Published in final edited form as:
Magn Reson Med. 2013 February ; 69(2): 571–582. doi:10.1002/mrm.24267.

Coil Compression for Accelerated Imaging with Cartesian Sampling

Tao Zhang[1], **John M. Pauly**[1], **Shreyas S. Vasanawala**[2], and **Michael Lustig**[3]

[1]Magnetic Resonance Systems Research Laboratory, Department of Electrical Engineering, Stanford University, Stanford, California.

[2]Department of Radiology, Stanford University, Stanford, California.

[3]Department of Electrical Engineering and Computer Science, University of California -Berkeley, Berkeley, California.

Abstract

MRI using receiver arrays with many coil elements can provide high signal-to-noise ratio and increase parallel imaging acceleration. At the same time, the growing number of elements results in larger datasets and more computation in the reconstruction. This is of particular concern in 3D acquisitions and in iterative reconstructions. Coil compression algorithms are effective in mitigating this problem by compressing data from many channels into fewer virtual coils. In Cartesian sampling there often are fully sampled *k*-space dimensions. In this work, a new coil compression technique for Cartesian sampling is presented that exploits the spatially varying coil sensitivities in these non-subsampled dimensions for better compression and computation reduction. Instead of directly compressing in *k*-space, coil compression is performed separately for each spatial location along the fully-sampled directions, followed by an additional alignment process that guarantees the smoothness of the virtual coil sensitivities. This important step provides compatibility with autocalibrating parallel imaging techniques. Its performance is not susceptible to artifacts caused by a tight imaging fieldof-view. High quality compression of *in-vivo* 3D data from a 32 channel pediatric coil into 6 virtual coils is demonstrated.

Keywords

Coil Compression; Parallel Imaging; Compressed Sensing

Introduction

With the development of parallel imaging (PI) (1–3), MRI data acquisitions have been significantly accelerated. This has made many lengthy MR exams clinically feasible. PI uses receiver arrays with multiple coils (4) to simultaneously acquire data. The different coil sensitivities of the receiver coils are used to partially replace traditional *k*-space encoding and reduce scan time. In addition to PI, there is a growing trend towards utilizing constrained reconstructions in MRI, such as compressed sensing (CS) (5). These techniques use prior information to achieve better image quality, enhanced resolution, and improved signal-to-noise ratio (SNR). Higher accelerations in data acquisition can be achieved by combining CS with PI (6–8). These reconstructions are often iterative and therefore much more computationally intensive.

Address correspondence to: Tao Zhang Packard Electrical Engineering, Room 208 350 Serra Mall Stanford, CA 94305-9510 TEL: (650) 723-3810 tzhang08@stanford.edu.

Because of the obvious benefits of large coil arrays for SNR and acceleration, both researchers and manufacturers have been developing and building large dedicated arrays (9–12). These large coil arrays enable even faster data acquisition with high SNR. However, the reconstruction and data storage of these large datasets are becoming increasingly problematic, especially for 3D and dynamic acquisitions. To reduce the computation cost with large receiver arrays, coil compression (13–15) can be used. Coil compression linearly combines the raw data from multiple coils into fewer virtual coils. Since the computation required for PI reconstruction and constrained reconstruction is highly dependent on the number of coils, reconstruction time can be significantly reduced by coil compression.

Coil compression was first applied in hardware (15, 16). The original multiple coils are linearly combined into fewer eigencoils using the knowledge of the noise covariance of the receiver array. However, since this hardware combiner does not take into consideration the spatially varying coil sensitivities or the received data, the compression is often not optimal. This leads to large compression signal loss.

Software coil compression algorithms are more flexible and provide higher accuracy. Several techniques have been developed so far. The method by Buehrer *et al.* requires the explicit knowledge of the coil sensitivities (13). Compression is achieved by optimizing the SNR in a region of interest (ROI) in the reconstructed image. This type of coil compression is suitable for sensitivity-based PI reconstructions like SENSE (1). Doneva *et al.* presented a different approach based on automatic selection of the “best” subset of coils (17). The individual coils are selected by a ranking that quantifies the contribution of each coil element to the reconstructed image. This type of coil compression requires the coil sensitivity measurements as well. Similar coil subset selection was applied to channel-by-channel parallel reconstruction by Feng *et al.* (18). Huang *et al.* proposed a data-based coil compression that does not require the coil sensitivity measurements and is solely based on the acquired data (14). This method is fast, and it works well with autocalibrating PI (ACPI) and CS. Channel-by-channel parallel reconstruction can also be speeded up by reconstructing a single direct virtual coil (19) or a synthetic target coil (20). A generalized hybrid method of compressing both the source channels and target channels has been recently proposed by Huang *et al.* (21).

All these software coil compression methods have been well demonstrated for 2D Cartesian data acquisitions. Coil compression for non-Cartesian BLADE acquisitions has been studied by Stemmer *et al.* (22). For 3D datasets, the direct extension of these coil compression methods is usually applied (14). The original coils are combined by the same linear coil compression weights for the entire 3D dataset. Since the coil sensitivities vary spatially in 3D, this direct extension needs more virtual coils to achieve a negligible compression loss. Huang *et al.* studied a spatially varying coil compression for 3D datasets with one phase encoding direction fully sampled (23), but no major advantages were reported.

In this work, a data-based coil compression method for accelerated Cartesian acquisitions is introduced. It is based on a geometric decomposition of the data. In Cartesian acquisitions, there are often one or more dimensions that are fully sampled (*e.g.,* the readout direction), and the variation of the coil sensitivities along these dimensions is not utilized to accelerate data acquisition. The proposed method minimizes the number of virtual coils (hence the reconstruction time) using a spatially varying coil compression. Coil compression is performed separately for each location along the fully sampled dimensions by a singular value decomposition (SVD). Then the spatially varying compression matrices are carefully aligned so that the virtual coils have smooth coil sensitivities. The smoothness of the coil sensitivities is crucial for the calibration step in ACPI methods, where the unacquired data are synthesized from data at multiple adjacent locations in the readout direction (24). In addition, because the coil compression is performed in *k*-space, this approach is not susceptible to tight imaging field-of-view (FOV) artifacts (25). Using *in-vivo* data acquired with a 32 channel dedicated pediatric coil array, we demonstrate near lossless geometric-decomposition coil compression (GCC) from 32 original coils to 6 virtual coils. The ACPI and CS reconstruction of data in the 6 virtual coils requires approximately 14 times less computation than that of the original data, without image quality degradation.

Theory

Data Redundancy and Coil Compression

In PI, the different coil sensitivities of the receiver array coils are used to accelerate data acquisition by subsampling *k*-space and reconstruct the missing data (1–3). Square root of sum-of-square (SSOS) images are usually used to combine individual coil images (4). Assume N is the number of the original coils used for data acquisition. At each location $k = [k_x, k_y, k_z]^T$ in *k*-space, define a vector $v^o(k) = \left[v_1^o(k), v_2^o(k), \cdots, v_N^o(k)\right]^T$ that represents data at this location from all the original coils. The acquired raw data $v^o(k)$ is usually contaminated with noise that can be modeled as a zero mean Gaussian random noise with covariance matrix Ψ. Define a new set of coils with data vectors $v(k) = [v_1(k), v_2(k), ..., v_N(k)]^T$:

$$v(k) = \Psi^{-1/2} v^o(k), \quad (1)$$

Then the noise in this set of coils is independently identically distributed (IID) (15). For simplicity, this set of coils ($v(k)$) is considered as the original set of coils for the rest of this paper.

In most cases, the different coil sensitivity functions do not form an orthogonal basis and often span a subspace with dimension much smaller than the number of coils. This limits the possible acceleration factor of PI. At the same time, this results in large amount of redundant and highly correlated multicoil data. Coil compression algorithms reduce this data redundancy by combining the original multicoil data into fewer virtual coils. The ultimate goal of coil compression is to minimize the number of virtual coils while preserving as much of the signal as possible. Let $A \in \mathscr{C}^{M\times N}$ represent a coil compression matrix and the data vector $v'(k) = \left[v_1'(k), v_2'(k), \cdots, v_M'(k)\right]^T$ represent data at location k from the M virtual coils. Coil compression can be formulated as (13,14):

$$v'(k) = Av(k), \quad (2)$$

where we need to find the coil compression matrix A. In this work, we are only interested in the data-based coil compression that does not assume explicit knowledge of the coil sensitivities. Previous studies have shown that the compression matrix A can be found by PCA in data-based coil compression (14). One good property of this compression matrix by PCA is that $AA^H = I$. Assume $n_v(k)$ is the IID Gaussian noise in $v(k)$, then the noise in $v'(k)$ is $n_{v'}'(k) = An_v(k)$. When the acquired data has sufficient SNR (usually true for dedicated coils), this compression matrix A is not sensitive to noise $n_v(k)$ and $n_{v'}'(k)$ is close to IID noise. Sufficient SNR in auto calibration signals (ACS) is assumed for the rest of the paper. SNR measurements in SSOS images can be found in reference (26). The intensity at each pixel in the SSOS image is noncentral chi distributed. Noise in SSOS images can be reduced by coil compression because of the decreased numbers of coils. This is also known as the denoising effect (27).

Since the set of the virtual coils and the set of the original coils approximately span the same space, each original coil can also be approximated by a linear combination of the virtual coils:

$$v(k) = A^H v'(k) + \epsilon, \quad (3)$$

where ε is the approximation error. If the number of virtual coils is fixed, the coil compression problem can be written as a minimization of the compression error:

$$\begin{array}{ll} \underset{A}{\text{minimize}} & \sum_k \| \left(A^H A - I \right) v(k) \|^2 \\ \text{subject to} & AA^H = I, \end{array} \quad (4)$$

The solution for Eq. (4) can also be found by PCA or SVD (13–15).

Geometric-decomposition Coil Compression for Cartesian Sampling

To illustrate the idea behind GCC, let us consider for example the 3D Cartesian case in Fig. 1(a). The figure shows a body coil array, the imaging volume and the readout direction. In general, 3D *k*-space data obtained using this array will contain non-negligible signal in all the coils. Therefore, straightforward coil compression in *k*-space using the same compression matrix will often require a substantial number of virtual coils.

However, slice 1 and slice 100 are seen by a completely different subset of coils in the coil array. One slice represents one spatial location in the readout direction. Since the readout direction is fully sampled, the original data can be spatially decomposed into separable 2D *k*-spaces by a Fourier transform along the readout direction. Data in each new coordinate (x, k_y, k_z) have non-negligible signal in fewer coils, and can be compressed into a smaller number of virtual coils if different compression matrices are applied at each spatial location. This concept can be extended to decomposition in additional fully sampled dimensions. For simplicity, we will limit most of the discussion to geometric-decomposition only along the readout direction. GCC for 3D Cartesian datasets can be formulated as:

$$\begin{array}{ll} \underset{A_x}{\text{minimize}} & \sum_{x, k_{yz}} \| \left(A_x^H A_x - I \right) v\left(k_{yz} \right) \|^2 \\ \text{subject to} & A_x A_x^H = I, \end{array} \quad (5)$$

Here, $A_x \in \mathscr{C}^{M \times N}$ are a family of compression matrices at different spatial locations along the x axis (the readout direction) and $v_x\left(k_{yz}\right) \in \mathscr{C}^{N \times 1}$ are data at some spatial location x and 2D k-space coordinate $k_{yz} = [k_y, k_z]^T$ in the y — z plane.

This optimization is separable. The compression matrices A_x can be separately calculated by SVD at each x location using the following steps:

1. Compute an inverse Fourier transform of the *k*-space multicoil data along the readout direction into [x, k_y, k_z] coordinates.
2. At each x location, construct a data matrix X_x in which each row consists of all the $v_x(k_{yz})$ from an individual original coil. Often, just ACS data are sufficient.
3. Perform SVD of X_x:

$$X_x = U_x \Sigma_x (V_x)^H, \quad (6)$$

Take the first M rows of $(U_x)^H$ to form an initial compression matrix A_x^0.

4. Repeat Step 2 and 3 until compression matrices are calculated for all slices.

Virtual Coil Alignment

The SVD method is good at selecting the signal subspace. However, the compression matrices are not unique. In fact, for any unitary matrix $P \in \mathscr{C}^{M\times M}$, the product,

$$A_x = PA_x^0, \quad (7)$$

is a valid solution to Eq. (5). Directly using the solution A_x^0 obtained by SVD as the compression matrices is likely to introduce non-smooth virtual coil sensitivities along the x direction. This can lead to failure of the ACPI reconstructions, particularly when 3D synthesis kernels are used (28). To address this problem, we introduce an additional alignment stage in which we find new compression matrices $A_x = P_x A_x^0$ such that the resulting virtual coils are smooth in the x direction.

Figure 1(b) illustrates the basic idea behind this alignment. Data in each slice along the readout direction lie in a smaller subspace that is spanned by the physical coils. In the figure, the subspaces of slice 1 and 2 are illustrated as planes in 3D. The subspaces of nearby slices are very close but not the same since the physical coil sensitivities vary slightly from slice to slice. In Fig. 1(b), the subspace of slice 2 is slightly tilted with respect to the subspace of slice 1. The compressed virtual coils can be any orthonormal vectors that span the subspace. Aligned virtual coils are sets of virtual coils in each subspace that are closest in Euclidean distance.

This alignment can be achieved by solving the following optimization problem:

$$\begin{array}{ll} \underset{A_x, P_x}{\text{minimize}} & \sum_x \|A_x - A_{x-1}\|_F^2 \\ \text{subject to} & A_x = P_x A_x^0, \\ & P_x (P_x)^H = (P_x)^H P_x = I, \end{array} \quad (8)$$

where the variables are P_X and A_X, and $\|A_X — A_{X — 1}\|_F$ is the Frobenius norm of the matrix $(A_X — A_{X — 1})$. This problem does not have a unique solution either. However, if we set $P_X = I$ at a specific slice x, the optimal solution of P_X at the other slices is then unique. From now on, without loss of generality, we assume $P_1 = I$. This problem actually has an analytical solution. The orthogonal alignment matrices P_X for all slices can be found sequentially by the following steps:

1. Given $A_{X — 1}$, define the matrix $C_x \in \mathscr{C}^{M\times M}$ to be $C_x = A_x^0 (A_{x-1})^H$.
2. Compute SVD: $C_x = U_x^C \Sigma_x^C \left(V_x^C\right)^H$.
3. Set $P_x = V_x^C \left(U_x^C\right)^H$ and $A_x = P_x A_x^0$. This A_X minimizes $\|A_x - A_{x-1}\|_F^2$.
4. Repeat steps 1-3 until the alignment is done for all slices.

Since each P_X minimizes $\|A_x - A_{x-1}\|_F^2$, $\sum_x \|A_x - A_{x-1}\|_F^2$ is ultimately minimized (see Appendix A). After the compression matrices are carefully aligned, the coil compression can be simply performed according to Eq. (2) at each x location. The final step is to compute the Fourier transform of the virtual coil data along x direction. At this point, the original 3D data from N physical coils have been compressed into M spatially smooth virtual coils. For 2D Cartesian acquisitions, similar steps can be applied: GCC is applied in (x, k_y) space, and all the acquired data at each slice can be used to formulate matrix X_X.

Relationship to Walsh's Adaptive Coil Combination

In the extreme case when *k*-space is fully sampled and there is no acceleration, we can perform geometric-decomposition in all three dimensions. Computing the coil compression for every spatial location is not possible since one measurement at each location is not enough to determine the subspace by SVD. However, since the sensitivities are smooth, we could use a neighborhood around a particular location to compute the compression matrix. This is exactly the approach described by Walsh *et al.* (29). The "best" possible coil compression is of course compressing data into a single virtual coil using compression matrices that span the sensitivity maps.

Image Reconstruction Using Compressed Virtual Coils

Any PI or constrained reconstruction method can be directly applied to the virtual coils for reconstruction. In this paper, we simulate GE Healthcare product ACPI method (28), which belongs to the GRAPPA (3) family of reconstructions. We also use $\ell_1\text{–SPIRiT}$ (8), which is an iterative autocalibrating parallel imaging compressed sensing reconstruction. Both reconstructions are coil-by-coil methods that generally include two sequential steps in the reconstruction: calibration and data synthesis. The total computation complexity of both reconstruction methods is approximately $O(N^2)$, where N is the number of coils. Brief computational analysis of the two reconstructions can be found in reference (24) and Tab. 1. With coil compression applied to the acquired data, the number of coils is significantly reduced and the reconstruction is highly accelerated.

PI is sometimes limited in cases with a tight imaging FOV, where the reconstructed FOV is smaller than the object being imaged. ACPI methods (*e.g.*, GRAPPA or SPIRiT (8)), which do not rely on the direct measurements of the coil sensitivities, are not sensitive to a tight FOV (25). The proposed coil compression is also not susceptible to tight FOV artifacts for two reasons. First, the *k*-space ACS are independent of the actual imaging FOV. The tight FOV only increases the spacing between *k*-space lines, but does not change the *k*-space lines. Thus the calculations of the compression matrices are not affected by a tight FOV. Second, at each slice, the entire 2D *k*-space data from each coil are combined with the same compression weighting to construct the virtual coils. This compression is also independent of the imaging FOV. Therefore, although the virtual coils may have a tight imaging FOV, the construction of the virtual coils is not affected. As long as the ACPI or CS methods that follow the coil compression are not susceptible to a tight FOV, the related artifacts will not appear in the final reconstruction.

Materials and Methods

The proposed GCC was evaluated on two 3D datasets. Both of them were acquired on a 3T GE MR750 scanner (GE Healthcare, Waukesha, WI, USA) with Cartesian sampling using a 32 channel pediatric coil. The pediatric coil contains two 4×4 coil arrays, with one on top of the subject and the other embedded in the table (9). The first dataset was fully sampled, and acquired with the following imaging parameters: TE/TR = 0.944 ms/3.832 ms, flip angle (FA) = 15°, bandwidth (BW) = ± 64 kHz, FOV = 35 cm, slice thickness = 2 mm, and matrix size = 192 × 224 × 184. The second dataset was accelerated in both phase encoding directions. The acquisition sampling was chosen according to a poisson-disc distribution (30) with an overall reduction factor $R \approx 4$. The imaging parameters were: TE/TR = 1.128 ms/4.832 ms, FA = 15°, BW = ± 64 kHz, FOV = 38 cm, slice thickness = 2 mm, matrix size = 308 × 230 × 156, and ACS = 24 × 20.

Coil Compression Methods Comparison

The first dataset was used to compare two coil compression methods: (1) single coil compression (SCC) and (2) the proposed GCC. For SCC, the same compression matrix was applied to compress the entire 3D dataset. The compression matrix was calculated using a 24 × 24 × 24 ACS in the center *k*-space. For GCC, the proposed steps were applied, including the alignment of the compression matrices. In both cases, various numbers of virtual coils were examined. SSOS images were used in all cases for coil combination. The SSOS images from the 32 original coils were used as the reference. The difference images between the reference and coil compression results were calculated in each case. The normalized root mean squared error (nRMSE) of SSOS images was also calculated using the formula:

$$nRMSE=\frac{1}{\max(x)-\min(x)}\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(x(i)-\tilde{x}(i))^2}, \quad (9)$$

where x is the reference, $\tilde{x}$ is the coil compression result, and N_p is the number of image pixels.

To compare the denoising effect of SCC and GCC, the following experiment was performed. The first dataset was normalized first (maximum intensity normalized to 1). IID Gaussian noise (SNR = 20 for each original coil) was added to the normalized dataset. The new dataset with the added noise was referred to as the noisy dataset, and the original dataset was referred to as the noiseless dataset. Coil compression matrices for SCC with 6 and 16 virtual coils and GCC with 6 virtual coils were calculated from the noisy dataset. These compression matrices were then used to compress both the noisy dataset and only the added noise. The SSOS images of the noiseless dataset, noisy dataset, GCC results of the noisy dataset and SCC results of the noisy dataset were compared. The histograms of the added noise in these reconstructions were also compared to demonstrate the denoising effect. The noise covariance matrices of the added noise in 32 coils, 6 virtual coils after GCC, 6 virtual coils after SCC and 16 virtual coils after SCC were calculated.

To test the robustness of GCC and SCC, the denoising experiment was repeated 300 times. The coil compression matrices were different each time due to the different random noise added. The noiseless dataset was compressed by these compression matrices, and the SSOS images of GCC and SCC results were recorded each time. The pixelwise signal standard deviations (PSTD) of GCC and SCC results were calculated, and their histograms were compared.

GCCs with different number of virtual coils were simulated with three different readout directions: superior/inferior (commonly used as the readout direction), left/right and anterior/posterior. The relationship of nRMSE versus the number of virtual coils was studied in all cases.

To demonstrate the importance of the alignment of the compression matrices in GCC, 6 virtual coils before and after alignment were compared for the first dataset. Magnitude and phase images of single virtual coils in both cases were studied.

Image Reconstruction using Virtual Coils

To demonstrate the feasibility of combining GCC with ACPI and CS reconstruction, three experiments were carried out. First, the fully sampled dataset was uniformly undersampled by a factor of 2 in both phase encoding directions with ACS 28 × 28. The total reduction factor R was 3.7. The undersampled dataset was reconstructed by: (1) simulated GE Healthcare product ACPI method (28) using 32 original coils; and (2) GCC followed by the same ACPI method using 6 virtual coils. The ACPI kernel size was 5 × 7 × 7 ($k_x \times k_y \times k_z$). SSOS images were reconstructed in both cases. Differences between the two reconstructions were compared. The theoretical computation of each steps in this ACPI method can be found in reference (24).

Second, the fully sampled dataset was used to simulate cases with a tight imaging FOV. The FOV in k_y phase encoding direction was reduced by 20 percent, resulting in an overlapped reconstructed FOV. The dataset with a tight FOV was then undersampled by a factor of 2 in both phase encoding directions with ACS 28 × 28. Similar to the previous experiment, the same ACPI reconstruction method with the same kernel size using both the 32 original coils and 6 virtual coils were performed.

Third, the pseudo-randomly undersampled dataset was reconstructed by the following methods: (1) $\ell_1-\text{SPIRiT}$ using 32 original coils; (2) SCC followed by $\ell_1-\text{SPIRiT}$ using 16 virtual coils; (3) GCC followed by $\ell_1-\text{SPIRiT}$ using 6 virtual coils; (4) GCC without the alignment of the coil compression matrices followed by $\ell_1-\text{SPIRiT}$ using 6 virtual coils. The SPIRiT kernel size was 5 × 7 × 7 ($k_x \times k_y \times k_z$). SSOS coil combination was used in all cases. Theoretical computation, numerical examples, and actual reconstruction time in each step of $\ell_1-\text{SPIRiT}$ were calculated and compared.

In the coil compression methods comparison, two coil compression methods were implemented in MATLAB (The MathWorks, Natick, MA, USA) on a 8-core 2.8GHz Intel Xeon E5462 PC with 32 GB DRAM. The GE Healthcare product ACPI reconstruction method was also simulated in Matlab. For $\ell_1-\text{SPIRiT}$ reconstructions, SCC and GCC were implemented in C++, and $\ell_1-\text{SPIRiT}$ was implemented in CUDA on the same PC with a 30-core, 8-SIMD 1.48 GHz GTX285 GPGPU with 2 GB GDRAM (30,31).

Results

Coil Compression Methods Comparison

The number of virtual coils was first set to 6 for both SCC and GCC. The experimental results on the first dataset are shown in Fig. 2. Compared with the reference images, the SCC compression results had more compression loss while the results of GCC looked very similar to the reference images. The 5× difference images between the SCC images and the reference images further showed large compression error due to the non-optimal coil compression. However, the compression error was very small and negligible in GCC. The nRMSE values were 0.020 for SCC and 0.005 for GCC. The experimental results showed that GCC had much better compression performance than SCC for a fixed number of virtual coils.

In the denoising experiment, the SSOS images of all the reconstructions are shown in Fig. 3a. The denoising effect of GCC and SCC can been seen as the reduced background noise after coil compression. The noise covariance matrices of the added noise in 32 coils, 6 virtual coils after GCC, 6 virtual coils after SCC and 16 virtual coils after SCC are shown in Fig. 3b. The added noise in the virtual coils was very close to IID, and the noise standard deviation of each virtual coil was approximately the same as that of each individual coil before coil compression. The histograms of the added noise in all SSOS images (Fig. 3a) are shown in Fig. 3c. The SSOS of the added noise approximately followed a chi distribution. The denoising effect of coil compression can be considered as the removal of those virtual coils that did not contain significant signal and were dominated by IID noise.

The histograms of PSTD over 300 SSOS images in the Monte Carlo experiment are shown in Fig. 3d. The PSTD of GCC results was very small (up to 0.0005). This indicated that

GCC results of the noiseless dataset did not change much over 300 experiments, and were not very sensitive to noise in ACS. Compared with the added noise ($\sigma = 0.02$ for each coil), we can see that if GCC is applied to the noisy dataset, the dominant signal variation of GCC results would directly come from the added noise in the dataset instead of the variation of compression matrices. On the other hand, PSTD of SCC results was even smaller since the size of ACS was larger in SCC than GCC. In cases where ACS is highly corrupted by noise, SCC would be more robust than GCC.

Next, the number of virtual coils was set from 1 to 32, and the corresponding nRMSE values were calculated for both coil compression methods. Similar experiments were carried out with three readout directions. The nRMSEs were plotted as a function of the number of virtual coils in Fig. 4. In Fig. 4, we can see that SCC always had larger compression error than GCC. In order to achieve the same level of compression error as GCC, more virtual coils were needed for SCC. This is due to the fact that SCC can not effectively exploit the spatially varying data redundancy. Figure 4 also shows that the nRMSE performance of GCC depended on the readout direction used in the data acquisition. Acquisition with the common superior/inferior readout direction only needed 6 virtual coils to represent the same information from 32 original coils, while SCC needed approximately 16 virtual coils.

The magnitude and phase images of the 6 virtual coils from GCC with and without proper alignment are shown in Fig. 5. From Fig. 5(a,b), we can notice abrupt changes along x direction (superior/inferior) in both magnitude and phase images without the alignment of the coil compression matrices. This indicates that the coil sensitivities of the virtual coils were not smooth. Non-smooth coil sensitivities can cause significant problems for the calibration step in ACPI reconstructions. With proper alignment, the magnitude and phase images (Fig. 5(c,d)) were smooth enough for calibration. It is important to note that the alignment affects both the phase and the magnitude images of the virtual coils.

Image Reconstruction using Virtual Coils

The simulated product ACPI reconstruction was applied in the experiment with uniform undersampling of the first dataset. The reconstructions with 32 original coils and 6 virtual coils were carried out separately. The total reconstruction times (including coil compression) were 876 seconds and 55 seconds respectively. Coil compression accelerated the reconstruction by approximately a factor of 16. Compared with the fully sampled data as the reference, the nRMSEs of the ACPI reconstructions ($R = 3.7$) were 0.008 with 32 original coils and 0.010 with 6 virtual coils. The reconstruction errors were very small in both cases.

Results for the experiment with a tight imaging FOV simulation are shown in Fig. 6b. As expected, both the proposed coil compression and the ACPI reconstruction were not sensitive to the tight FOV. The reconstructions were robust and had similar reconstruction errors to the previous experiment with the normal imaging FOV. The nRMSEs of the ACPI reconstructions ($R = 3.7$) were 0.008 with 32 original coils and 0.011 with 6 virtual coils.

The second dataset with random undersampling was reconstructed by ℓ_1−SPIRiT both with and without GCC. The analytical formula, numerical examples, and actual reconstruction time of each step in ℓ_1−SPIRiT are shown in Tab. 1. From Tab. 1 we can see that coil compression theorectically reduces the calibration computation by a factor of over 100, and data synthesis by a factor of 28. Data synthesis was performed on GPGPU, and the actual time spent was reduced by a factor of 6. GCC did require extra computation, but it was very fast, and negligible compared to ℓ_1−SPIRiT reconstruction. The total reconstruction time was reduced from 652 seconds to 46 seconds. The reconstruction time was speeded up by approximately a factor of 14. This is fast enough to be clinically practical. The reconstruction results with GCC were also compared with those with SCC (16 virtual coils).

The total reconstruction time with SCC was 134 seconds (approximately 3 times of that with GCC). The reconstruction results (shown in Fig. 7) had very similar image quality. Without the alignment of the coil compression matrices, the reconstructed images suffered from artifacts due to non-smooth virtual coil sensitivities and inaccurate SPIRiT kernel calibration.

Discussion

GCC uses the fact that the coil sensitivity variation in the readout direction is not used for acceleration in Cartesian data acquisition. The acquired data can be spatially separated in the readout direction. Therefore at each slice, GCC can effectively minimize the number of virtual coils, and hence the PI or CS reconstruction time. The performance of GCC depends on several factors including the original coil geometry and the readout direction used in the data acquisition. Generally speaking, the optimal readout direction is the direction where the number of effective virtual coils is maximized, so that largest acceleration can be achieved for PI. However, the actual readout direction is practically chosen for several clinical imaging considerations. The number of effective virtual coils in GCC can be determined by thresholding the singular values in SVD. The nRMSE is another useful indication for choosing the number of effective virtual coils.

The proposed GCC can also effectively compress 2D Cartesian datasets. Figure 8 shows the results of SCC and GCC on a simluated 2D fully sampled dataset. For a similar nRMSE (less than 0.01), only 3 virtual coils were needed in GCC compared to 9 virtual coils in SCC. This can be used to effectively compress time-series data from large arrays that appear, for example, in many fMRI experiments.

The initial compression matrices in GCC are calculated from the center *k*-space. In 3D ACPI datasets, the size of ACS at each x location is sufficiently large for this calculation. Further discussion of the required inputs of coil compression matrix calculation can be found in reference (14). However, in 2D datasets, the size of ACS at each x location is not usually big enough for the calculation. In this case, all the acquired data at one x location can be used to calculate the compression matrices; or the ACS in the adjacent x locations can be included to calculate the compression matrices based on the fact that the original coil sensitivities are smooth.

Both SCC and GCC require two major steps to achieve coil compression. The first step is to find the appropriate coil compression matrices; and the second step is to apply the coil compression matrices to the original coils and construct the virtual coils. Assuming the same number of virtual coils is chosen for both SCC and GCC, then the second step in SCC and GCC requires the same amount of computation (shown in Tab. 1). Take 3D ACPI datasets as an example. The first step in SCC only requires a SVD of the 3D ACS, which is a $O(N_{ACS,x}N_tN^2)$ computation, where $N_{ACS,x}$ is the number of data points chosen in the k_x direction for calibration, and N_t is the number of ACS points in two phase encoding directions. The first step in GCC requires two operations: the initial coil compression matrices calculation and the alignment of the compression matrices. The calculation of the initial compression matrices is a $O(N_xN_tN^2)$ computation, and can be accelerated by parallel computing; the alignment of the compression matrices is a $O(N_xNM^2)$ computation. The computation of the alignment is negligible compared to the initial compression matrices calculation. Since usually the number of virtual coils in GCC is less than SCC, the actual computation time of SCC and GCC is close and much smaller than the ACPI or CS reconstruction time. If a synthetic target coil, a direct virtual coil or flexible virtual coils is used as the ACPI reconstruction, GCC can speed up the reconstruction by replacing the source channels with the virtual coils.

Note that after the operation in Eq. 1, the noise in each coil is IID. When the coil compression matrices are calculated from ACS with sufficient SNR, they are usually not sensitive to noise. As a result, the noise in each virtual coil is close to IID. In GCC, different coil compression matrices are applied at different spatial locations, but the noise distribution does not change significantly in the entire set of virtual coils.

For multi-echo or multi-phase data acquisition, the spatially varying coil compression matrices can be calculated from data in the first echo or the first phase. The same compression matrices should be applied to data in the other echoes or phases so that GCC is consistent in the entire dataset.

The proposed method can also address the data storage problem. However it requires the access of raw data during data acquisition. To store the virtual coil data directly, ACS has to be first acquired so that the compression matrices can be calculated. Then for each of the other acquisitions, an inverse Fourier transform along the readout direction needs to be done before coil compression. Depending on the reconstruction method used, the compressed data can be stored either in *k*-space, which requires another Fourier transform along the readout direction, or (x, k_y, k_z) space.

Conclusion

Data-based coil compression can transform signals from a large number of physical coils into fewer virtual coils without the direct measurements of coil sensitivities. A geometric-decomposition coil compression technique for Cartesian datasets has been proposed in this work. The proposed method can minimize the number of virtual coils without noticeable compression loss. The compression matrices are properly-aligned to achieve smooth virtual coil sensitivities. Autocalibrating parallel imaging and compressed sensing reconstruction can be directly carried out on the virtual coils, which significantly reduces reconstruction time. The image quality of reconstructions using the virtual coils is comparable to that of reconstructions using the original coils. Based on the experimental results, 3D datasets from a 32 channel pediatric coil can be effectively compressed into 6 virtual coils. The proposed coil compression is also robust in cases with tight imaging FOVs.

Acknowledgments

The authors would like to thank Marcus Alley and Mark Murphy for the help with the offline reconstruction platform, Philip Beatty for the help with the simulated GE Healthcare product ACPI reconstruction routine, and Joseph Cheng for the help in the preparation of this manuscript.

This work was supported by NIH grants R01 EB009690, P41 RR09784, UC Discovery Grant #193037, American Heart Association Award #12BGIA9660006 and GE Healthcare.

Appendix A: Analytical solution of the alignment of the compression matrices

The objective function in Eq. (8) can be minimized sequentially at each slice. Assume P_{x-1} (and A_{x-1}) is known at slice $x - 1$, we can then minimize $\|A_x - A_{x-1}\|_F^2$ at slice x.

$$\begin{aligned} \|A_x - A_{x-1}\|_F^2 = \|P_x A_x^0 - A_{x-1}\|_F^2 &= Tr\left[\left(P_x A_x^0 - A_{x-1}\right)^H \left(P_x A_x^0 - A_{x-1}\right)\right] \\ &= Tr\left[\left(P_x A_x^0\right)^H P_x A_x^0\right] + Tr\left[(A_{x-1})^H A_{x-1}\right] - Tr\left[2(A_{x-1})^H P_x A_x^0\right] \\ &= Tr\left[\left(A_x^0\right)^H A_x^0\right] + Tr\left[(A_{x-1})^H A_{x-1}\right] - 2Tr\left[A_x^0 (A_{x-1})^H P_x\right] \\ &= Tr\left[\left(A_x^0\right)^H A_x^0\right] + Tr\left[(A_{x-1})^H A_{x-1}\right] - 2Tr\left[C_x P_x\right] \\ &= Tr\left[\left(A_x^0\right)^H A_x^0\right] + Tr\left[(A_{x-1})^H A_{x-1}\right] - 2Tr\left[U_x^C \Sigma_x^C \left(V_x^C\right)^H P_x\right] \\ &= Tr\left[\left(A_x^0\right)^H A_x^0\right] + Tr\left[(A_{x-1})^H A_{x-1}\right] - 2Tr\left[\Sigma_x^C \left(V_x^C\right)^H P_x U_x^C\right] \end{aligned} \quad (10)$$

with the use of $C_x = U_x^C \Sigma_x^C \left(V_x^C\right)^H$ and the property of $Tr[M_1 M_2] = Tr[M_2 M_1]$.

To minimize $\|A_x - A_{x-1}\|_F^2$ is equivalent to maximize $2Tr\left[\Sigma_x^C \left(V_x^C\right)^H P_x U_x^C\right]$. Define $Q = \left(V_x^C\right)^H P_x U_x^C = \left[q_{ij}\right]$. Since V_x^C, P_x, and U_x^C are all orthogonal matrices, then Q is also an orthogonal matrix, and $|q_{ij}| \le 1$. Define $\Sigma_x^C = \text{diag}\,(\sigma_{11}, \sigma_{22}, \ldots, \sigma_{MM})$, then $Tr\left[\Sigma_x^C \left(V_x^C\right)^H P_x U_x^C\right] = \sum_{k=1}^M \sigma_{kk} q_{kk} \le \sum_{k=1}^M \sigma_{kk}$. To maximize $Tr\left[\Sigma_x^C Q\right]$, we have $Q = I$, and $P_x^{opt} = V_x^C \left(U_x^C\right)^H$. Thus, the closed form solution for A is $A_x^{opt} = V_x^C \left(U_x^C\right)^H A_x^0$.

Define another set of unitary matrices R_X, where $R_x = P_{x-1}^H P_x$. Then $\|A_x - A_{x-1}\|_F^2 = \|P_x A_x^0 - P_{x-1} A_{x-1}^0\|_F^2 = \|R_x A_x^0 - A_{x-1}^0\|_F^2$. To prove the global optimality of A_x^{opt}, let us reformulate Eq. (8) into the following form,

$$\begin{array}{ll} \underset{R_x}{\text{minimize}} & \sum_x \|R_x A_x^0 - A_{x-1}^0\|_F^2 \\ \text{subject to} & R_x (R_x)^H = (R_x)^H R_x = I, \end{array} \quad (11)$$

Define $D_x = A_x^0 \left(A_{x-1}^0\right)^H$ and assume the SVD of D_X is $D_x = U_x^D \Sigma_x^D \left(V_x^D\right)^H$. Similar to the previous proof, it is easy to show that the optimal solution of the reformulated problem (Eq. (11)) is $R_x^{opt} = V_x^D \left(U_x^D\right)^H$. In fact, R_x^{opt} is the global optimal solution of Eq. (11), since for different slices x_1 and x_2, $\|R_{x_1} A_{x_1}^0 - A_{x_1-1}^0\|_F^2$ and $\|R_{x_2} A_{x_2}^0 - A_{x_2-1}^0\|_F^2$ are independent. Notice $C_X = D_X (P_{X-1})^H$, then $D_x = U_x^C \Sigma_x^C \left(V_x^C\right)^H P_{x-1}$. Without loss of generality, we can assume $U_x^D = U_x^C$, $\Sigma_x^D = \Sigma_x^C$ and $V_x^D = (P_{x-1})^H V_x^C$. Therefore, $R_x^{opt} = V_x^D \left(U_x^D\right)^H = \left((P_{x-1})^{opt}\right)^H P_x^{opt}$. When R_x^{opt} achieves the global minimum of $\sum_x \|R_x^{opt} A_x^0 - A_{x-1}^0\|_F^2$, $P_x^{opt}\left(A_x^{opt}\right)$ achieves the same global minimum of $\sum_x \|P_x^{opt} A_x^0 - A_{x-1}^{opt}\|_F^2 = \sum_x \|R_x^{opt} A_x^0 - A_{x-1}^0\|_F^2$. In other words, the previous solution to Eq. (8) $\left(P_x^{opt} = V_x^C \left(U_x^C\right)^H \quad \text{and} \quad A_x^{opt} = P_x^{opt} A_x^0\right)$ is global optimal.

References

1. Pruessmann K, Weiger M, Scheidegger M, Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med. 1999; 42:952–962. [PubMed: 10542355]
2. Sodickson D, Manning W. Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays. Magn Reson Med. 1997; 38:591–603. [PubMed: 9324327]

3. Griswold M, Jakob P, Heidemann R, Nittka M, Jellus V, Wang J, Kiefer B, Haase A. Generalized autocalibrating partially parallel acquisitions (GRAPPA). Magn Reson Med. 2002; 47:1202–1210. [PubMed: 12111967]
4. Roemer P, Edelstein W, Hayes C, Souza S, Mueller O. The NMR phased array. Magn Reson Med. 1990; 16:192–225. [PubMed: 2266841]
5. Lustig M, Donoho D, Pauly J. Sparse MRI: The application of compressed sensing for rapid MR imaging. Magn Reson Med. 2007; 58:1182–1195. [PubMed: 17969013]
6. Liang D, Liu B, Wang J, Ying L. Accelerating SENSE using compressed sensing. Magn Reson Med. 2009; 62:1574–1584. [PubMed: 19785017]
7. Otazo R, Kim D, Axel L, Sodickson D. Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion MRI. Magn Reson Med. 2010; 64:767–776. [PubMed: 20535813]
8. Lustig M, Pauly J. SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary *k*-space. Magn Reson Med. 2010; 64:457–471. [PubMed: 20665790]
9. Vasanawala, S.; Grafendorfer, T.; Calderon, P.; Scott, G.; Alley, M.; Lustig, M.; Brau, A.; Sonik, A.; Lai, P.; Alagappan, V.; Hargreaves, B. Proceedings of the 19th Annual Meeting of ISMRM. Montreal: 2011. Millimeter isotropic resolution volumetric pediatric abdominal MRI with a dedicated 32 channel phased array coil.; p. 161
10. Zhu Y, Hardy C, Sodickson D, Giaquinto R, Dumoulin C, Kenwood G, Niendorf T, Lejay H, McKenzie C, Ohliger M, Rofsky N. Highly parallel volumetric imaging with a 32-element RF coil array. Magn Reson Med. 2004; 52:869–877. [PubMed: 15389961]
11. McDougall M, Wright S. 64-channel array coil for single echo acquisition magnetic resonance imaging. Magn Reson Med. 2005; 54:386–392. [PubMed: 16032696]
12. Schmitt M, Potthast A, Sosnovik D, Polimeni J, Wiggins G, Triantafyllou C, Wald L. A 128-channel receive-only cardiac coil for highly accelerated cardiac MRI at 3 Tesla. Magn Reson Med. 2008; 59:1431–1439. [PubMed: 18506789]
13. Buehrer M, Pruessmann K, Boesiger P, Kozerke S. Array compression for MRI with large coil arrays. Magn Reson Med. 2007; 57:1131–1139. [PubMed: 17534913]
14. Huang F, Vijayakumar S, Li Y, Hertel S, Duensing G. A software channel compression technique for faster reconstruction with many channels. Magn Reson Imaging. 2008; 26:133–141. [PubMed: 17573223]
15. King S, Varosi S, Duensing G. Optimum SNR data compression in hardware using an eigencoil array. Magn Reson Med. 2010; 63:1346–1356. [PubMed: 20432305]
16. King, S.; Duensing, G. Proceedings of the 11th Annual Meeting of ISMRM. Toronto: 2003. The MRI eigencoil: 2N-channel SNR with N-receivers.; p. 712
17. Doneva M, Bornert P. Automatic coil selection for channel reduction in SENSE-based parallel imaging. Magn Reson Mater Phy. 2008; 21:187–196.
18. Feng S, Zhu Y, Ji J. Efficient large-array *k*-domain parallel MRI using channel-by-channel array reduction. Magn Reson Imaging. 2011; 29:209–215. [PubMed: 20980116]
19. Beatty, P.; Holmes, J.; Chang, S.; Bayram, E.; Brittain, J.; Reeder, S.; Brau, A. Proceedings of the 18th Annual Meeting of ISMRM. Stockholm: 2010. Coil-by-coil vs. direct virtual coil (DVC) parallel imaging reconstruction: an image quality comparison for contrast-enhanced liver imaging.; p. 2879
20. Chen, W.; Hu, P.; Meyer, C. Proceedings of the 16th Annual Meeting of ISMRM. Toronto: 2008. Rapid partially parallel reconstruction using a single synthetic target coil.; p. 1296
21. Huang F, Lin W, Duensing G, Reykowski A. A hybrid method for more efficient channel-by-channel reconstruction with many channels. Magn Reson Med early view.
22. Stemmer, A.; Jellus, V.; Kannengiesser, S.; Kiefer, B. Proceedings of the 16th Annual Meeting of ISMRM. Toronto: 2008. Channel compression for BLADE.; p. 1274
23. Huang, F.; Vijayakumar, S.; Akao, J. Conf Proc IEEE Eng Med Biol Soc. Shanghai: 2005. Software compression for partially parallel imaging with multi-channels.; p. 1348-1351.
24. Brau A, Beatty P, Skare S, Bammer R. Comparison of reconstruction accuracy and efficiency among autocalibrating data-driven parallel imaging methods. Magn Reson Med. 2008; 59:382–395. [PubMed: 18228603]

25. Griswold M, Kannengiesser S, Heidemann R, Wang J, Jakob P. Field-of-view limitations in parallel imaging. Magn Reson Med. 2004; 52:1118–1126. [PubMed: 15508164]

26. Constantinides C, Atalar E, McVeigh E. Signal-to-noise measurements in magnitude images from NMR phased arrays. Magn Reson Med. 1997; 38:852–857. [PubMed: 9358462]

27. Vijayakumar, S.; Huang, F.; Akao, J.; Limkeman, M.; Duensing, G. Proceedings of the 15th Annual Meeting of ISMRM. Berlin: 2007. Channel compression and denoising.; p. 1908

28. Beatty, P.; Brau, A.; Chang, S.; Joshi, S.; Michelich, C.; Bayram, E.; Nelson, T.; Herfkens, R.; Brittain, J. Proceedings of the 15th Annual Meeting of ISMRM. Berlin: 2007. A method for autocalibrating 2-D accelerated volumetric parallel imaging with clinically practical reconstruction times.; p. 1749

29. Walsh D, Gmitro A, Marcellin M. Adaptive reconstruction of phased array MR imagery. Magn Reson Med. 2000; 43:682–690. [PubMed: 10800033]

30. Vasanawala, S.; Murphy, M.; Alley, M.; Lai, P.; Keutzer, K.; Pauly, J.; Lustig, M. Proceedings of IEEE International Symposium on Biomedical Imaging. Chicago: 2011. Practical parallel imaging compressed sensing MRI: Summary of two years of experience in accelerating body MRI of pediatric patients.; p. 1039-1043.

31. Murphy, M.; Keutzer, K.; Vasanawala, S.; Lustig, M. Proceedings of the 18th Annual Meeting of ISMRM. Stockholm: 2010. Clinically feasible reconstruction time for ℓ_1-SPIRiT parallel imaging and compressed sensing MRI.; p. 4854

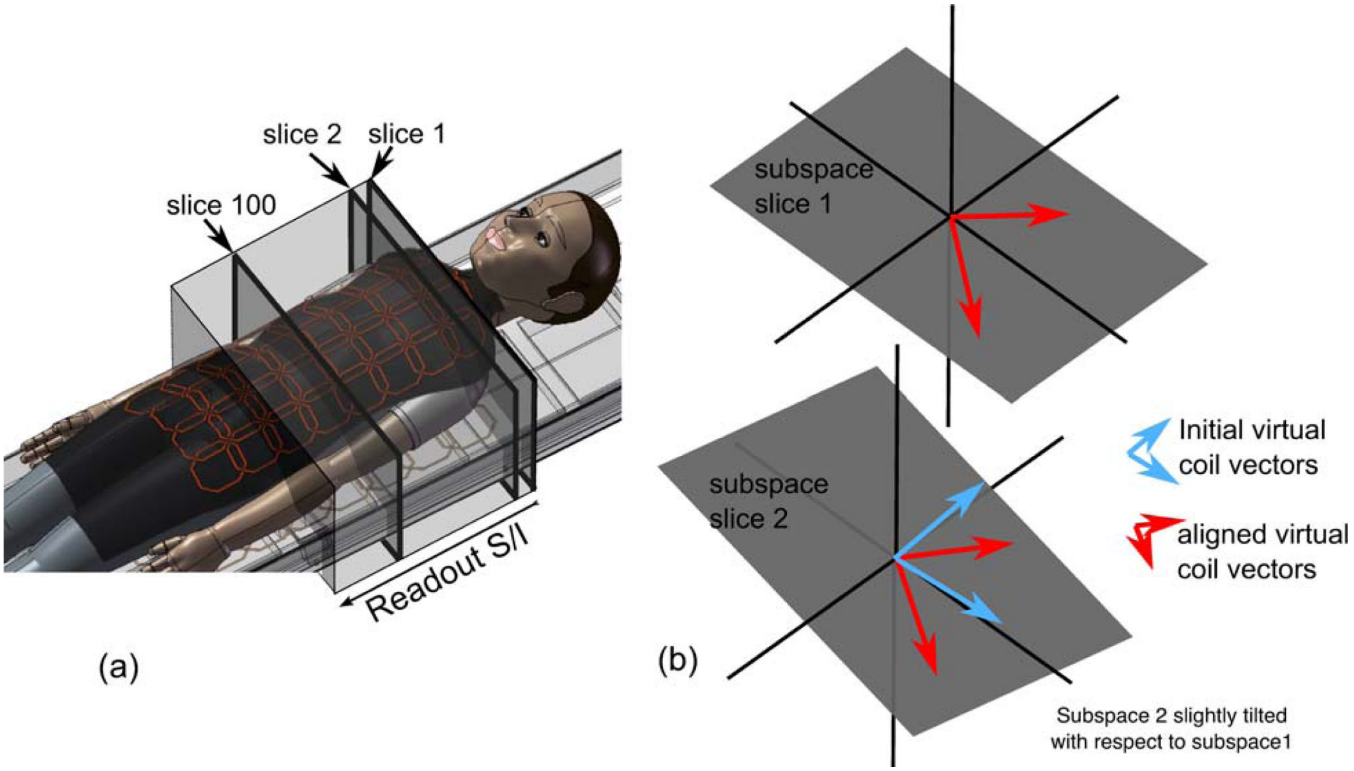
slice 2
slice 1
slice 100
Readout S/I
(a)
subspace slice 1
subspace slice 2
Initial virtual coil vectors
aligned virtual coil vectors
(b)
Subspace 2 slightly tilted with respect to subspace1

Figure 1.

The geometry of GCC and virtual coil alignment in 3D Cartesian imaging. (a) The imaging setup showing a body coil array and the imaging volume with S/I readout. Slices 1 and 2 along the readout “see” the same coils with slightly different sensitivities. On the other hand, slice 100 “sees” a completely different set of coils. This enables better compression after geometric decomposition. (b) Illustration of virtual coil alignment: Data in each slice lies in a smaller subspace (illustrated by planes in 3D) spanned by the compressed virtual coil vectors (arrows). The subspaces of nearby slices are close, but not the same. There are many sets of orthogonal vectors that span each subspace and the ones given by SVD can be arbitrary. The alignment finds new sets of orthogonal vectors in one subspace that are closest to the ones spanning the other. This process guarantees smoothness of the virtual coil sensitivities.

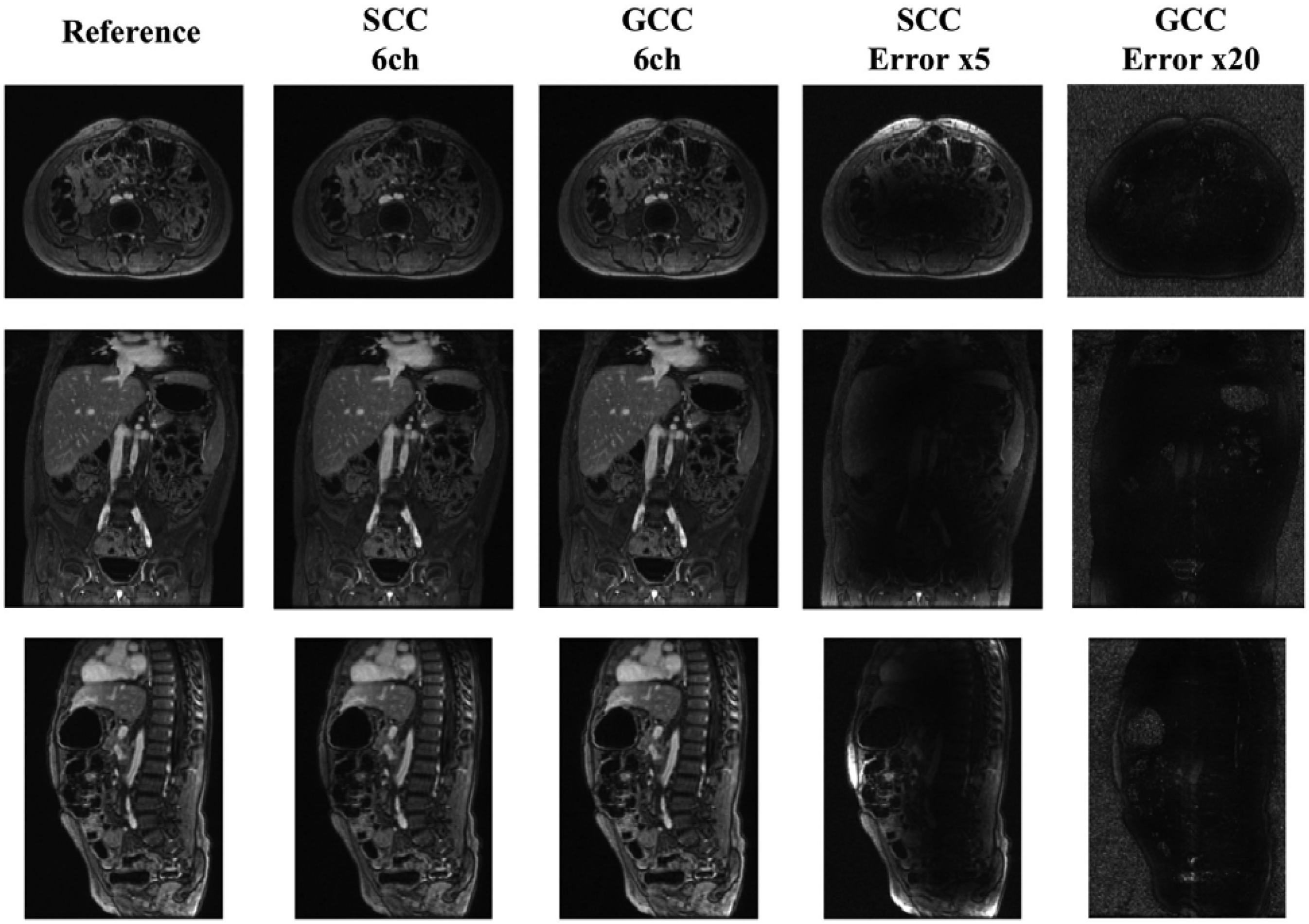

Figure 2.

Comparison of SCC and GCC. SSOS images (axial/coronal/sagital) were compared. The original images in the first column were used as the reference. The compression results of SCC and GCC with 6 virtual coils were shown in the second and third column respectively. Compression errors (5× in SCC and 20× in GCC) were shown in the fourth and last column. GCC had very little compression loss even with only 6 virtual coils, while SCC suffered from much larger compression loss.

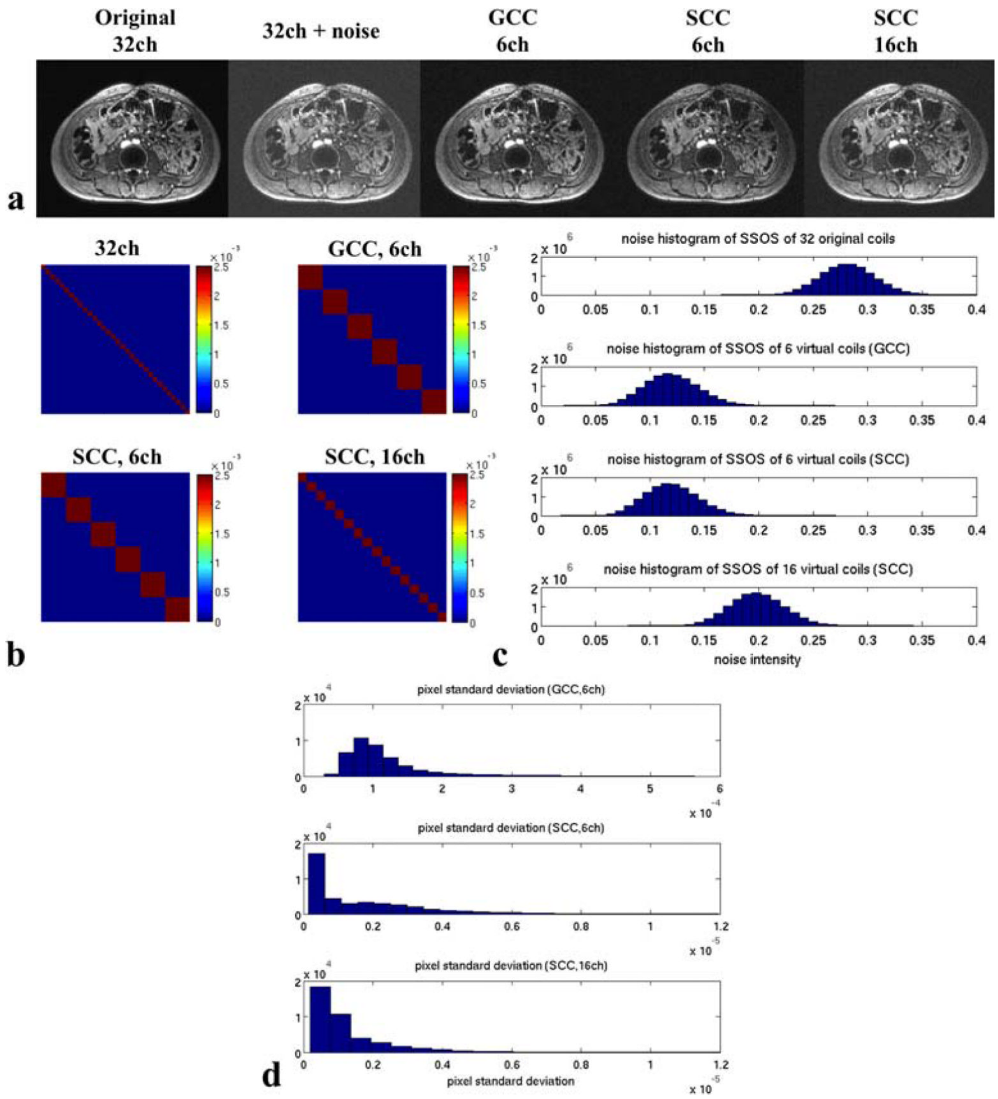

Figure 3.

Results of the denoising and Monte Carlo experiment. (a) SSOS images of the noiseless dataset, noisy dataset, GCC results of the noisy dataset and SCC results of the noisy dataset. The denoising effect can be seen as the reduced background noise after GCC and SCC. (b) Noise covariance matrices of the added noise in 32 coils, 6 virtual coils after GCC, 6 virtual coils after SCC and 16 virtual coils after SCC. Noise was close to IID after GCC or SCC. (c) Histograms of the SSOS of the added noise. The noise level was dependent on the number of virtual coils. The fewer the number of virtual coils, the less noise there was in SSOS images. (d) Histograms of PSTD of GCC with 6 virtual coils and SCC with 6 and 16 virtual coils. PSTD was very small (up to 0.0005) in GCC. SCC of the noiseless dataset was even less sensitive to the added noise due to larger ACS.

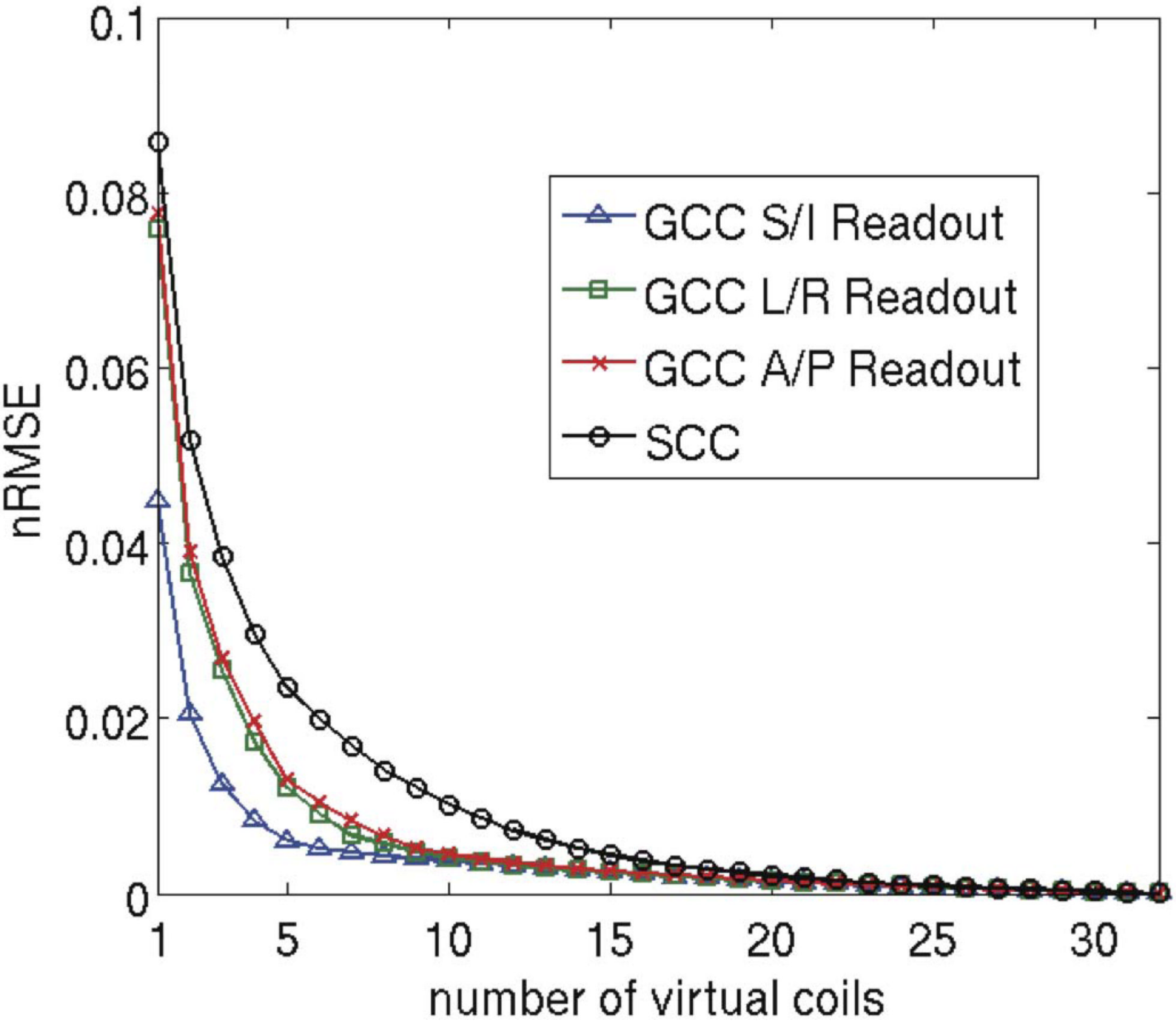

Figure 4.
Compression with different numbers of virtual coils and the corresponding nRMSE. The entire 3D dataset was used for nRMSE calculation. Data from 32 original coils were used as the reference. GCC had less compression loss than SCC. To achieve similar nRMSE as in GCC, more virtual coils were needed in SCC.

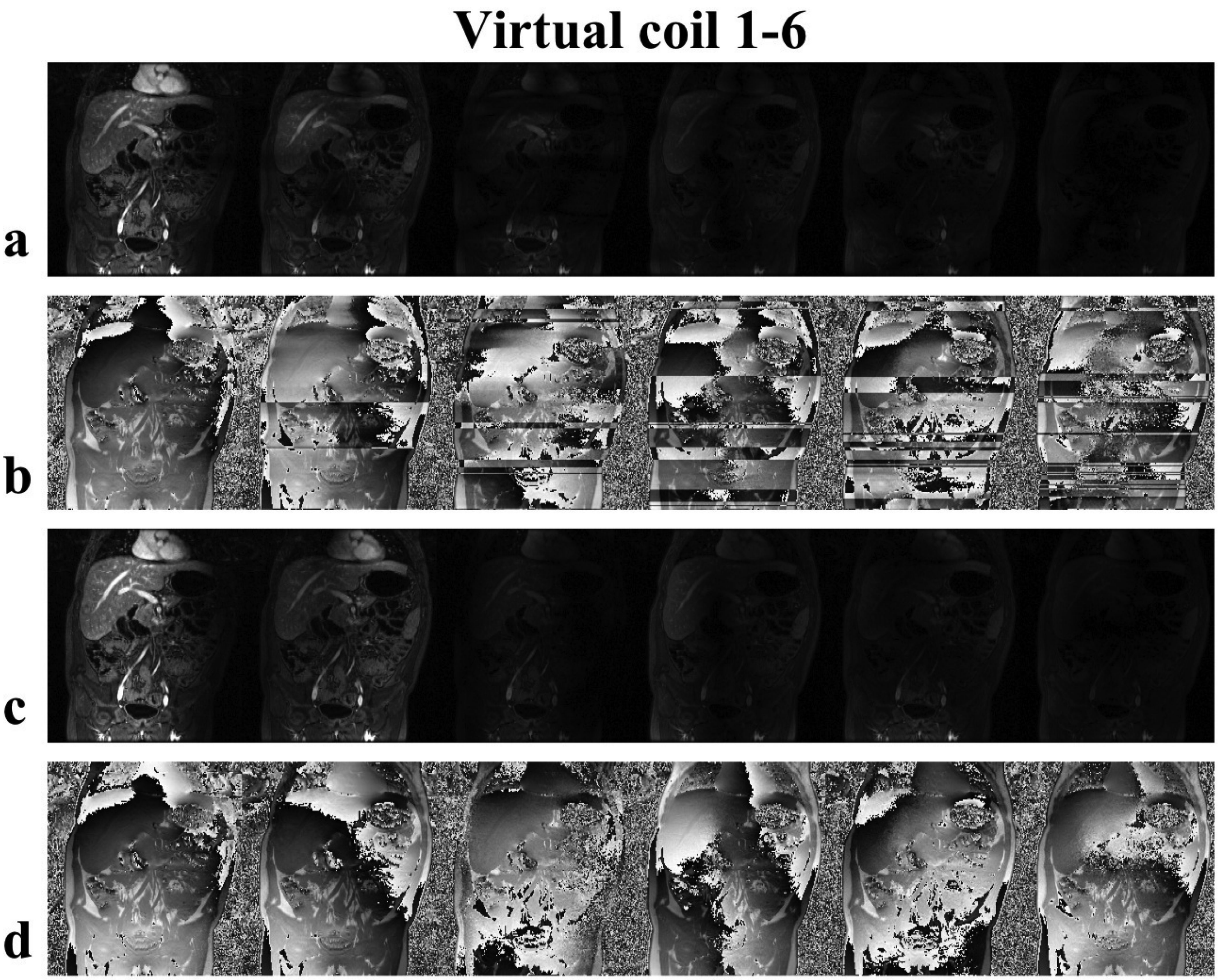

Figure 5.
Compression matrices alignment comparison. (a,b) Magnitude and phase images of the individual virtual coil from 1 to 6 by GCC without alignment. (c,d) Magnitude and phase images of the virtual coil from 1 to 6 by GCC after alignment. After proper alignment of the compression matrices, the images from individual virtual coil no longer have random abrupt phase changes or discontinuous coil sensitivities.

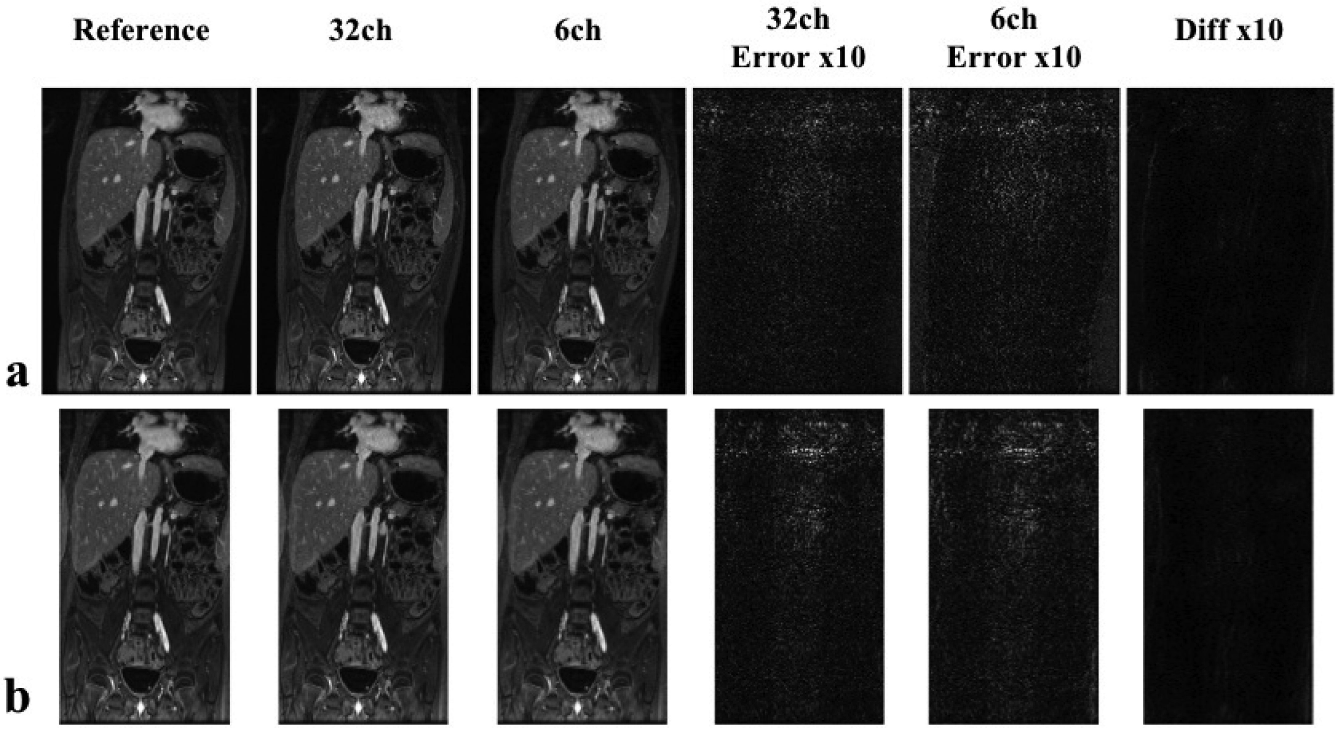

Figure 6.

Reconstruction results by the simulated product ACPI method with and without GCC. Reference images of the fully sampled data using 32 original coils were shown in the first column. ACPI reconstruction results of 32 original coils and 6 virtual coils were shown in the second and third column respectively. 10× error images in both reconstructions were shown in the fourth and fifth column. The last column is the difference between two ACPI reconstructions. (a) In the normal imaging FOV case, the nRMSE was 0.008 for reconstruction with 32 original coils, and 0.010 for the reconstruction with 6 virtual coils. (b) In the tight imaging FOV case, the nRMSE was 0.008 for reconstruction with 32 original coils, and 0.011 for the reconstruction with 6 virtual coils. ACPI reconstructions with GCC and with original data had very similar image quality independent of the imaging FOV.

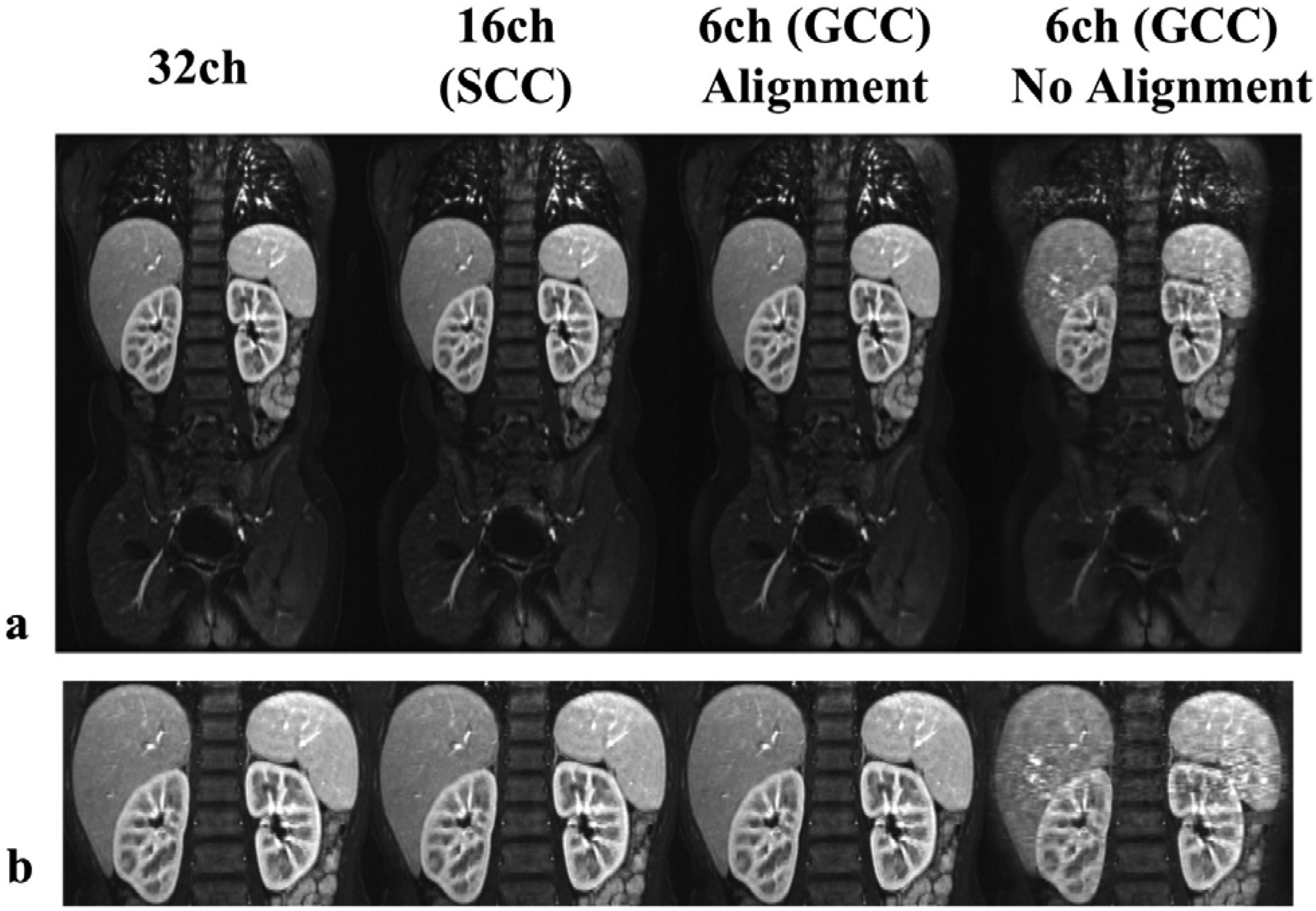

Figure 7.

In-vivo reconstruction results by ℓ_1−SPIRiT. (a) Reconstruction results of ℓ_1−SPIRiT with 32 original coils, with 16 virtual coils after SCC, with 6 virtual coils after GCC with the alignment of the coil compression matrices, and with 6 virtual coils after GCC without the alignment of the coil compression matrices; (b) Zoomed-in images of (a). The difference between ℓ_1−SPIRiT with 32 original coils, with 16 virtual coils after SCC, and with 6 virtual coils after GCC with the alignment of the coil compression matrices was very small and negligible. Without proper alignment of the coil compression matrices in GCC, the ℓ_1−SPIRiT reconstruction suffered from artifacts due to inaccurate calibration.

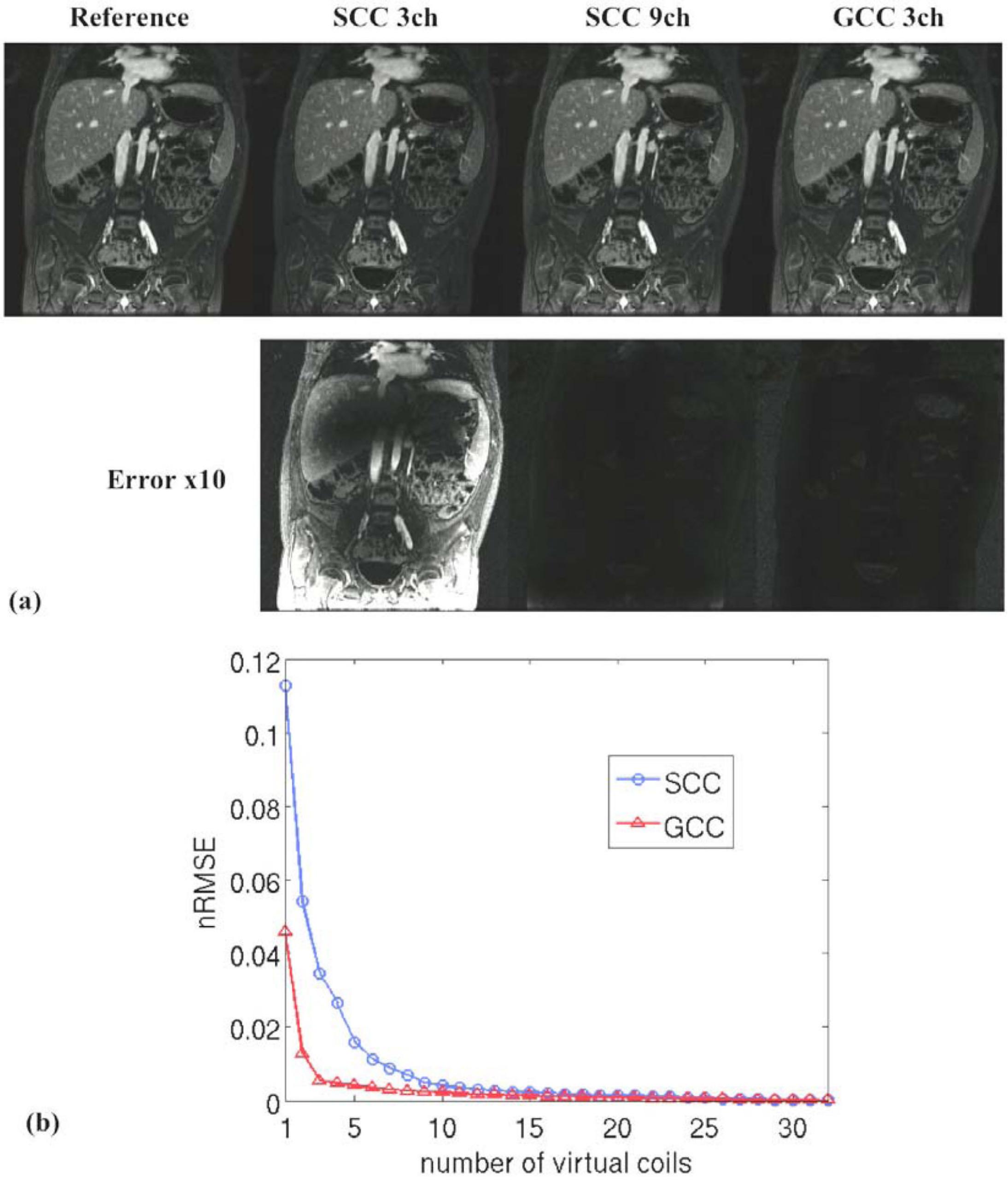

Figure 8.

Simulation of a 2D fully sampled Cartesian dataset (S/I readout direction) with SCC and GCC. (a) Coil compression results and errors (×10) of SCC with 3 and 9 virtual coils and GCC with 3 virtual coils; (b) Compression error (nRMSE) with different numbers of virtual coils using SCC and GCC. Only 3 virtual coils were needed in GCC compared to 9 in SCC.

Table 1

Approximate major computation required for $\ell_1-\text{SPIRiT}$ with and without GCC. The major computation of each step is measured by the number of complex-valued multiplications. The numerical example is calculated from the parameters used in the second dataset in Fig. 7. The number of samples in x, y, z directions are $N_x = 308$, $N_y = 230$, $N_z = 156$, the total ACS points in one coil $N_t = 480$, number of coils $N = 32$, number of virtual coils $M = 6$, SPIRiT kernel size $G_x = 5$, $G_y = G_z = 7$, reduction factor $R \approx 4$, and the number of iterations in data synthesis $N_i = 50$.

Method		**ℓ_1-SPIRiT**	**ℓ_1-SPIRiT GCC**
GCC SVD	Generalized Formula	-	$N_x N_t N^2$
	Numerical Example	-	1.5×10^8
	Actual Time (seconds)	-	2
GCC Compression	Generalized Formula	-	$\frac{1}{R} N_x N_y N_z NM$
	Numerical Example	-	5.3×10^8
	Actual Time (seconds)	-	3.5
Calibration	Generalized Formula	$NN_t(G_x G_y G_z N)^2$	$MN_t(G_x G_y G_z M)^2$
	Numerical Example	9.4×10^{11}	6.2×10^9
	Actual Time (seconds)	516	4
Data Synthesis	Generalized Formula	$N_i N_x N_y N_z N^2$	$N_i N_x N_y N_z M^2$
	Numerical Example	5.7×10^{11}	2.0×10^{10}
	Actual Time (seconds)	130	22