

my working notes on HYPR for Math 597

Nasser M. Abbasi

California State University, Fullerton. Summer 2008

Compiled on July 9, 2025 at 6:24am

1 Notations and definitions

1. MLEM Maximum-Likelihood Expectation-Maximization
2. PET Positron Emission Tomography
3. SPECT Single-Photon Emission Computed Tomography
4. I A 2-D image. This represent the original user image at which the HYPR algorithm is applied to.
5. I_t When the original image content changes during the process, we add a subscript to indicate the image I at time instance t .
6. R radon transform.
7. R_ϕ radon transform used at a projection angle ϕ .
8. ϕ_t When the projection angle ϕ is not constant but changes with time during the MRI acquisition process, we add a subscript to indicate the angle at time instance t .
9. R_{ϕ_t} radon transform used at an angle ϕ_t .
10. $s = R_\phi[I]$. radon transform applied to an image I at angle ϕ . This results in a projection vector s .
11. H Forward projection matrix. The Matrix equivalent to the radon transform R .
12. θ Estimate of an image I .
13. $H\theta$ Multiply the forward projection matrix H with an image estimate θ .
14. $g = H\theta$ Multiply the forward projection matrix H with an image estimate θ to obtain a projection vector g .
15. $R_\phi^u[s]$ The inverse radon transform applied in unfiltered mode to a projection s which was taken at angle ϕ . This results in a 2D image.

16. $R_\phi^f[s]$ The inverse radon transform applied in filtered mode to a projection s which was taken at angle ϕ . This results in a 2D image.
17. $H^T g$ The transpose of the forward projection matrix H multiplied by the projection vector g . This is the matrix equivalent of applying the inverse radon transform in an unfiltered mode to a projection s (see item 12 above).
18. $H^+ g$ The pseudo inverse of the forward projection matrix H being multiplied by the projection vector g . This is the matrix equivalent of applying the inverse radon transform in filtered mode to a projection s (see item 13 above).
19. C Composite image generated by summing all the filtered back projections from projections s_t of the original images I_t . Hence $C = \sum_{i=1}^N R_{\phi_{t_i}}^f [s_{t_i}]$
20. P_t The unfiltered backprojection 2D image as a result of applying $R_{\phi_t}^u [s_t]$ where s_t is projection from user image I_t taken at angle ϕ_t .
21. P_{c_t} The unfiltered backprojection 2D image as a result of applying $R_{\phi_t}^u [s_t]$ where s_t is projection from the composite image C taken at angle ϕ_t .
22. N_p Number of projections used to generate one HYPR frame image. This is the same as the number of projections per one time frame.
23. N The total number of projections used. This is the number of time frames multiplied by N_p
24. J_k The k^{th} HYPR frame image. A 2-D image generate at the end of the HYPR algorithm. There will be as many HYPR frame images J_k as there are time frames.
25. Image fidelity: " (inferred by the ability to discriminate between two images)" reference: The relationship between image fidelity and image quality by Silverstein, D.A.; Farrell, J.E

Sci-Tech Encyclopedia: Fidelity

"The degree to which the output of a system accurately reproduces the essential characteristics of its input signal. Thus, high fidelity in a sound system means that the reproduced sound is virtually indistinguishable from that picked up by the microphones in the recording or broadcasting studio. Similarly, a television system has a high fidelity when the picture seen on the screen of a receiver corresponds in essential respects to that picked up by the television camera. Fidelity is achieved by designing each part of a system to have minimum distortion, so that the waveform of the signal is unchanged as it travels through the system. "

26. "image quality (inferred by the preference for one image over another)". Same reference as above

27. TE (Echo Time) "represents the time in milliseconds between the application of the 90° "pulse and the peak of the echo signal in Spin Echo and Inversion Recovery pulse sequences". reference: <http://www.fonar.com/glossary.htm>
28. TR (Repetition Time) "the amount of time that exists between successive pulse sequences applied to the same slice." reference: <http://www.fonar.com/glossary.htm>



Figure 1: MRI picture

We will analyze the following HYPR algorithms:

1. Original HYPR
2. Wright-Hyper
3. I-HYPR

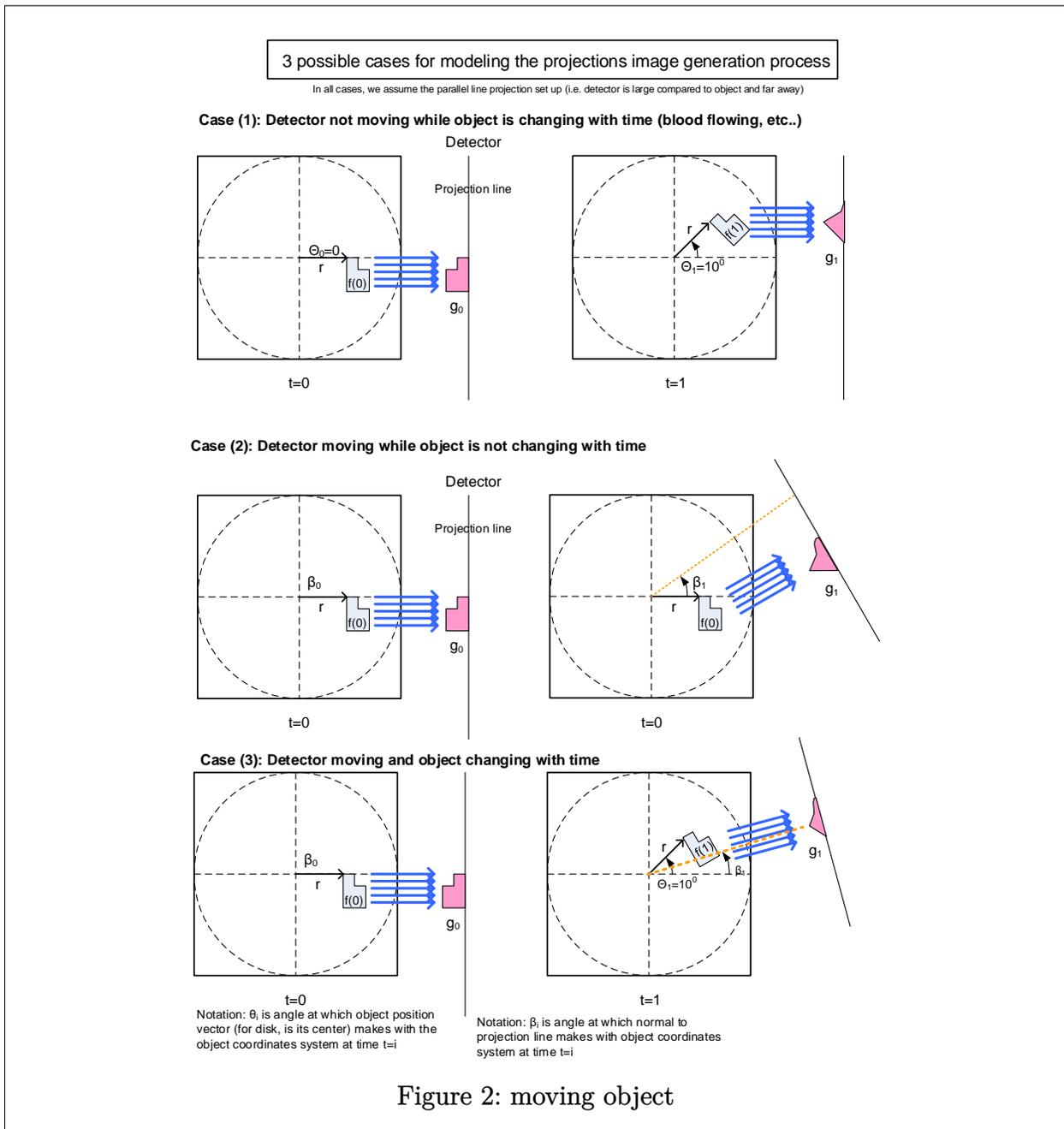
For each algorithm we show the schematic flow chart and the mathematical description. At the end, we will run a simulation of each of the above 3 algorithm from the same set of images to describe the findings and compare the results.

The description of the simulation software and the results found are shown below in this report.

Before we discuss the Mathematical formulation, we need to better understand how to generate a set of projections from a well defined image which we can express mathematically.

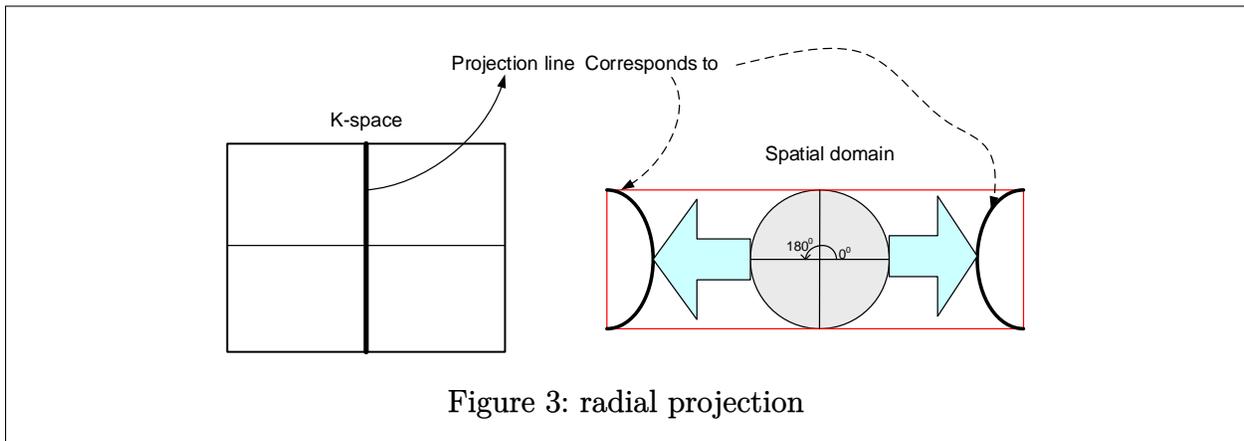
For this purpose the following diagram shows the 3 possible cases in which projections can occur at.

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

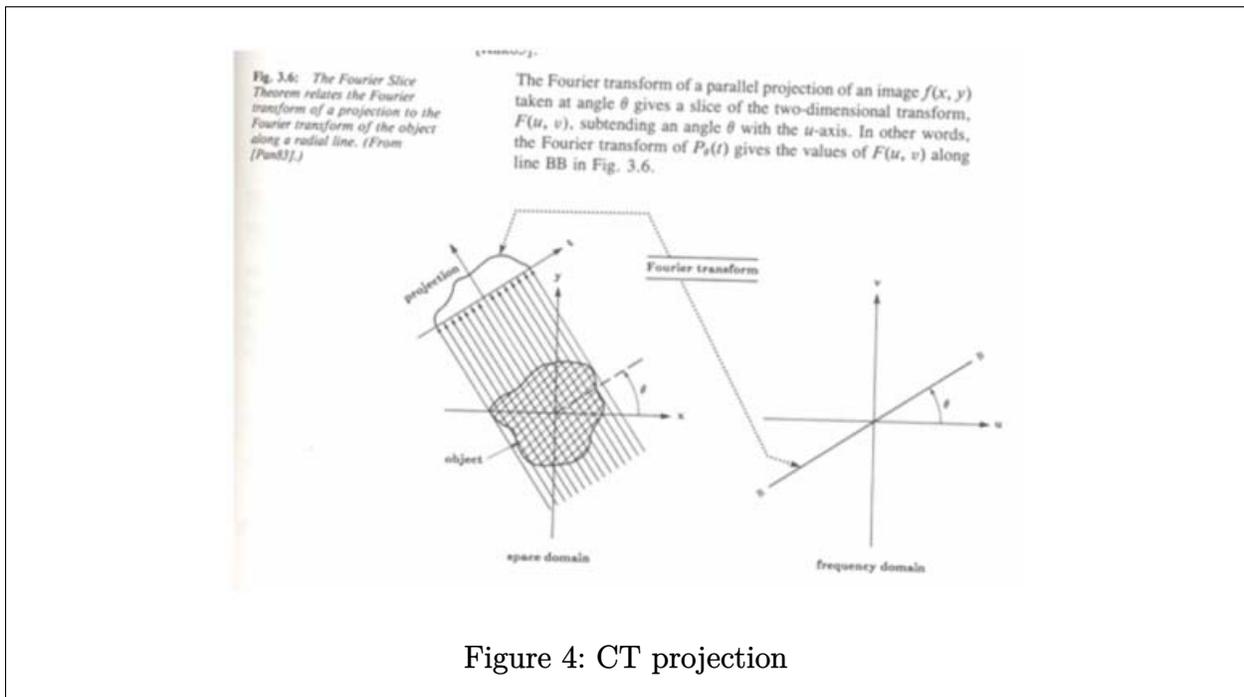


1.1 Clarification of projections

In MRI, projection line refers to a line through the center of the k-space (a radial line). In the spatial domain one k-space projection line will be mapped to 2 projections (right and left projections). To clarify, the following diagram shows a disk at the center and the corresponding horizontal spatial projections. The projections are across the image and hence will generate a projection to the right and one to the left.



In CT scans however, the source is in one position and the detector is on another, hence we have forward projections only, not across the image as above. This is shown in the Kak and Slaney book



What all the above means, is that the Write-Huang paper, when they mention 8 projections, they mean 16 projections in the CT-sense. I.e. forward and backward projections. The simulation software written uses the CT projection convention. Hence use 16 projections to produce the same result as the Wright-Huang paper which uses 8 projections.

2 HYPR mathematical formulation

2.1 Original HYPR

This mathematics of this algorithm will be presented by using the radon transform R notation and not the matrix projection matrix H notation.

The projection s_t is obtained by applying radon transform R on the image I_t at some angle ϕ_t

$$s_t = R_{\phi_t}[I_t]$$

When the original object image does not change with time then we can drop the subscript t from I_t and just write $s_t = R_{\phi_t}[I]$

The composite image C is found from the filtered back projection applied to all the s_t

$$C = \sum_{i=1}^N R_{\phi_{t_i}}^f [s_{t_i}]$$

Notice that the sum above is taken over N and not over N . Next a projection s_{c_t} is taken from C at angle ϕ as follows

$$s_{c_t} = R_{\phi_t}[C]$$

The the unfiltered back projection 2-D image P_t is generated

$$P_t = R_{\phi_t}^u [s_t]$$

And the unfiltered back projection 2-D image P_{c_t} is found

$$P_{c_t} = R_{\phi_t}^u [s_{c_t}]$$

Then the ratio of $\frac{P_t}{P_{c_t}}$ is summed and averaged over the time frame and multiplied by C to generate a HYPR frame J for the time frame. Hence for the k^{th} time frame we obtain

$$\begin{aligned} J_k &= C \left(\frac{1}{N_p} \sum_{i=1}^{N_p} \frac{P_{t_i}}{P_{c_{t_i}}} \right) \\ &= \frac{1}{N_p} \left(\sum_{i=1}^N R_{\phi_{t_i}}^f [s_{t_i}] \right) \sum_{j=1}^{N_p} \frac{R_{\phi_{t_j}}^u [s_{t_j}]}{R_{\phi_{t_j}}^u [s_{c_{t_j}}]} \end{aligned}$$

2.1.1 Schematic diagram of original HYPR algorithm

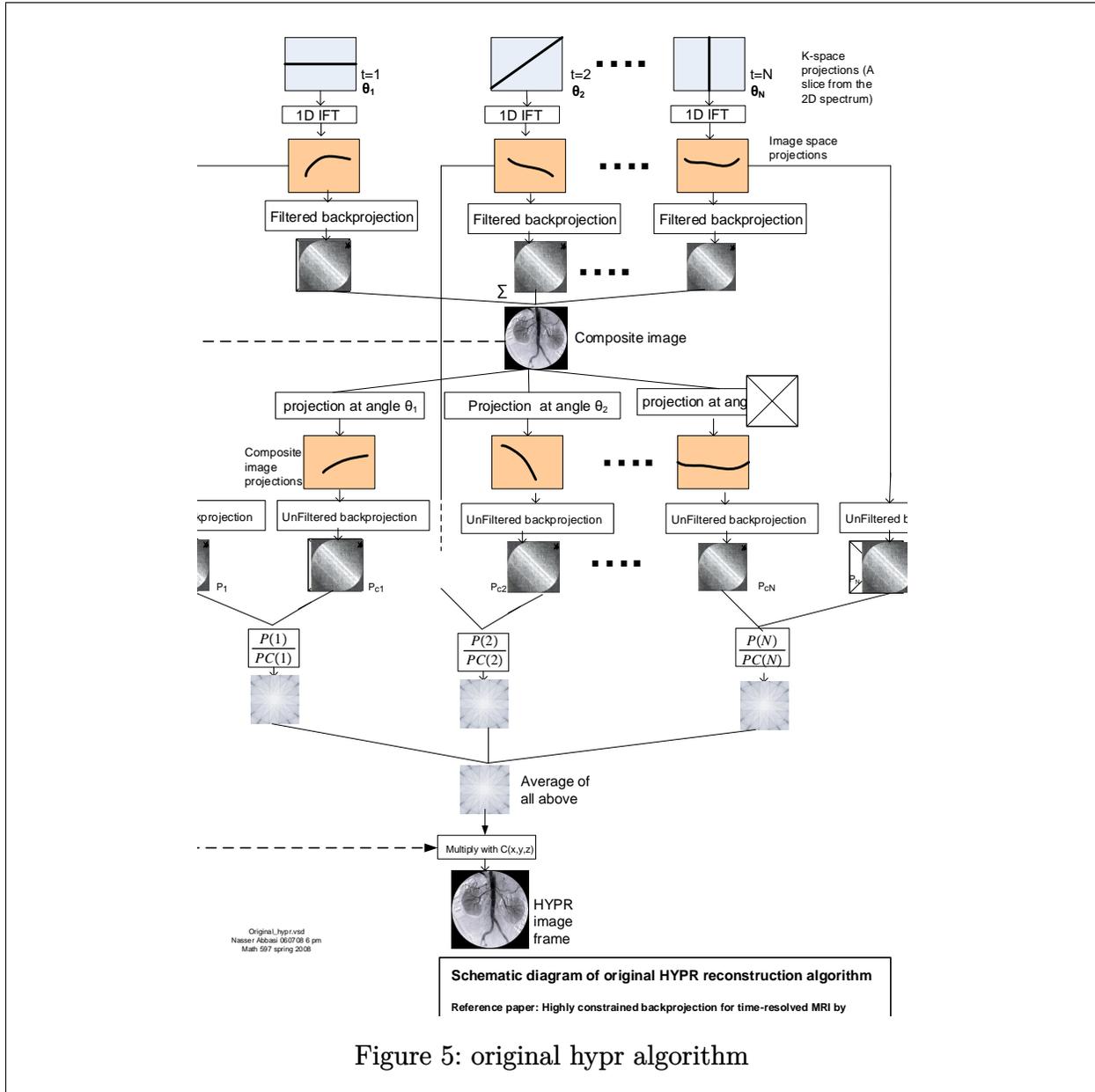


Figure 5: original hypr algorithm

2.1.2 Original HYPR algorithm

The following is the algorithm for original HYPR written in high level pseudo-code. For our implementation we will have the input to the HYPR algorithm as a set of original images I_i . In other words, we will not start from the frequency domain, but from the spatial domain directly.

INPUT: N total number of projections
 M number of time frames

```

        set of 2D images {I}. The set size should be of size N*M
        Theta_Initial,Theta_End -- initial and final angle (limited view)
OUTPUT:  M number of HYPR 2D images

-- divide angles from Theta_Initial to Theta_Final into N*M equal divisions
Theta = linspace(Theta_Initial,Theta_Final,M*N)

FOR i from 1 to N LOOP
    g(i) = radon(I(i),Theta(i)) -- generate forward projections
    Filtered_Back_Projection(i) = iradon(g(i), Theta(i))
END LOOP

C = SUM( Filtered_Back_Projection ) -- composite image
Nr = N/M -- Number of projections per time frame

Starting_Projection = 1
FOR i from 1 to M LOOP
    J(i) = Original_HYPR( C, g( Starting_Projection:Starting_Projection+Nr-1),
                        Theta( Starting_Projection:Starting_Projection+Nr-1) )
    Starting_Projection = Starting_Projection + Nr
END LOOP

--
-- Original HYPR function
--
FUNCTION Original_HYPR( C, g, Theta) RETURNS HYPR_IMAGE
    Nr = Length(g) --- Number of projections Per Frame

    FOR i from 1 to Nr
        gc(i) = radon(C, Theta(i))
        PC(i) = iradon( gc(i), Theta(i), FILTER='NONE')
        P(i) = iradon( g(i), Theta(i), FILTER='NONE')
        Z(i) = P(i)/PC(i) -- element by element division
    END LOOP

    mask = SUM(z)/Nr
    RETURN( mask * C ) -- Element by element multiplication

END FUNCTION Original_HYPR

```

2.2 Wright HYPR

This mathematics of this algorithm will be presented by using the radon transform R notation and not the matrix projection matrix H notation. The conversion between the notation can be easily made by referring to the notation page at the end of this report.

The projection s_t is obtained by applying radon transform R on the image I_t at some angle ϕ_t

$$s_t = R_{\phi_t}[I_t]$$

When the original object image does not change with time then we can drop the subscript t from I_t and just write $s_t = R_{\phi_t}[I]$

The composite image C is found from the filtered back projection applied to all the s_t

$$C = \sum_{i=1}^N R_{\phi_{t_i}}^f [s_{t_i}]$$

Notice that the sum above is taken over N and not over N . Next a projection s_c is taken from C at angle ϕ as follows

$$s_{c_t} = R_{\phi_t}[C]$$

The the unfiltered back projection 2-D image P_t is generated

$$P_t = R_{\phi_t}^u [s_t]$$

And the unfiltered back projection 2-D image P_{c_t} is found

$$P_{c_t} = R_{\phi_t}^u [s_{c_t}]$$

Now the set of P_t and P_{c_t} over one time frame are summed the their ratio multiplied by C to obtain the k^{th} HYPR frame

$$\begin{aligned} J_k &= C \frac{\sum_{i=1}^{N_p} P_{t_i}}{N_p} \\ &= C \frac{\sum_{i=1}^{N_{pr}} R_{\phi_t}^u [s_t]}{N_{pr}} \\ &= C \frac{\sum_{i=1}^{N_{pr}} R_{\phi_t}^u [s_{c_t}]}{N_{pr}} \end{aligned}$$

2.2.1 Schematic diagram of Wright HYPR algorithm

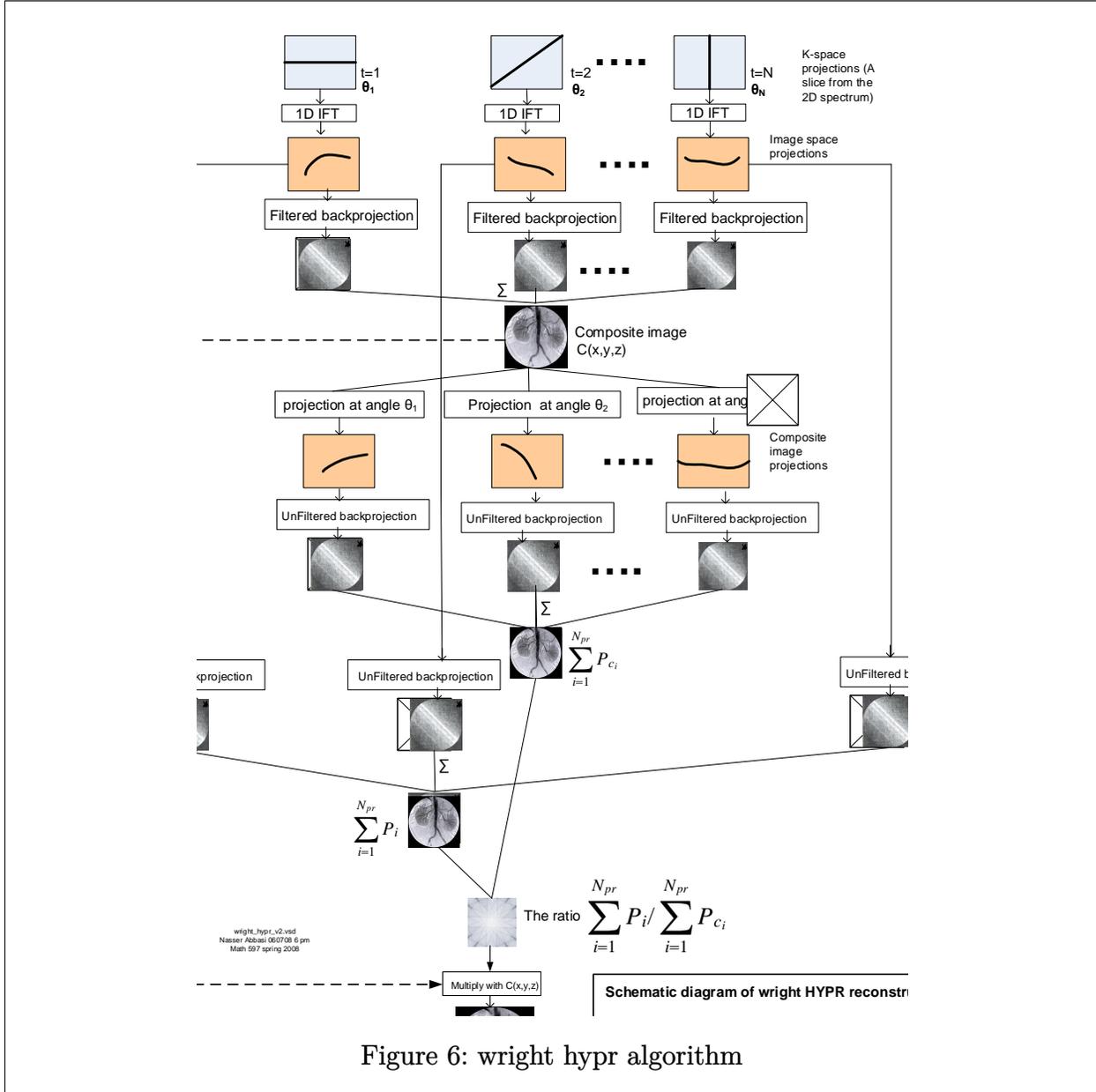


Figure 6: wright hypr algorithm

2.2.2 Wright-HYPR algorithm

The following is the algorithm for Wright HYPR written in high level pseudo-code. For our implementation we will have the input to the Wright-HYPR algorithm as a set of original images I_i . In other words, we will not start from the frequency domain, but from the spatial domain directly.

INPUT: N total number of projections
 M number of time frames

```

        set of 2D images {I}. The set size should be of size N*M
        Theta_Initial,Theta_End -- initial and final angle (limited view)
OUTPUT:  M number of HYPR 2D images

-- divide angles from Theta_Initial to Theta_Final into N*M equal divisions
Theta = linspace(Theta_Initial,Theta_Final,M*N)

FOR i from 1 to N LOOP
    g(i) = radon(I(i),Theta(i)) -- generate forward projections
    Filtered_Back_Projection(i) = iradon(g(i), Theta(i))
END LOOP

C = SUM( Filtered_Back_Projection ) -- composite image
Nr = N/M -- Number of projections per time frame

Starting_Projection = 1
FOR i from 1 to M LOOP
    J(i) = Wright_HYPR( C, g( Starting_Projection:Starting_Projection+Nr-1),
                        Theta( Starting_Projection:Starting_Projection+Nr-1) )
    Starting_Projection = Starting_Projection + Nr
END LOOP

--
-- Wright HYPR function
--
FUNCTION Wright_HYPR( C, g, Theta) RETURNS HYPR_IMAGE
    Nr = Length(g) --- Number of projections Per Frame

    FOR i from 1 to Nr
        gc(i) = radon(C, Theta(i))
        PC(i) = iradon( gc(i), Theta(i), FILTER='NONE')
        P(i) = iradon( g(i), Theta(i), FILTER='NONE')
    END LOOP

    mask = SUM(P) / SUM(PC) -- Element by element division
    RETURN( mask * C ) -- Element by element multiplication

END FUNCTION Wright_HYPR

```

2.2.3 On the difference between original HYPR and Wright-HYPR

This table shows the difference between the original HYPR and Wright-HYPR

HYPR frame J in original HYPR	$\frac{1}{N_{pr}} C \sum_{i=1}^{N_{pr}} \frac{P_i}{P_{c_i}}$
HYPR frame J in Wright HYPR	$C \frac{\sum_{i=1}^{N_{pr}} P_i}{N_{pr} \sum_{i=1}^{N_{pr}} P_{c_i}}$

This table shows the difference when we express the HYPR frame as function of original image I and the operator R only

HYPR frame J in original HYPR	$\frac{1}{N_{pr}} \left(\sum_{i=1}^N R_{\phi_i}^+ [R_{\phi_i} [I_i]] \right) \sum_{j=1}^{N_{pr}} \frac{R_{\phi_j}^* [R_{\phi_j} [I_j]]}{R_{\phi_j}^* \left[R_{\phi_j} \left[\sum_{\mu=1}^{N_{pr}} R_{\phi_\mu}^+ [R_{\phi_\mu} [I_\mu]] \right] \right]}$
HYPR frame J in Wright HYPR	$= \left(\sum_{i=1}^N R_{\phi_i}^+ [g_i] \right) \frac{\sum_{j=1}^{N_{pr}} R_{\phi_j}^* [R_{\phi_j} [I_j]]}{\sum_{j=1}^{N_{pr}} R_{\phi_j}^* \left[R_{\phi_j} \left[\sum_{\mu=1}^N R_{\phi_\mu}^+ [g_\mu] \right] \right]}$

3 Simulation software and results found

A Matlab simulation software was written to implement the HYPR algorithms as described above. A User interface developed to make it easier for the user to run different simulations using different parameters. The appendix describes the user interface in more details.

The following are the initial results of the simulation. relative RMSE (root mean square error relative to the user image frame) was computed for the HYPR frame generated and the corresponding original image over the same time frame used to generate the HYPR frame (The average of the original image over the time frame was used if the time frame contained more than one snap shot of the user image). One time frame was used, and hence one HYPR frame was generated. The number of projections was increased, and the RMSE was calculated each time. The following table summarizes the finding.

Number of projections	Original HYPR	Wright-Huang HYPR
16	2.172299	1.897816
128	0.884654	0.932832
256	0.779807	0.817493
512	0.743475	0.767085
700	1.241370	1.167796

Initial results shows that the original HYPR algorithm results in smaller RMSE than the Wright-Huang-HYPR algorithm using the Wright-Huang disk (WH disk), as the user image sampled for all the simulation. The WH disk is a disk of 50 pixels in diameter whose density is increased linearly from 1 to 128 units in time. It is also found that for later in time generated HYPR frames (i.e. HYPR frames generated from user images whose intensities were large), the RMSE is also larger than earlier HYPR images. This shows the larger the intensity of the original image, the less approximate is the HYPR image to the original image. This needs to be investigated more to see the reason why the HYPR algorithm is sensitive to intensity.

In addition to measuring the RMSE between each HYPR frame and corresponding image frame, the mean of each frame is calculated.

Another result found is that the Iterative HYPR algorithm using the original HYPR algorithm produces the least RMSE with 5 iterations. This shows that I-HYPR does improve the final quality of the HYPR images. Convergence seems to be fast. After 4 iterations, the RMSE error did not change too much (please see table below). Most of the image improvement (as measured by lower RMSE) occurred in the first 3 iterations.

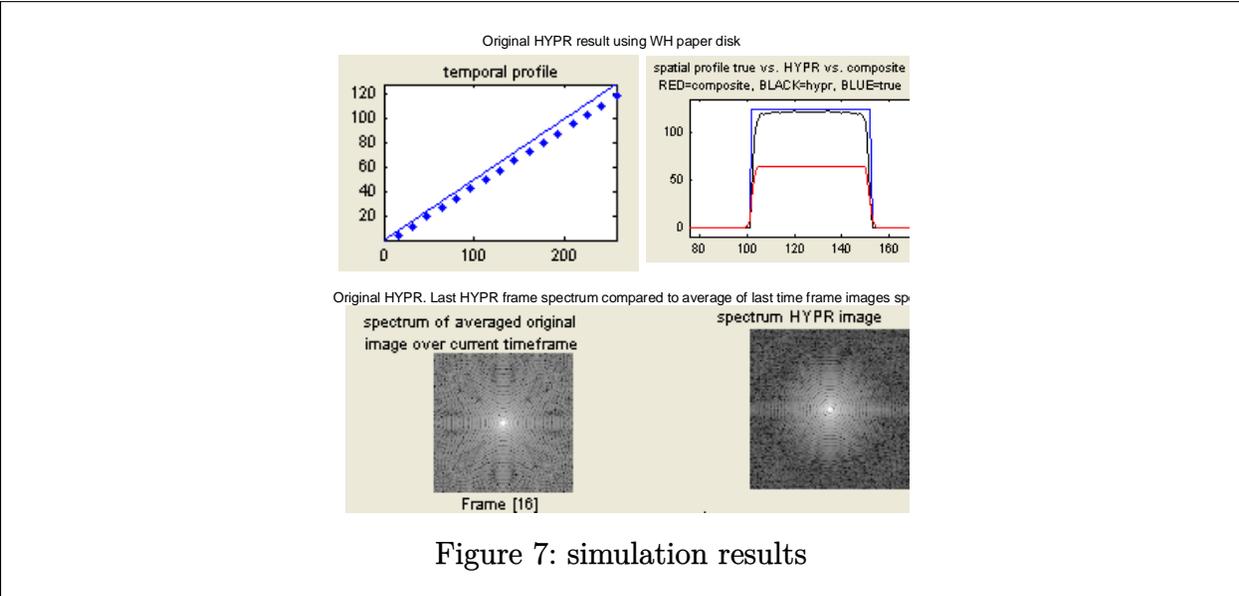
The following table lists the RMSE and means found when running the original and Wright-Huang HYPR and with running I-HYPR for 5 iterations.

16 Frames were generated for each simulation. All simulations used 16 projections (or 8 per MRI definition) per a time frame, over full view (0-180 degrees) and used 16 time frames.

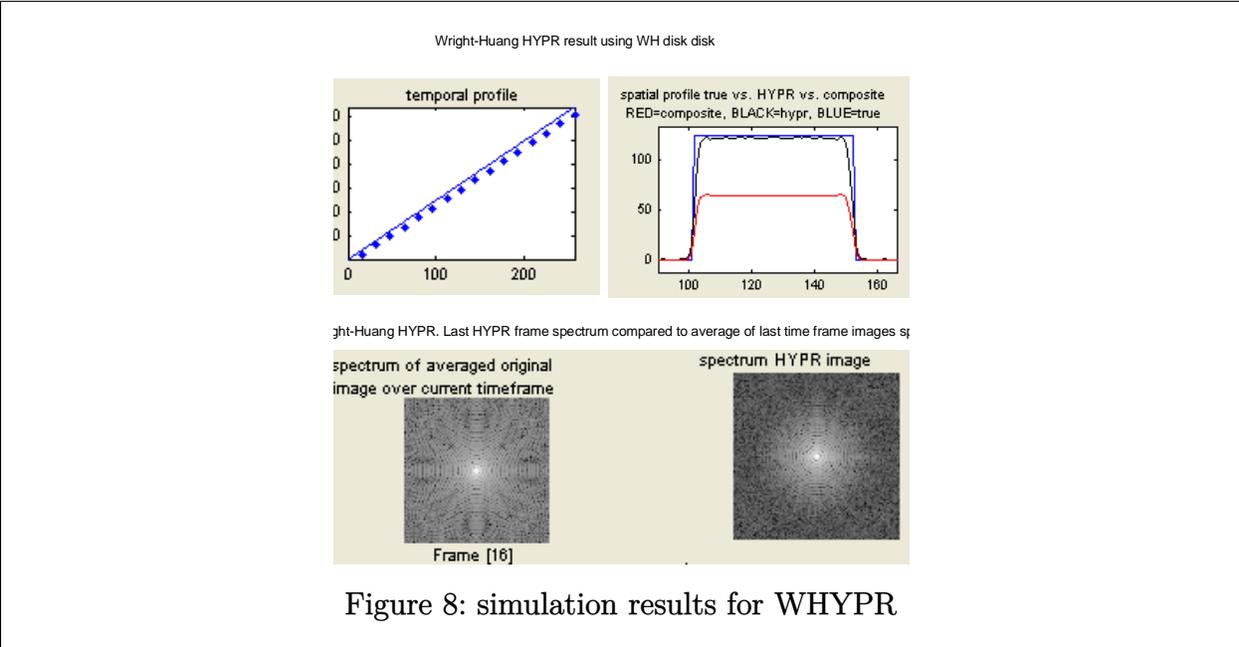
The following table shows the convergence of the I-HYPR. The RMSE is measured as more iterations are performed.

16 Frames were generated for each simulation. All simulations used 16 projections (or 8 per MRI definition) per a time frame, over full view (0-180 degrees) and used 16 time frames. Notice that converges seem to occur after 4 steps. (not counting the initial zero step to create the first composite image).

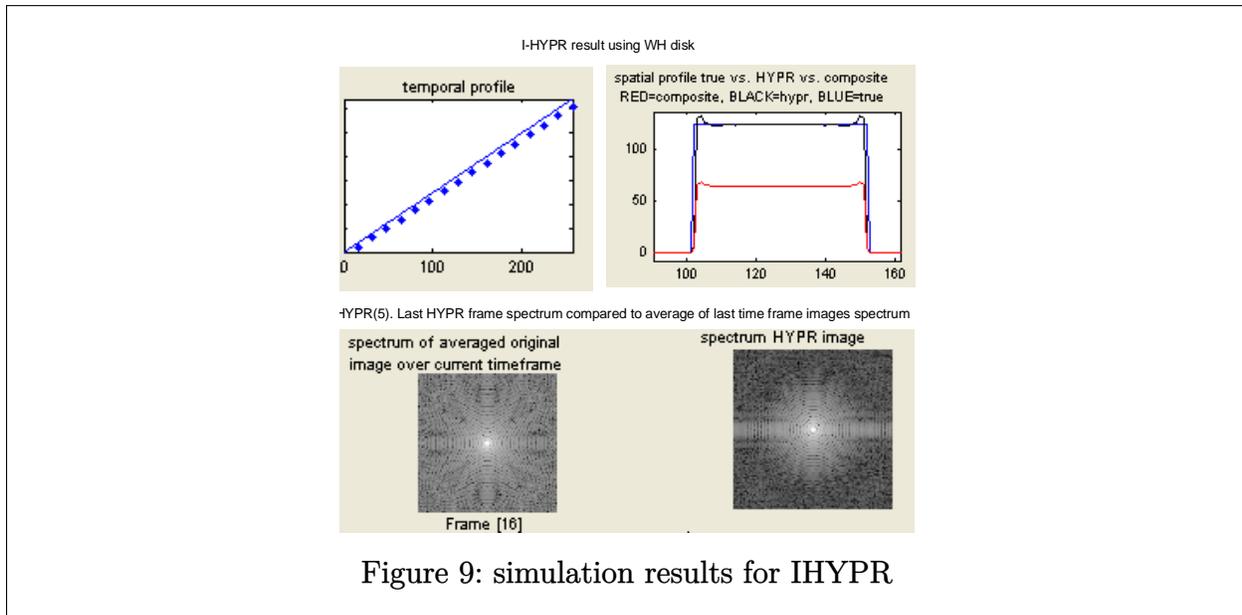
This diagram shows the result of running the Original HYPR algorithm on the WH disk. Notice that the composite image intensity (red line below) is averaged over all time frames. This image shows the last HYPR frame and it is compared with the average of the user images over the same time frame. Notice that the HYPR frame is close in intensity to the corresponding user image, but some variations on the circle edges show up.



This is the same result as above, but for the Wright-Huang HYPR algorithm.



This is the same result for I-HYPR with 5 iterations



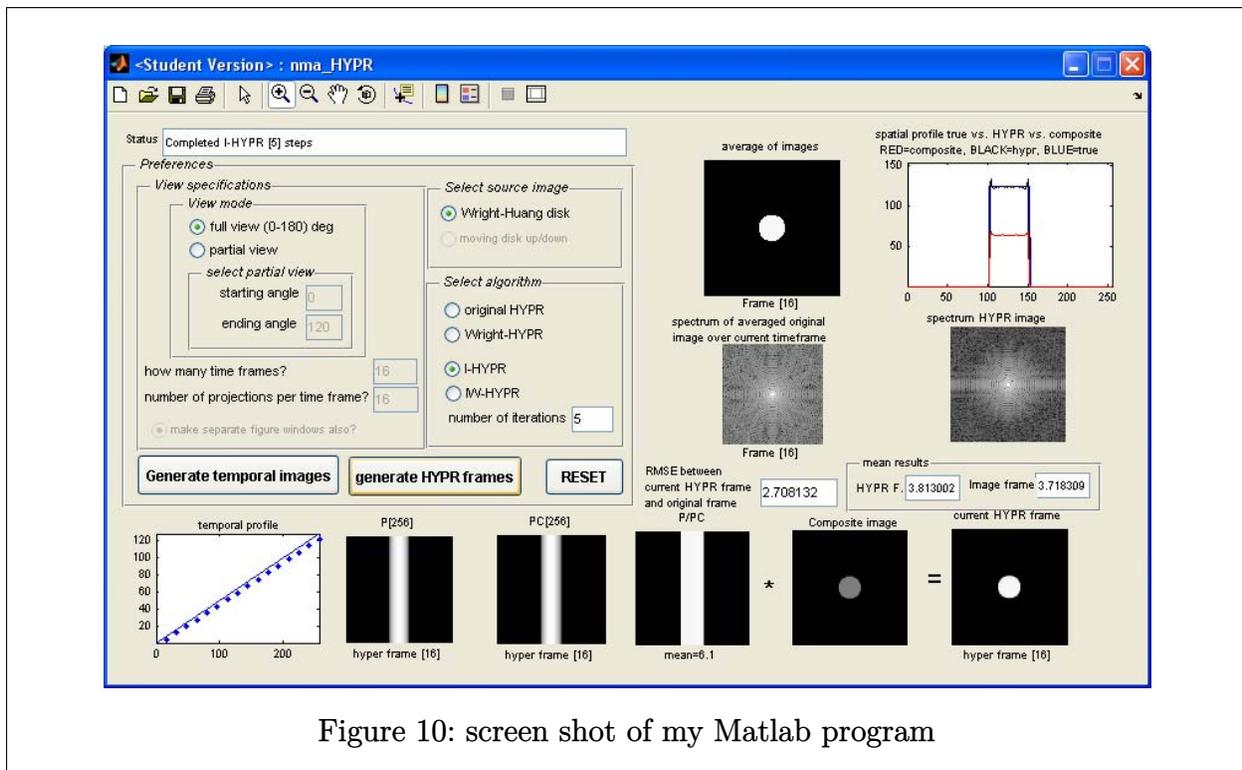
Notice the above spatial profile. The HYPR image (black line) is closer the true line than the earlier results using original and WH HYPR. However, we notice an interesting artifact here, a Gibbs like phenomena around the edges of the disk shows up where the HYPR converges is not coming to exact at those places. This needs more investigation.

3.1 Additional work to be done

1. Implement iterative HYPR using the Wright-Huang version of HYPR as its iterative step. Compare results.
2. Run all the above simulation for user images with small object close to each others (small disks) and change intensity to simulate time changes. Compare quality of HYPR images reconstructed with those when the images are further apart.
3. Run all the above simulation for user image which moves in time and changes intensity as well. Use a DISK that moves in different configurations.
4. Investigate the Gibbs-like phenomena observed around the edges of the disk when using I-HYPR.
5. Investigate why original HYPR produces a better RMSE than Wright-Huang, while the temporal profile of the last HYPR frame has more offset from the true profile as compared to the Wriugh-Huang profile. This seems counter intuitive. It is possible I am doing something wrong in the simulation?

3.2 Matlab software

The following is a screen shot of the Matlab simulation software.



4 References

1. Highly Constrained Back projection for Time-Resolved MRI by C. A. Mistretta, O. Wieben, J. Velikina, W. Block, J. Perry, Y. Wu, K. Johnson, and Y. Wu
2. Iterative projection reconstruction of time-resolved images using HYPR by O'Halloran et.all
3. Time-Resolved MR Angiography With Limited Projections by Yuexi Huang¹,and Graham A. Wright
4. GE medical PPT dated 6/6/2008
5. Book principles of computerized Tomographic imaging by Kak and Staney