasks that request data from the acquisition database. For example, in the polarity split program, the task reconstructing the negative polarity requests only the negative polarity readouts.



All three reconstruction programs were attached as full-fledged "online" reconstruction programs, with the ability to process readouts during scan acquisition and interact with the system micro-services (for example to publish status information). (a) Screen captures of a status monitor taken during scan acquisition showing reconstruction programs consuming data at different rates. (b) Screenshot of the status monitor taken shortly after scan acquisition finished showing successful completion of all three reconstruction programs.



Imaging results shown for one slice. (a) Window/Level to view images (b) Window/Level values reduced 10X compared to (a) to view N/2 ghosts. As expected, Cartesian2D (which does not have bipolar EPI calibration) exhibits a shift in the phase encode direction and an N/2 ghost that is visible even in the image in row (a). Both Epi2D and Epi2D-PolaritySplit ameliorate the shift and N/2 ghost issues. Epi2D-PolaritySplit results in a slightly reduced N/2 ghost compared to Epi2D, as illustrated in row (b).