Analysis of HYPR

Nasser M. Abbasi

California State University, Fullerton. Summer 2008 Compiled on January 30, 2024 at 6:36am

Contents

1 Introduction

In the context of reconstruction of MRI images from K-space data sets, the following are two desirable properties which are difficult to achieve simultaneously: High spatial resolution and High temporal resolution.

The first requires longer acquisition time while the second requires less time be consumed acquiring each image. Hence the inherent conflict in achieving both simultaneously. One possible remedy is to undersample image acquisition (along a radial or other trajectories such as Cartesian) which results in speed up of data acquisition, hence improving the temporal resolution. Next, an appropriate image reconstruction method is applied to the acquired data which attempts to compensate for some of the effects of the image undersampling.

Due to undersampling, streaking artifacts will be present in the final image. These streaking artifacts become more visible the larger the undersampling. Radial undersampling is a preferred method of acquisition compared to using other trajectories such as Cartesian: "*the aliasing artifacts from radial under-sampling usually appear as streaks, which are visually less distracting than the wrap-around artifacts obtained with Cartesian under-sampling.*"[**?**]

Mathematically, the problem of image reconstruction from undersampled K-space data is an inverse problem: "*Mathematically, the reconstruction problem from sparse K-space samples is an ill-posed inverse problem with infinitely many solutions*"[**?**]

Highly constrained backprojection reconstruction (HYPR) was introduced recently for the reconstruction of radially undersampled MRI images. HYPR is able to reconstruct these images with less visible artifacts while maintaining good SNR[**?**].

The method starts with the construction of one composite image made up from filtered backprojections of a large number of projections (each of these projections is the inverse Fourier transform of a corresponding radial lines from the K-space data set. This follows from the central slice theorem).

Since the composite image is made up of individual images collected over longer time period, it posses good spatial characteristics. In addition, its SNR is larger since a larger amount of images data is contained in it. However, the composite image temporal characteristics are poor since it combines images that were generated from varying time instances into a single image.

A weight image is then constructed from the ratio of a small number of unfiltered backprojections obtained from the original projections and from the composite image. Since the weight image is constructed from images that span a smaller time window than the case is with the composite image, the weight image posses good temporal characteristics. However, its spatial characteristic is poor due to the undersampling effect in the original data.

HYPR now generates a new image by multiplying the weight image with the composite image. This results in a HYPR image which combines the best characteristics found in the composite image and in the weight image, resulting in an image with good SNR, good temporal and spatial characteristics and with limited artifacts. The above process is repeated for the next HYPR image reconstruction until all the K-space data set is processed. The relationship between the weight image and the composite image and the resulting HYPR image is summarized in the following diagram.

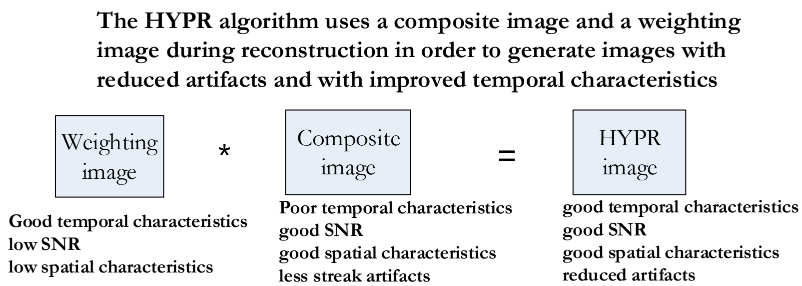

Figure 1: Summary of HYPR

2 Terminologies

Let I be a 2D image. Let $R_{\theta(t)}$ be the forward projection matrix (implemented as Radon transformation in practice) which when applied to the image I at some projection angle θ at time t will result in a 1D projection image $P(t)$.

Since the image itself will change with time (blood flows, etc...) therefore we need to associated a time index t as well with the 2D image itself. Hence we write $I(t)$ from now on, to indicate the 2D image at time t. Notice that the image as a whole does not move (relative to fixed initerial frame of reference) but the image content can change as described above.

To avoid confusion in what follows, I will use the notation of $\circledast$ to indicate a multiplication between 2 matrices *element by element* and will use $*$ to indicate the normal matrix by matrix or matrix by vector multiplication. And will use /. to indicate division between vectors or matrices being done element by element.

3 Introduction

We will analyze the following HYPR algorithms: Original HYPR, Wright-Hyper, I-HYPR and a new variation of I-HYPR based on on the Wright-HYPR. This new variation is different from I-HYPR since in I-HYPR the iteration is made on a composite image based on the Original HYPR construction while in this variation, the composite image is constructed is with the Write-HYPR algorithm. We will call this variation IW-HYPR.

For each algorithm we show the schematic flow chart and the mathematical description based on matrix formulation and an algorithm for the implementation. At the end, we will run a simulation of each of the above 4 algorithm from the same set of images and attempt to describe the finding and compare the results.

In the mathematical formula we derive an expression of the HYPR image as a function of the set of original images $I(t)$ and the set of the forward projection matrices $R_{\theta(t)}$.

Before we discuss the Mathematical formulation, we need to better undertstand how to generate a set of projections from a well defined image which we can express mathematically. For this purpose the following diagram shows the 3 possible cases in which projections can occure at.

We need to generate an analytical expression as a function of time of a simple object shape for each of the following 3 cases.

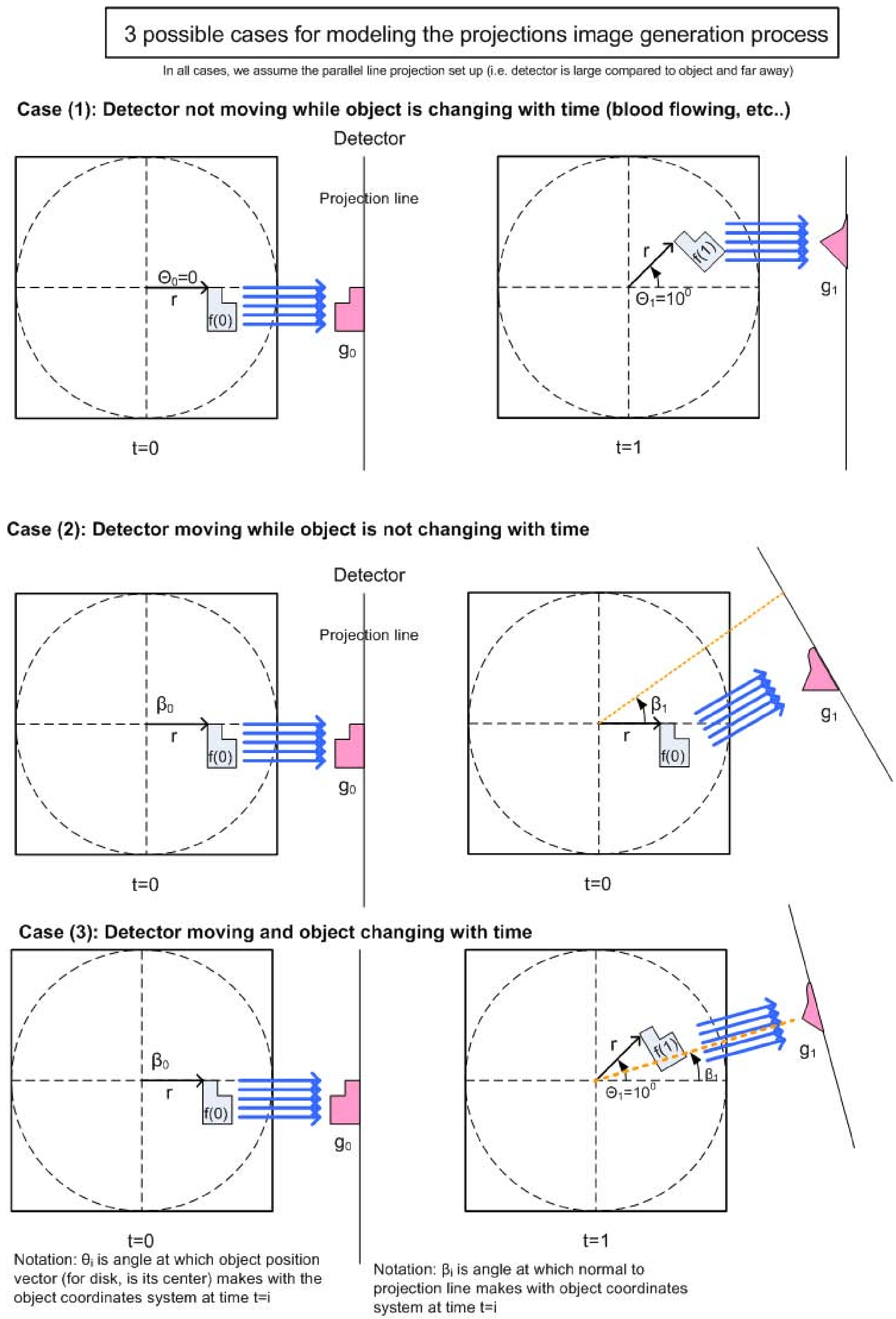

Figure 2: moving object

4 Review of HYPR

Given a set of projections $P(t)$ over a number of time frames say M and assuming there are N projections generated per a time frame, the data set will consists of $M * N$ projections in total. Within one time frame, a number of projections can be collected. These projections within each time frame will generate one HYPR frame using the HYPR algorithm. A time frame can have only one projection, but normally a time frame will have much larger number of projections.

For example, assuming there are $M = 10$ time frames, and $N = 20$ projections generated by time frame, then the total number of projections is 200. In this case, we will obtain 10 HYPR frame images at the end.

HYPR starts by constructing a composite image from all the projections in the data set (200 in the above example). This is done by computing the filtered backprojection of all the $M * N$ projections into one image called C.

Next, we process the projections from each time frame. For each time frame we generate a set of N projections from C by perform a radon tranform (forward projection) on C at the same angle corresponding to the projection being processed. This will generate set of projections called the P_c projections. Hence there will be N such projections per each time time.

Next the ratio of the each P projection over the corresponding P_c projection is found. This ratio is done pixel by pixel. Then each such ratio is multiplied by the composite image C generating a a new set of size $M * N$ of weighted composite images. Now to generate a HYPR frame image, the average of these weighted compsite images is taken. The average is carried over each time frame at a time. Hence the first N weighted compsite images will generate one HYPR frame, and the next N weighted compsite images will generate the next HYPR frame and so on, resulting in M HYPR frames.

4.1 High level schematic of original HYPR with many projections per time frame

The following diagram illustrates the above where we used the original HYPR algorithm. Later on, we describe in detailes the different HYPR algorithms variations. In the following diagram we show 4 projections per time frame and a total of 3 time frames. This results in 3 HYPR frames.

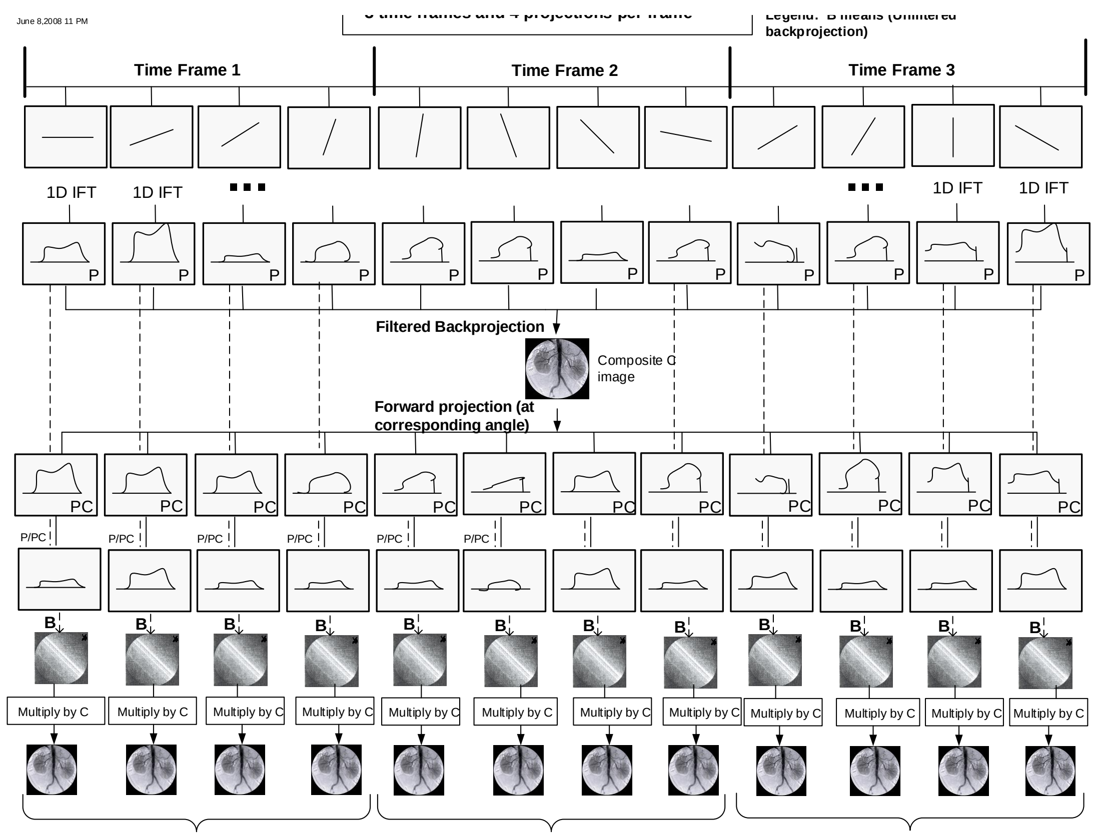

Figure 3: original HYPR simplified multi frame

5 HYPR mathematical formulation

5.1 Original HYPR

The projection $P(t)$ is obtained by applying forward projection on the image $I(t)$, Hence we write

$$P(t) = R_{\theta(t)}[I(t)]$$

Next, the set of $\{P(t)\}$ are combined and a filtered backprojection is applied to the result to generate a composite image C

$$C = \mathbf{FBP}\left[\sum_{i=1}^{M*N} R_{\theta(t_i)}[I(t_i)]\right]$$

Where **FBP** is operator for the filtered backprojection.

The composite image C can be written as

$$\begin{aligned} C &= \sum_{i=1}^{N} A_i^{+} * P_i \\ &= \sum_{i=1}^{N} A_i^{+} * (A_i * f_i) \end{aligned}$$

Where N in the number of projections. Now applying forward projection to C at angle θ_i will generate a projection P_{c_i},Hence

$$P_{c_i} = A_i * C$$

Let z_i be the ratio $P_i/.P_{c_i}$ where this division is being carried out element by element between the two projections. Hence

$$\begin{aligned} z_i &= P_i/.P_{c_i} \\ &= (A_i * f_i)\ /.\ (A_i * C) \\ &= \left(A_i^T * f_i\right)\ /.\ \left(A_i * \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right) \end{aligned}$$

Now apply unfiltered backprojection on the above projection ratio to obtain an unfiltered 2D image and then mutiply that with the composit image to obtain a HYPR frame F_i

$$\begin{aligned} F_i &= C \circledast \left(A_i^T * z_i\right) \\ &= \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right) \circledast \left(A_i^T * \left[\left(A_i^T * f_i\right)\ /.\ \left(A_i * \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right)\right]\right) \end{aligned}$$

Hence HYPR image is

$$\begin{aligned} I &= \frac{1}{N}\sum_{i=1}^{N} F_i \\ &= \frac{1}{N}\sum_{i=1}^{N}\left\{\left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right) \circledast \left(A_i^T * \left[\left(A_i^T * f_i\right)\ /.\ \left(A_i * \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right)\right]\right)\right\} \end{aligned} \tag{1}$$

We see that the above expression for HYPR image depends only on f_i and A_i.

Therefore, given a set of images f_i and a set of projection angles θ_i we can compute the forward projection matrices A_i (analytically we can do this for simple shapes such as a disk rotating around a unit circle for example). And once the set of matrices A_i is computed, equation (1) could then be computed to obtain a HYPR image.

We can then generate the same HYPR images by performing backprojection (filtered an unfiltered) using the Fourier transform method. The algorithm for the backprojection (filtered) is known and given on page 62 of Kak and Slaney book which I will post a scan of. Or we could simply use the radon/iradon for the implementaion of A_i,A_i^T, A_i^+, where A_i corresponds to applying radon on image f at angle θ_i, and A_i^T corresponds to applying iradon on $g_i$with filter 'none' and A_i^+corresponds to applying iradon on projection g_i with a specified filter.

5.1.1 Schematic diagram of original HYPR algorithm

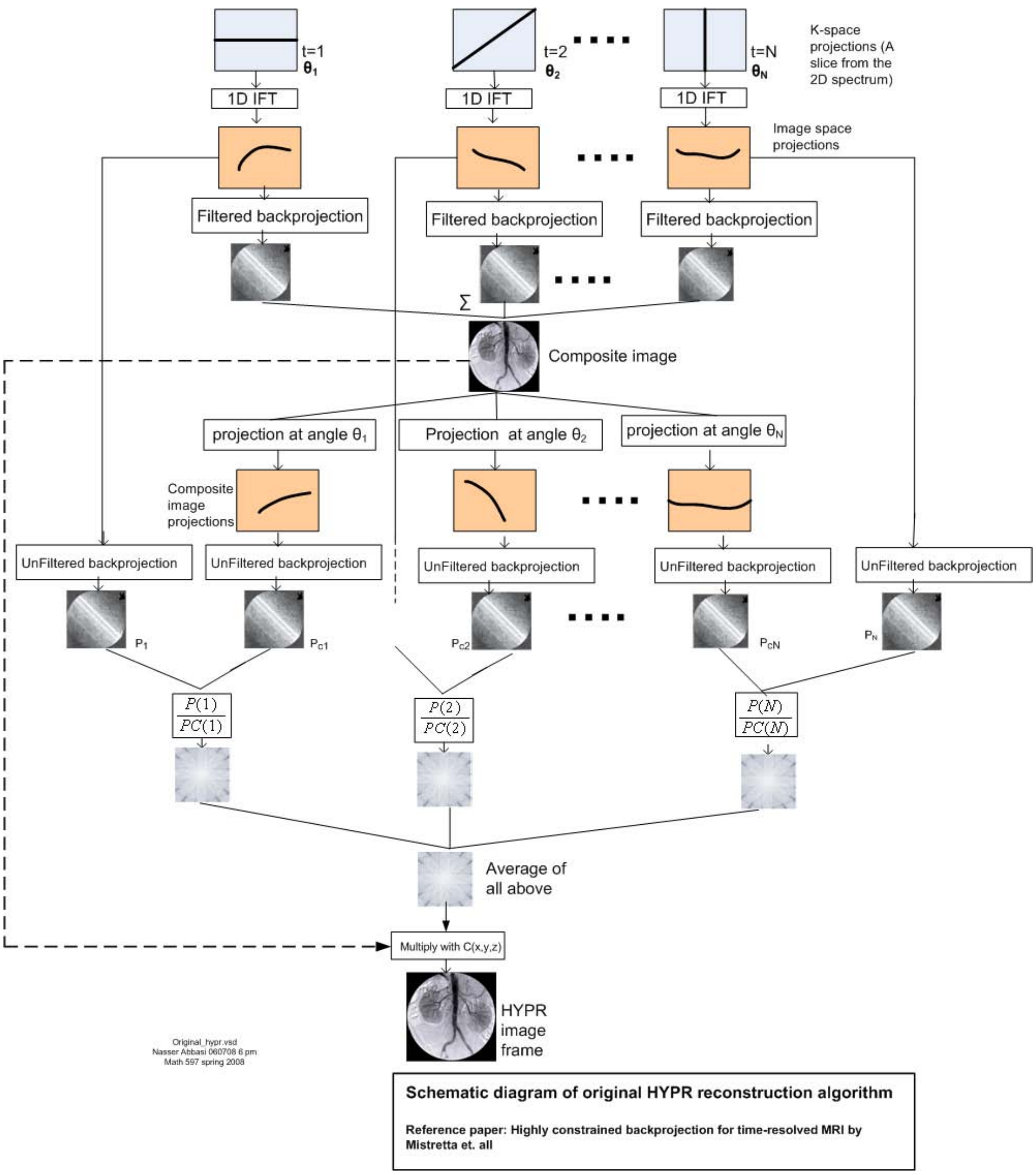

Figure 4: original HYPR

5.2 Wright HYPR

The projection g_i is obtained by applying forward projection on the image f at time i.Hence we write

$$g_i = A_i * f_i$$

Where in the above the 2D image f_i needs to be first converted to a 1D vector as was done in the first assignment by folding the 2D image into a 1D column vector.

Let A_i^+ be an psudoinverse of A_i which performs a filtered backprojection when applied on a projection vector g_i and let A_i^T be the transpose of the matrix A_i which performs an unfiltered backprojection on g_i.

The composite image C can be written as

$$\begin{aligned} C &= \sum_{i=1}^{N} A_i^+ * g_i \\ &= \sum_{i=1}^{N} A_i^+ * (A_i * f_i) \end{aligned}$$

where N in the number of projections. Now, each unfiltered backprojection is

$$\begin{aligned} P_i &= A_i^T * g_i \\ &= A_i^T * (A_i * f_i) \end{aligned}$$

Now applying forward projection to C at angle θ_i will generate a projection, call it r_i. Hence

$$r_i = A_i * C$$

Now applying an unfiltered backprojection on r_i will result in P_{c_i} hence

$$\begin{aligned} P_{c_i} &= A_i^T * r_i \\ &= A_i^T * (A_i * C) \end{aligned}$$

Let z_i be the ratio $P_i /. P_{c_i}$ where this division is being carried out element by element between the two images. Therefore

$$\begin{aligned} z_i &= P_i /. P_{c_i} \\ &= \left(A_i^T * (A_i * f_i)\right) \;/.\; \left(A_i^T * (A_i * C)\right) \\ &= \left(A_i^T * (A_i * f_i)\right) \;/.\; \left(A_i^T * \left(A_i * \left(\sum_{k=1}^{N} A_k^+ * (A_k * f_k)\right)\right)\right) \end{aligned}$$

Hence one HYPR frame i is

$$
\begin{aligned}
F_i &= z_i \circledast C \\
&= \left(A_i^T * (A_i * f_i)\right) \ /. \ \left(A_i^T * \left(A_i * \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right)\right) \circledast \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)
\end{aligned}
$$

Hence HYPR image is

$$
\begin{aligned}
I &= \frac{1}{N}\sum_{i=1}^{N} F_i \\
&= \frac{1}{N}\sum_{i=1}^{N}\left\{\left(A_i^T * (A_i * f_i)\right) \ /. \ \left(A_i^T * \left(A_i * \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right)\right) \circledast \left(\sum_{k=1}^{N} A_k^{+} * (A_k * f_k)\right)\right\}
\end{aligned}
\tag{1}
$$

5.2.1 Schematic diagram of Wright HYPR algorithm

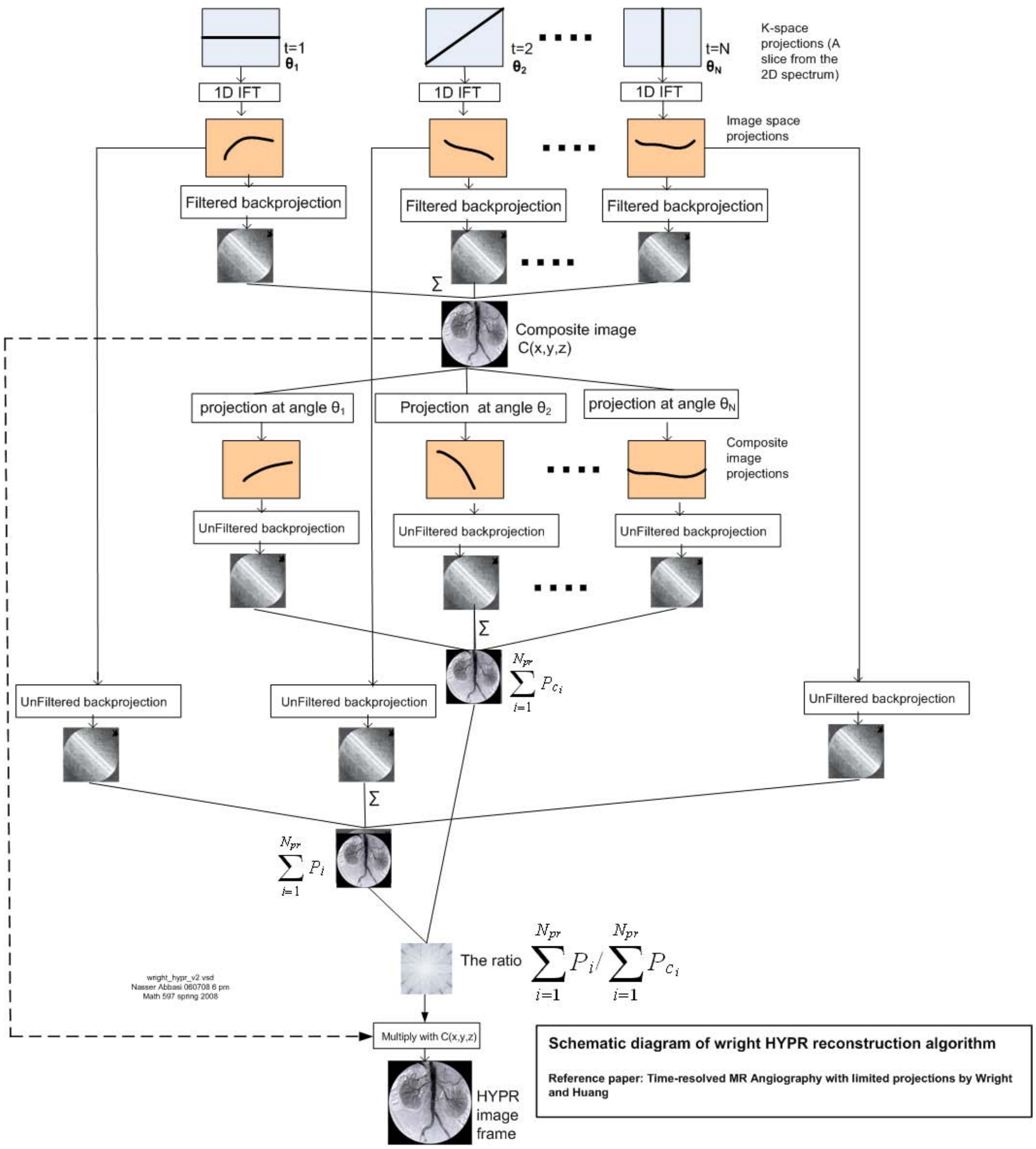

Figure 5: Wright hypr

5.2.2 On the difference between original HYPR and Wright-HYPR

The difference between the original HYPR and Wright-HYPR can be seen in the following simplified diagram. We see than in the original HYPR, the ratio P/PC is performed on the 1-D projections, then the unfiltered backprojection is applied on the resulting 1-D set of images. In the Wright-HYPR algorithm, the unfiltered backprojection is first applied to the set of the 1-D projections and then the ratio is performed on the result 2-D set of images.

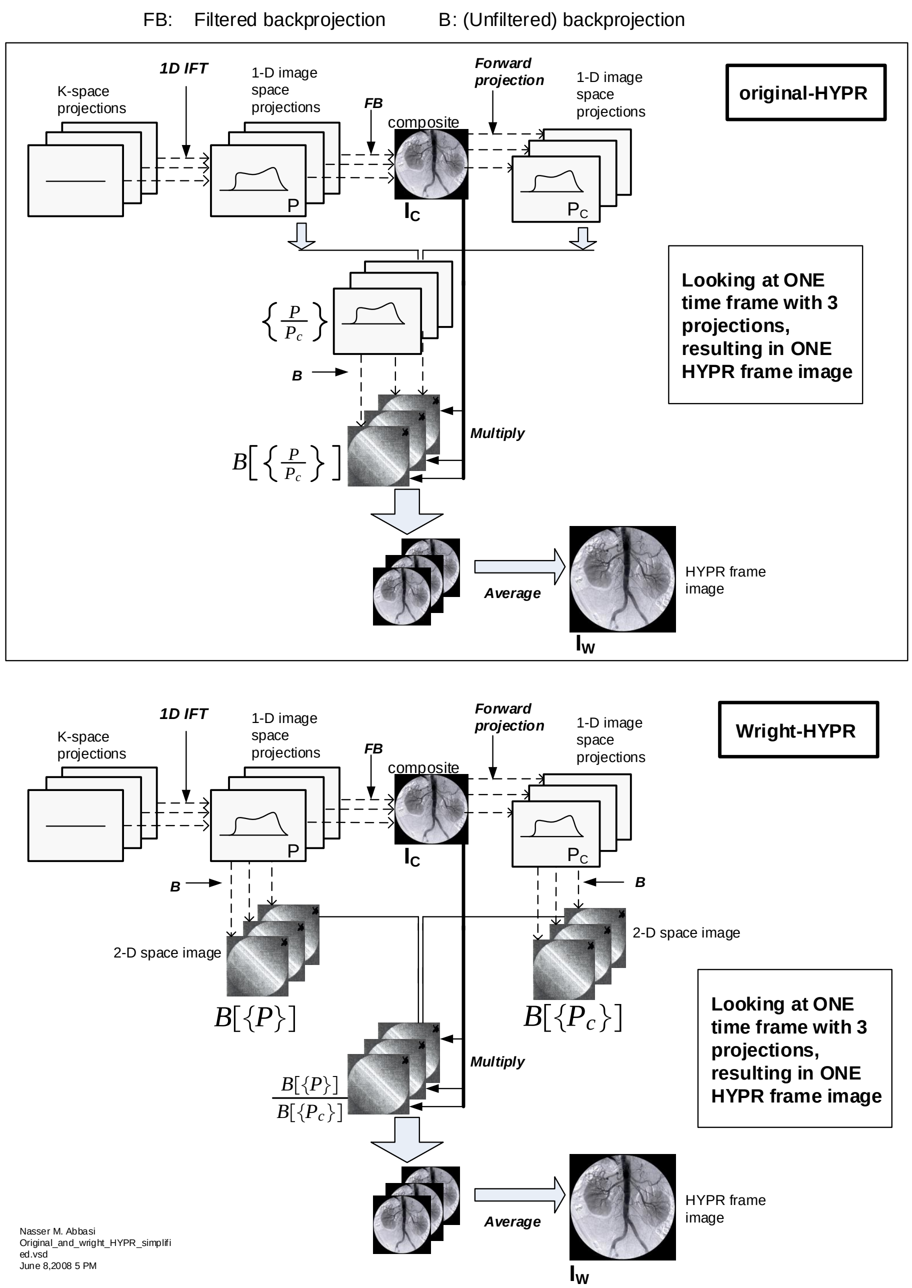

Figure 6: original and wright HYPR simplified

5.3 I-HYPR

This is an iterative method where the composite image itself is updated and a new HYPR image determined with the hope of obtaining a better HYPR image (how to determine how many iterations? need to read more on this) as more iterations are performed. Let the number of iterations by M therefore this algorithm will generate $C_1, C_2, \cdots, C_M$ composite images.

This is the mathematical formulation for I-HYPR

The projection P_i is obtained by applying forward projection on the image f at time i.In the original HYPR, g_i is what is refered to P_i projection. Therefore

$$P_i = A_i * f_i$$

The composite image C at iteration (1) can be written as

$$C_1 = \sum_{i=1}^{N} A_i^{+} * P_i$$

Where N in the number of projections. Now applying forward projection to C at angle θ_i will generate a projection P_{c_i}. Hence

$$P_{c(1)_i} = A_i * C_1$$

Let z_i be the ratio $P_i/.P_{c_i}$ where this division is being carried out element by element between the two projections. Hence

$$\begin{aligned} z_i &= P_i/.P_{c(1)_i} \\ &= P_i \ /. \ (A_i * C_1) \end{aligned}$$

Now apply unfiltered backprojection on the above projection ratio to obtain an unfiltered 2D image and then mutiply that with the composit image to obtain a HYPR frame F_i

$$\begin{aligned} F_i &= C_1 \circledast \left(A_i^T * z_i\right) \\ &= C_1 \circledast \left(A_i^T * [P_i \ /. \ (A_i * C_1)]\right) \end{aligned}$$

Hence HYPR image at interation (1) is

$$\begin{aligned} I_1 &= \frac{1}{N}\sum_{i=1}^{N} F_i \\ &= \frac{1}{N}\sum_{i=1}^{N} C_1 \circledast \left(A_i^T * [P_i \ /. \ (A_i * C_1)]\right) \end{aligned} \tag{1}$$

Now use I_1 as the composite image for the next iteration, we obtain

$$C_2 = I_1$$

Hence

$$P_{c(2)_i} = A_i * C_2$$

And

$$\begin{aligned} z_i &= P_i/.P_{c(2)_i} \\ &= P_i \;/.\; (A_i * C_2) \end{aligned}$$

Now apply unfiltered backprojection on the above projection ratio to obtain an unfiltered 2D image and then mutiply that with the composit image to obtain a HYPR frame F_i

$$\begin{aligned} F_i &= C_2 \circledast \left(A_i^T * z_i\right) \\ &= C_2 \circledast \left(A_i^T * [P_i \;/.\; (A_i * C_2)]\right) \end{aligned}$$

Hence HYPR image at interation (2) is

$$\begin{aligned} I_2 &= \frac{1}{N}\sum_{i=1}^{N} F_i \\ &= \frac{1}{N}\sum_{i=1}^{N} C_2 \circledast \left(A_i^T * [P_i \;/.\; (A_i * C_2)]\right) \end{aligned} \tag{1}$$

But $C_2 = I_1 = \frac{1}{N}\sum_{i=1}^{N} C_1 \circledast \left(A_i^T * [P_i \;/.\; (A_i * C_1)]\right)$ hence the above becomes

$$\begin{aligned} I_2 = \frac{1}{N}\sum_{i=1}^{N} &\left(\frac{1}{N}\sum_{i=1}^{N} C_1 \circledast \left(A_i^T * [P_i \;/.\; (A_i * C_1)]\right)\right) \circledast \\ &\left(A_i^T * \left[P_i \;/.\; \left(A_i * \left(\frac{1}{N}\sum_{i=1}^{N} C_1 \circledast \left(A_i^T * [P_i \;/.\; (A_i * C_1)]\right)\right)\right)\right]\right) \end{aligned}$$

But $C_1 = \sum_{i=1}^{N} A_i^+ * P_i$, hence the above becomes

$$\begin{aligned} I_2 = \frac{1}{N}\sum_{i=1}^{N} &\left(\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{i=1}^{N} A_i^+ * P_i\right) \circledast \left(A_i^T * \left[P_i \;/.\left(A_i * \left(\sum_{i=1}^{N} A_i^+ * P_i\right)\right)\right]\right)\right) \circledast \\ &\left(A_i^T * \left[P_i \;/.\; \left(A_i * \left(\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{i=1}^{N} A_i^+ * P_i\right) \circledast \left(A_i^T * \left[P_i \;/.\; \left(A_i * \left(\sum_{i=1}^{N} A_i^+ * P_i\right)\right)\right]\right)\right)\right)\right]\right) \end{aligned}$$

Repeate this processes by setting $C_3 = I_2$ and generate I_3. Continue untill I_M where M is number of iterations or untill some other convergence criteria is achived.

5.3.1 Schematic diagram of I-HYPR algorithm

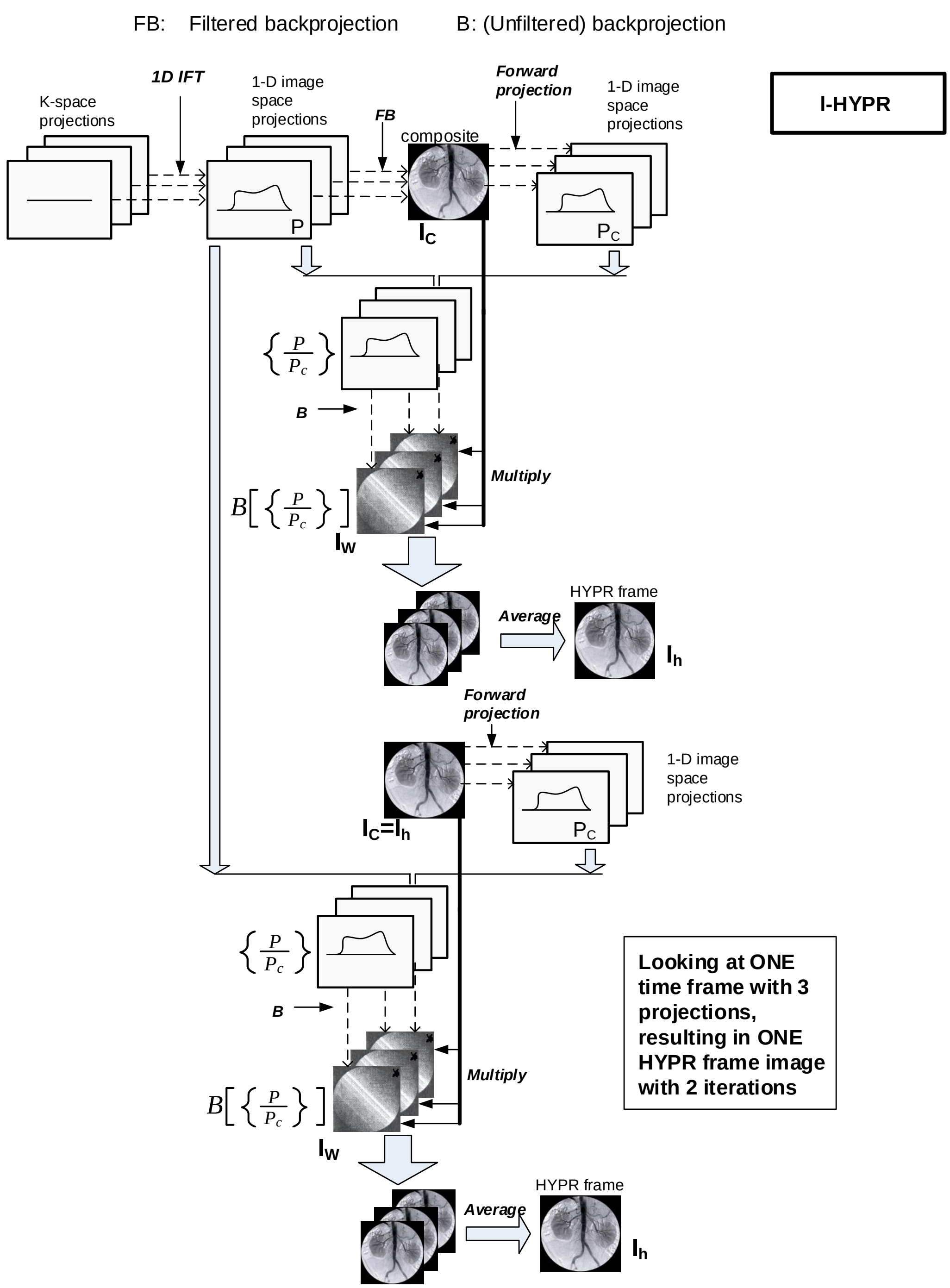

Figure 7: I HYPR simplified

Simplied version of the schematic diagram

6 WI-HYPR (Iiterative HYPR based on Wright-HYPR flow)

Need to finish the mathematics of this version. (TODO)

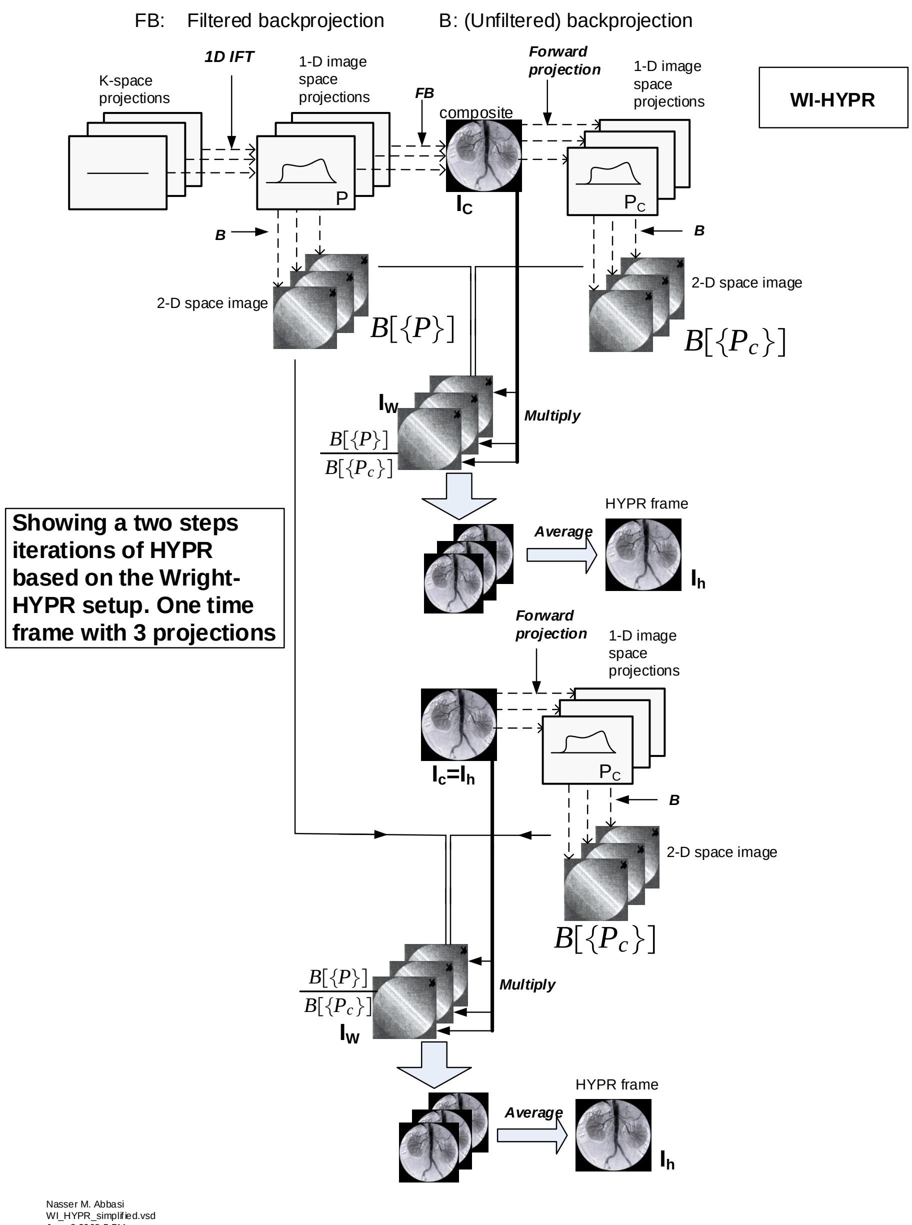

Figure 8: WI HYPR simplified