GRAPPA Documentation

Vishwas Bharti, Rishabh Ramteke, Ameya Anjarlekar

July 2019

1 Pre-reading

We learnt about the basic principles behind the Magnetic Resonance Imaging(MRI) through youtube videos.We learnt about the phenomenon of **nuclear magnetism** in hydrogen.Scientists know that human body is 70 percent water and water contains hydrogen.This makes the diagnosis of human body(or anything that has water in it) makes possible.Through Magnetic Resonance Imaging we can highlight different parts of the body by just changing some parameters.We can change the Contrast Resolution and Spatial Resolution of our image.
We learnt that how the image is captured.It is based on the principle of **Magnetic Induction**.When the Nuclear Magnetisation Vector(the net vector that arises from small magnetic vectors around single protons) rotates in transverse direction of the main field it induces currents in the coils.This current can be analysed to know the nature of tissue.

2 Problem Statement explained

We need to reconstruct the image **(256X256)** through its kspace data obtained from **two** sources. The alternate rows except the ones at the middle of the kspace data are missing and need to be found out. This has been done using the GRAPPA algorithm as explained below.

3 Code explained

The code has been written in C++ language

It starts with extracting the input kspace data(with zeros at alternate rows). However, there are no spaces of zeros left in the middle rows

Hence, we need to find these missing data. This is done with the help of this algorithm.

Suppose A is the kspace matrix, then we assume that **if A[i][j] is a missing value it will be a linear combination of A[i+1][j], A[i-1][j], A[i+1][j+1], A[i+1][j-1], A[i-1][j+1], A[i-1][j-1] values corresponding to all the images available**. In our case we had 2 images available. So each value depends on **12 variables**

These values are computed using matrix formulae as mentioned in the matlab/c++ code

After the weight matrix is computed, in the next step we multiply it with respective values in matrix A to get the unknown values present in the kspace matrix

Sum of squares has been taken in k-space and no conversion of any array has been made in image space

4 Implementing on the chip

We used the ZyBo(Zync Board) to run our code on.Only the Processing System(PS) of ZyBo was used.We used Vivado 16.4 to build the hardware of our system and Xilinx SDK 16.4 to build and test the software of our system.

There was no problem in making the hardware and generating the bitstream file.We faced some problems in Xilinx SDK during debugging because of the lack of availability of example projects in C++ for Xilinx SDK.But it was solved later.

Each function is tested individually and code is compile-error free

There might be some differences in output values obtained through MATLAB and C++ as a bug during computing inverse was found in MATLAB. However, each function has been tested and verified manually in C++

If code doesn't work or some error occurs, try pressing the push button below the DONE LED before programming the chip.