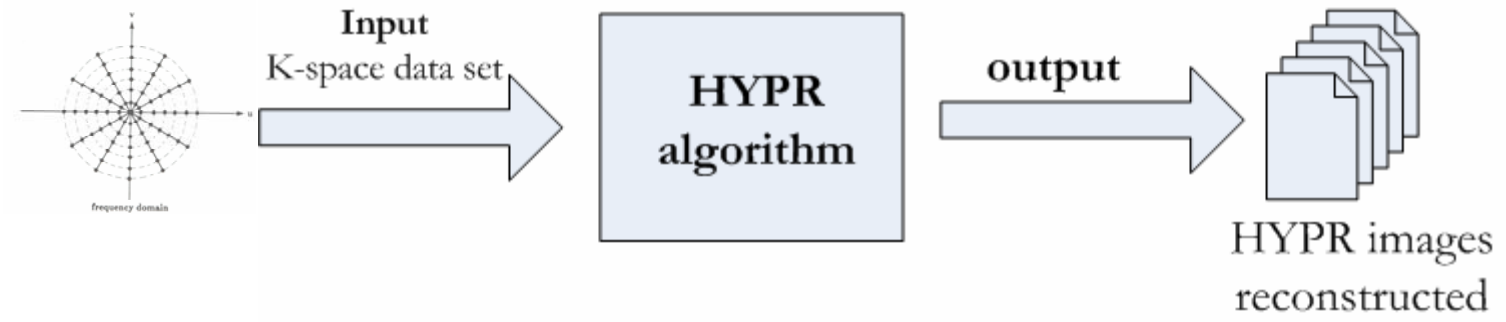
HYPR Project Presentation

By Nasser Abbasi

HYPR Input-Output view



The HYPR algorithm uses a composite image and a weighting image during reconstruction in order to generate images with reduced artifacts and with improved temporal characteristics

Weighting image	*	Composite image	=	HYPR image
Good temporal characteristics low SNR low spatial characteristics		Poor temporal characteristics good SNR good spatial characteristics less streak artifacts		good temporal characteristics good SNR good spatial characteristics reduced artifacts

Five HYPR Based Algorithms Studied

- Original HYPR $J_k = \frac{1}{N_p} C \sum_{i=1}^{N_p} \frac{P_i}{P_{c_i}}$
- Wright-Huang HYPR $J_k = C \left(\sum_{i=1}^{N_p} P_i \Big/ \sum_{i=1}^{N_p} P_{c_i} \right)$
- Iterative original HYPR
- Iterative Wright-Huang HYPR
- HYPR-LR $J_k = C \left[\left(F \otimes \sum_{i=1}^{N_p} \tilde{P}_i \right) \Big/ \left(F \otimes \sum_{i=1}^{N_p} \tilde{P}_{c_i} \right) \right]$

Notation

N_p : Number of projections in one time frame

C : Composite image

P_i : Unfiltered backprojection from original data

P_{c_i} : Unfiltered backprojection from C

i : Angle θ_i was used for backprojection

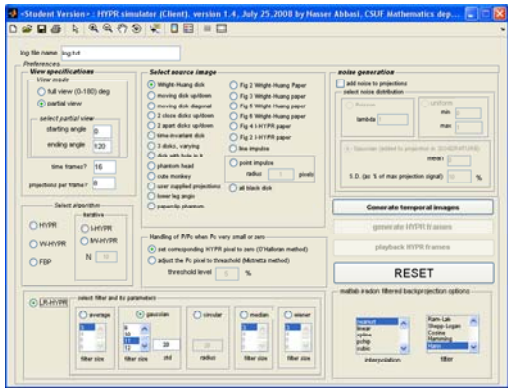
$\tilde{P}_{c_i}$: Filtered backprojection from original data

$\tilde{P}_i$: Filtered backprojection from C

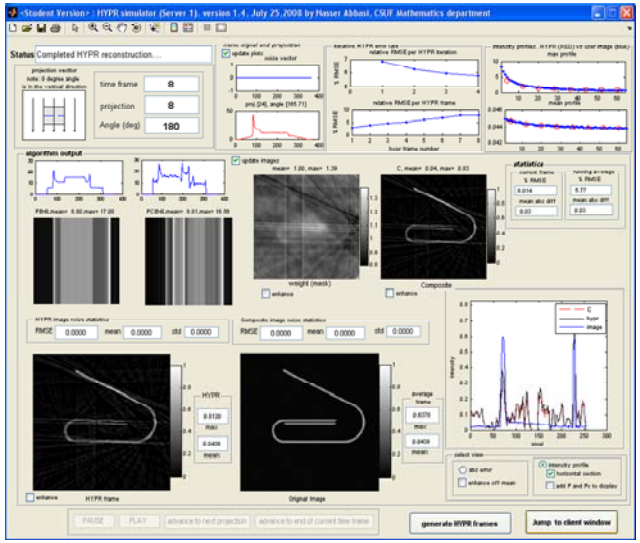
F : Low pass filter $\otimes$: Convolution operator

HYPR Simulator Review

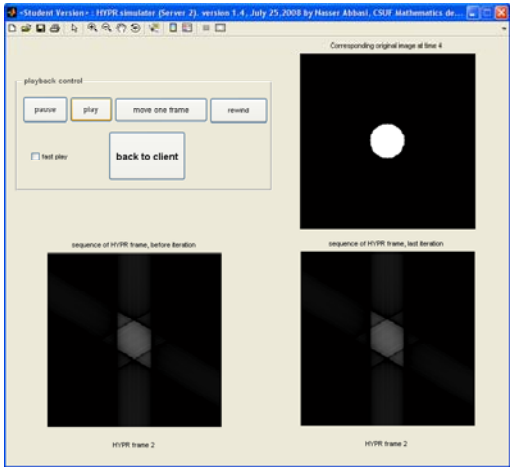
Client : HYPR simulation Options and preferences



Server 1: HYPR simulation computation image output and plots



Server 2: Playback of reconstructed HYPR images and Iterative HYPR



HYPR log file

Log.txt

Over 8,000 lines of Matlab code.

Implement HYPR, W-HYPR, FBP, I-HYPR, IW-HYPR, and HYPR-LR.

Implements a noise process as a separate work flow.

Three types of noise can be added: Normal, Poisson and Uniform.

Detailed log file contains all the results and statistics generated.

HYPR-LR implemented with five different low pass filters.

Allows the user to load their own set of projection data.

More than 20 prepackaged test image cases included.

Allows the user to play back the final reconstructed HYPR images…and many more features…

Comparing the Algorithms

Algorithm	Advantages	Disadvantages
O-HYPR	**Suitable to use with images with high sparsity and limited object movements.**	• **Crosstalk when objects are close to each others.** • **Difficulty with images that exhibit significant spatial and temporal dynamics.**
W-HYPR	• **Better noise response than O-HYPR, higher SNR.**	**Similar to O-HYPR.**
LR-HYPR	• **Can be applied to images acquired with arbitrary k-space trajectories.** • **Reconstruction time is shorter than with for iterative methods or O-HYPR.**	• **Crosstalk still exists. Use of sliding window can reduce this problem.** • **Using inappropriate low pass filter type and parameters can result in worst reconstruction.**
I-HYPR	**Improves temporal characteristics and accuracy.**	**Noise amplified making reconstruction worst.**
IW-HYPR	**Suppresses noise amplification more than I-HYPR.**	**Noise is still amplified (but at lower levels).**

Accuracy of Algorithms Using the GE Phantom Clip

Results of two tests cases, one with noise (zero mean, 5% S.D. of maximum projections) and one without noise. Both used 8 time frames and 8 projections per time frame. For Iterative HYPR, 10 iterations were used. HYPR-LR used the circular low pass filter with size 20 pixels.

test	O-HYPR	W-HYPR	HYPR-LR	I-HYPR		IW-HYPR	
				1^{st}	10^{th}	1^{st}	10^{th}
No noise	6.83	6.77	6.7	6.83	5.41	6.77	5.55
With noise	10.76	9.55	13.7	10.76	14.22	9.55	11.436

Wright-Huang based HYPR algorithms have the best overall results