A fast approach to processing coil sensitivity maps using a neural network

M. Buehrer[1], P. Boesiger[1], S. Kozerke[1]

[1]Institute for Biomedical Engineering, ETH Zuerich, Zuerich, Zuerich, Switzerland

Introduction

Obtaining information about coil sensitivities is crucial when using sensitivity encoding (SENSE) [1]. Sensitivity maps can be calculated by dividing single-coil images by the body-coil reference. These raw sensitivities are often impaired by noise, especially in regions with low spin density (e.g. in the lungs). Knowing the coil sensitivities in these areas is, however, of importance when imaging dynamic objects. To address this problem SENSE reference scans are designed to gain as much signal-to-noise ratio as possible by choosing large voxel sizes in combination with signal averages which sufficiently blur sensitivity information. However, when imaging with small coil elements as contained in large coil arrays, higher spatial resolution is required to capture the variation in sensitivity close to the individual coil elements. For processing such high resolution maps a polynomial fitting approach has been described [1]. In this method regions exhibiting pure noise are identified by thresholding followed by an extrapolation of sensitivities into background regions. The polynomial fitting yields good results in most applications, but requires considerable computational time since processing is performed pixelwise. In this work a fast method is presented to smooth and extrapolate sensitivity information using a neural network.

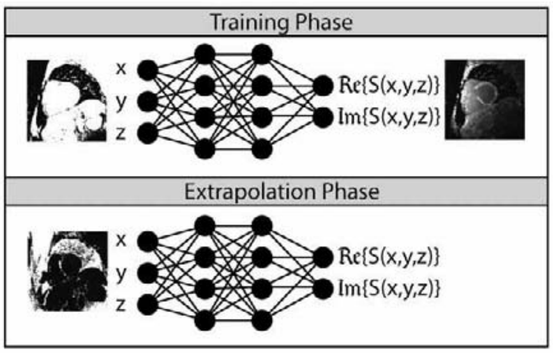

Figure 1: Schematic plot of the neural network method. In the training phase the regions with well known sensitivities are input to the network and it learns the connection between a coordinate and its sensitivity. In the extrapolation phase the "holes" are given as input and the network predicts the missing sensitivity values.

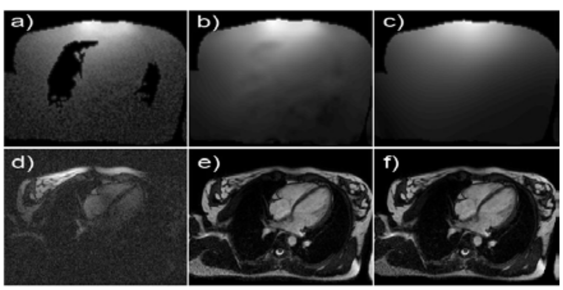

Figure 4: Reconstructed images with differently processed sensitivity maps using a 32 channel cardiac coil array. a) raw sensitivities from single coil (single-coil divided by body-coil), b) local polynomial fit, c) neural network fit, d) single coil image, e) reconstructed image using the sensitivity maps in b), f) reconstructed image using the sensitivity maps in c). Note that the polynomial fit in b) still shows some distortions in areas with low spin density (lung). The image quality of the reconstructed images shows no differences.

Methods

Neural networks are learning systems. Therefore processing of the sensitivity maps is performed in two phases. In the training phase, regions with well known sensitivities are presented to the network and it "learns" the relationship between a coordinate and its sensitivity. Since neural network learning can also be seen as a non-linear fit with a limited degree of freedom (given by the number of hidden neurons) additional smoothing and noise reduction takes place during the training phase if the number of hidden neurons has been chosen well. In the extrapolation phase the regions with unknown sensitivities ("holes") are presented to the network and it predicts the sensitivities in these regions based on the information trained in the learning phase (Figure 1). Simulations have shown that only a fraction of all existing voxels (randomly selected) is needed as input to "learn" sensitivities with sufficient accuracy. The network contains three input neurons, one for each coordinate of a pixel and two output neurons for the real and imaginary part of the coil sensitivity. Two hidden layers each consisting of 6 neurons were chosen. As training method the Bayesian-regularization-backpropagation algorithm was selected [2].

To evaluate the performance of the neural network a cine dataset of a four-chamber view was acquired using a 32 channel cardiac coil array in a healthy volunteer. Scan parameters were: TR=3.3ms, TE=1.6ms, flip angle=60°, scan matrix 256x256, FOV=350x350mm^2. Coil sensitivities data were obtained from a low resolution reference scan (TR=4.9ms, TE=1.1ms, flip angle=7°, scan matrix 128x128x20 FOV=350x350x150mm^3, 3 signal averages). To simulate parallel imaging, the fully sampled data set of the heart was undersampled in one direction by a factor of 2. Raw sensitivity maps were calculated from the low resolution reference scan. These data were processed further using the neural network and the polynomial fitting approach. The undersampled data set of the heart was then reconstructed by either using sensitivity maps determined with the neural network or by using maps processed with the polynomial fitting approach. All calculations were done in MATLAB (The MathWorks, Natick, MA, USA).

Results

Figure 2a displays the raw coil sensitivity of one coil element and the coil sensitivities after processing with the polynomial fitting approach (Figure 2b) and the neural network (Figure 1c). Relative to the sensitivity data obtained from polynomial fitting, less distortions of sensitivity result from neural network processing in particular in regions with low spin density. For calculating the sensitivity map of a single coil 900 randomly selected points (approximately 1/350 of all points) were sufficient for the neural network as input. For that reason the neural network fit and extrapolation was significantly faster than the pixelwise polynomial fitting. The computation time for the whole dataset was reduced by a factor of 12 from 61min using polynomial fitting to 5min using the neural network while no differences in image quality was observed in the images reconstructed from two-fold undersampled data.

Discussion

It has been shown that the neural network based approach allows fast fitting and extrapolation of coil sensitivity information. The neural network fitting could be an excellent method for applications where higher resolution sensitivity maps are needed or where sensitivities are directly extracted from the data. While the network proposed here provides good results in most cases further investigation has to be done regarding the network structure and parameters for a given dataset to improve performance.

References:

[1] Pruessmann K.P. et al., MRM **42**(5):952-62 (1999)

[2] MacKay, D. J. C., Neural Computation, vol. **4**(3):415-447, 1992.