



# **Compressive Imaging via Approximate Message Passing with Wavelet-Based Image Denoising**

*Jin Tan, Yanting Ma, and Dror Baron*

Atlanta, GA  
December 4<sup>th</sup>, 2014

Supported by NSF and ARO



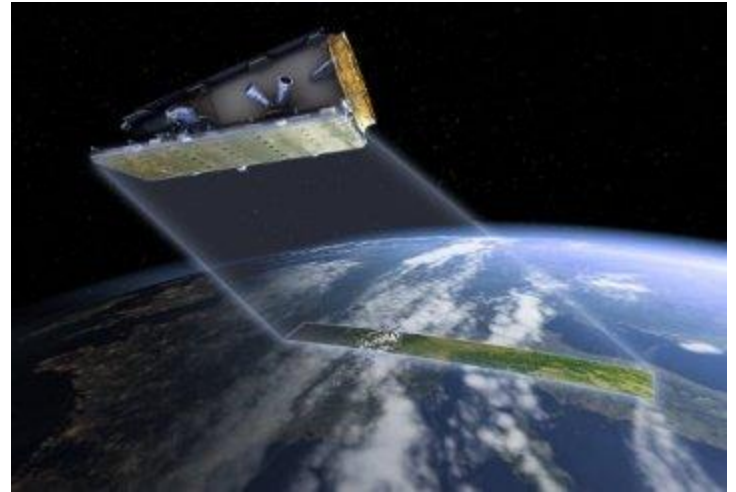
# Compressive Imaging

# Compressive imaging

*Less radiation*



*Less power consumption*



# Compressive imaging



Dimension  $N$



$N \gg M$



Dimension  $M$



Linear  
measurements

Dimension  $M$

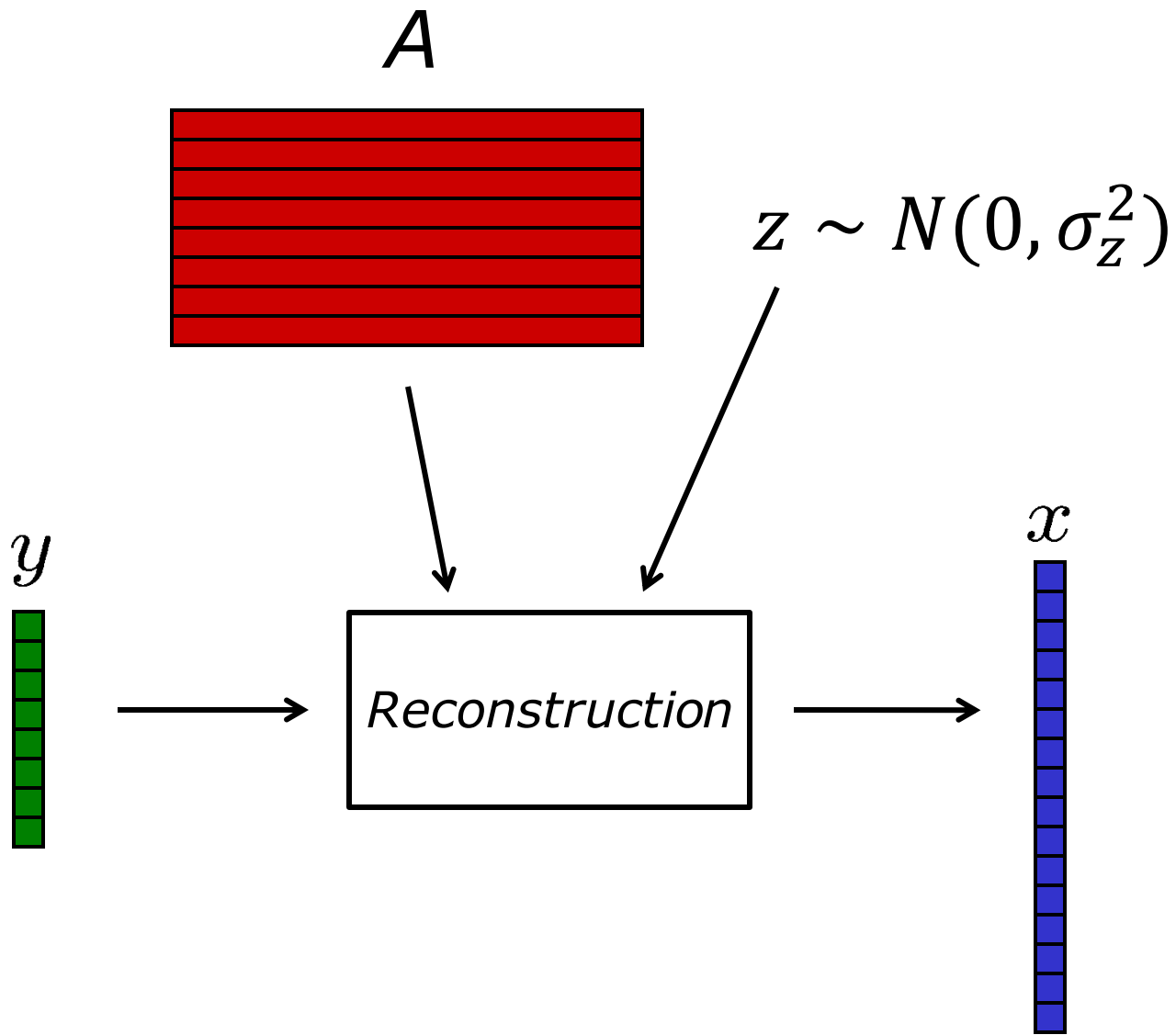


Dimension  $N$

# Compressive imaging

- Length- $N$  input  $x$
- Matrix  $A$ , dimension  $M \times N$ ,  $M < N$
- Additive white Gaussian noise
- Well known: modest # measurements  $M$  suffices for robust signal reconstruction

The diagram illustrates the equation  $y = Ax + z$ . On the left, a green vertical vector labeled  $y$  is shown. To its right is an equals sign. Further right is a red matrix labeled  $A$ , which is wider than it is tall. To the right of the matrix is a blue vertical vector labeled  $x$ . To the right of the vector  $x$  is a plus sign. Finally, on the far right, a yellow vertical vector labeled  $z$  is shown.

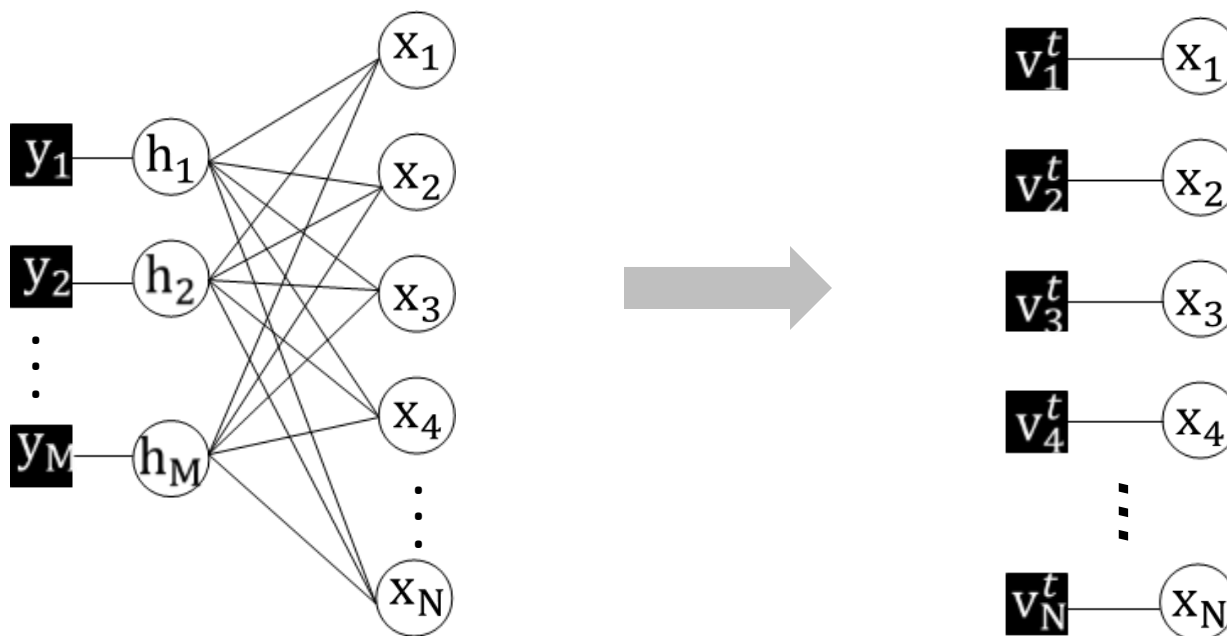


# Approximate Message Passing

# Approximate message passing (AMP)

[Donoho, Maleki, & Montanari 2009]

- Fast iterative algorithm
- Based on belief propagation



$$y = h + z = Ax + z$$

$$v = x + \text{noise}$$



# Approximate message passing (AMP)

Initialize  $x^0 = 0$

At iteration  $t$ , do

$$\text{Residual: } r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$$

$$\text{Noisy image: } v^t = x^t + A^T r^t$$

$$\text{Denoising: } x^{t+1} = \eta_t(v^t)$$



# Approximate message passing (AMP)

Initialize  $x^0 = 0$

At iteration  $t$ , do

*Onsager term*

$$\text{Residual: } r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$$

$$\text{Noisy image: } v^t = x^t + A^T r^t$$

$$\text{Denoising: } x^{t+1} = \eta_t(v^t)$$

# Approximate message passing (AMP)

Initialize  $x^0 = 0$

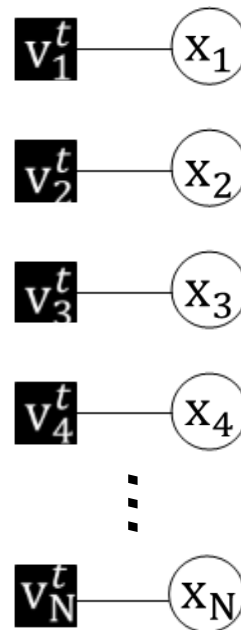
At iteration  $t$ , do

$$\text{Residual: } r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$$

$$\text{Noisy image: } v^t = x^t + A^T r^t$$

$$\text{Denoising: } x^{t+1} = \eta_t(v^t)$$

Standard AMP:  $\eta_t(v^t)$  is **scalar**



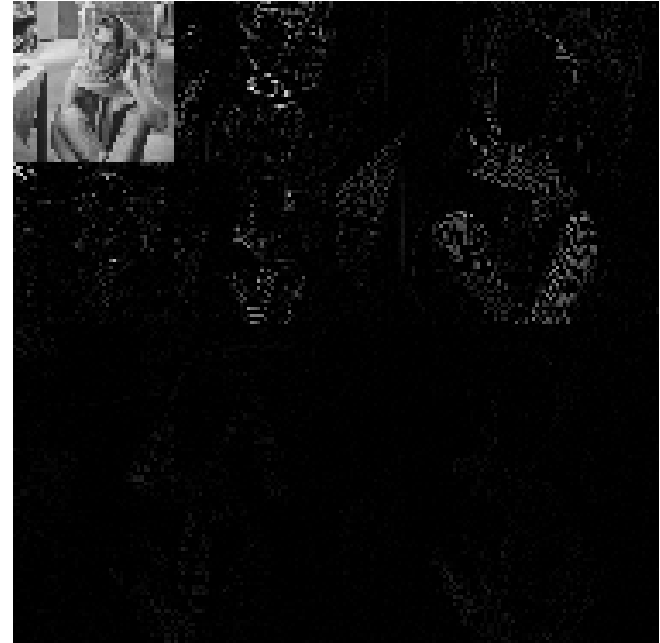
# Wavelet-Based Image Denoising

# Wavelet-based image denoising

Wavelet-based, convenient for Onsager term computation



Original image



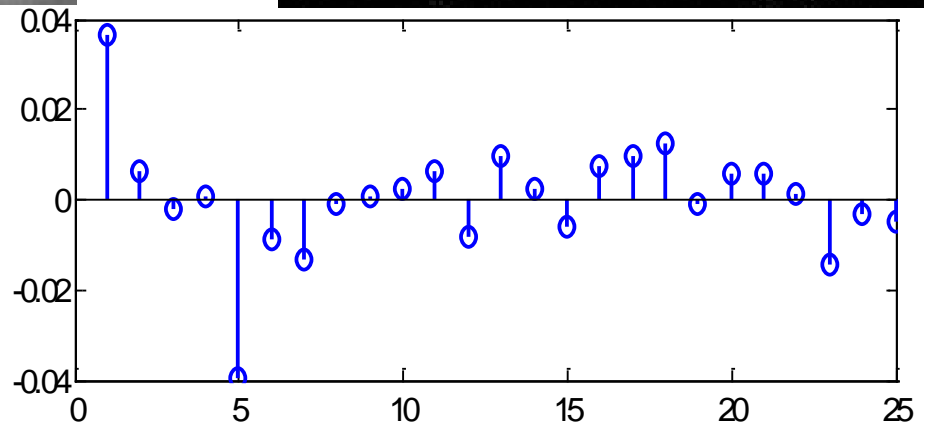
Wavelet transform

# Wavelet-based image denoising

Adaptive Wiener filter [Mihcak et al. 1999]



*Values of neighboring coefficients*



# Wavelet-based image denoising

Adaptive Wiener filter [Mihcak et al. 1999]

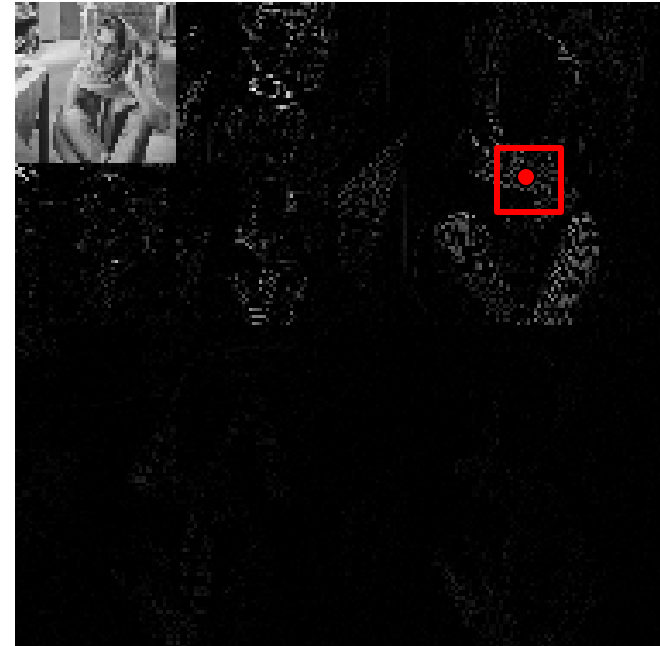
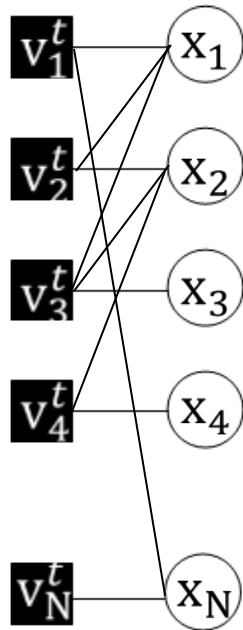


Compute variances of wavelet coefficients based on neighborhood,  $\sigma_i^2$

Adaptive Wiener filtering:  $\frac{\sigma_i^2}{\sigma_i^2 + \sigma_z^2} \cdot i\text{-th noisy wavelet coefficient}$

# Wavelet-based image denoising

Adaptive Wiener filter [Mihcak et al. 1999]



**Non-scalar denoiser**

Compute variances of wavelet coefficients based on neighborhood,  $\sigma_i^2$

Adaptive Wiener filtering:  $\frac{\sigma_i^2}{\sigma_i^2 + \sigma_z^2} \cdot i\text{-th noisy wavelet coefficient}$



# AMP with Adaptive Wiener Filter (AMP-Wiener)

# AMP-Wiener

Initialize  $x^0 = 0, r^0 = 0$

At iteration  $t$ , do

Residual:  $r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$

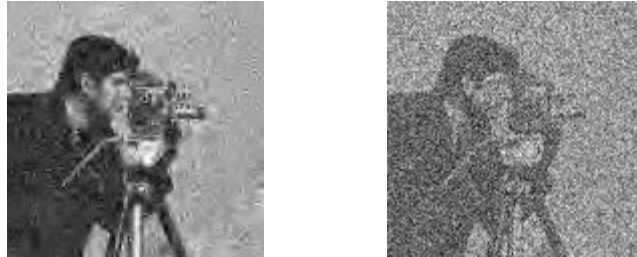
Noisy image:  $v^t = x^t + A^T r^t$

Denoising:  $x^{t+1} = \eta_t(v^t)$

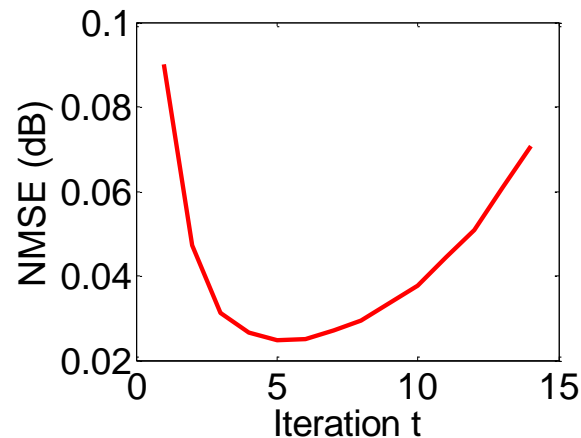


# AMP-Wiener

$$\text{Denoising: } x^{t+1} = \eta_t(v^t)$$



- ✓ Noise variance is approx.  $\|r^t\|_2^2/M$  [Montanari 2012]
- ✓ Divergence problem



# AMP-Wiener

Denoising:  $x^{t+1} = \eta_t(v^t)$



- ✓ Noise variance is approx.  $\|r^t\|_2^2/M$  [Montanari 2012]
- ✓ Divergence problem: damping [Rangan et al. 2014]  
 $\alpha \cdot$  current estimate +  $(1 - \alpha) \cdot$  previous estimate  
( $0 < \alpha \leq 1$ )

# Numerical Results

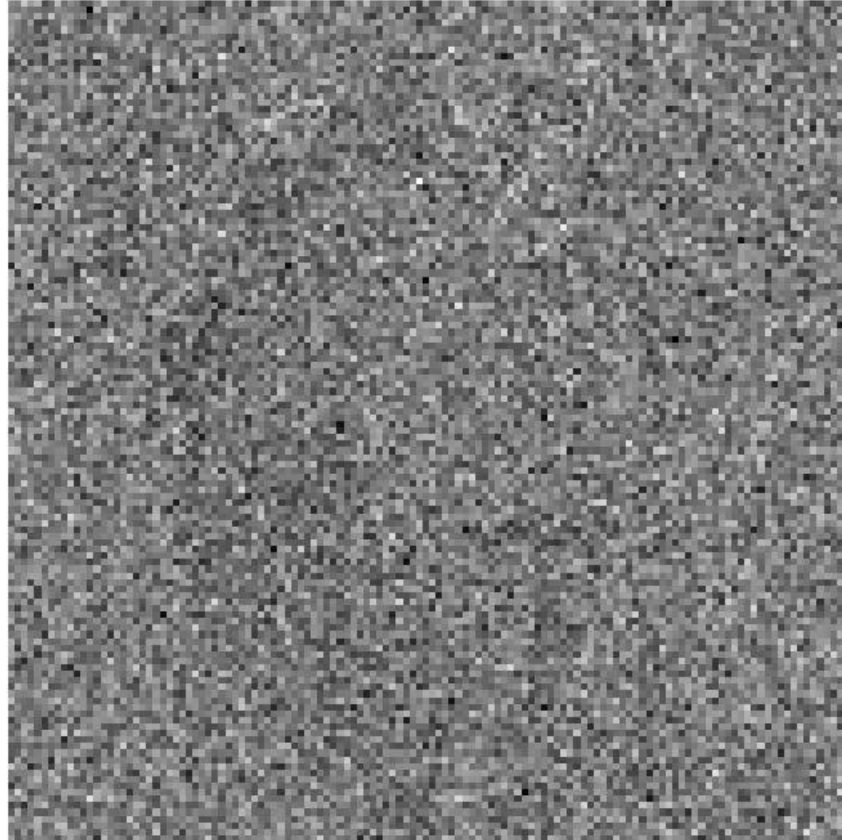
# Numerical results

Original



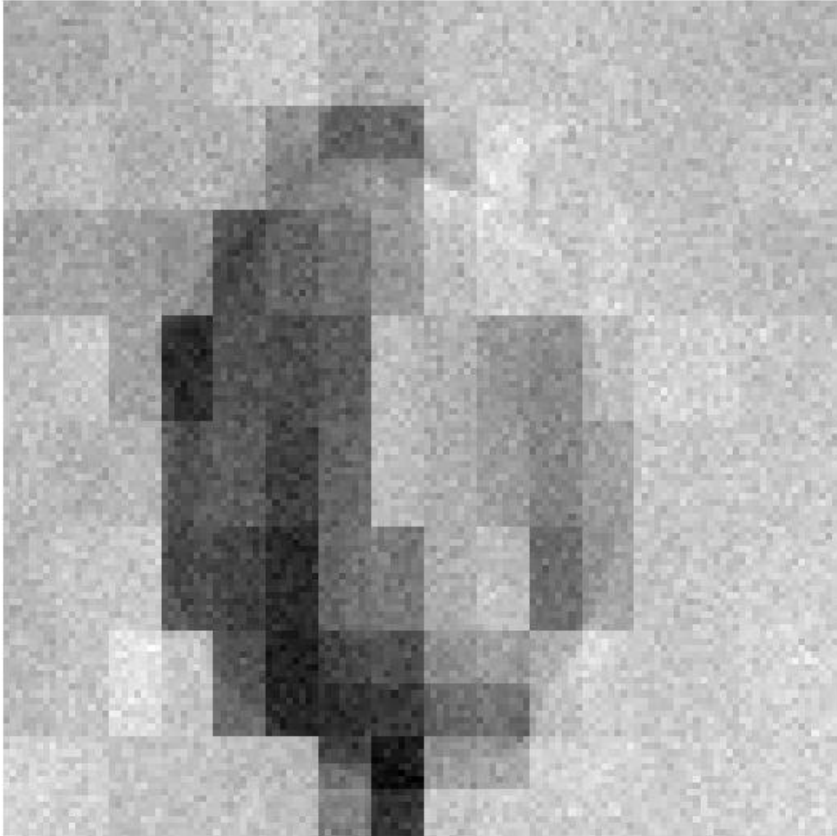
# Numerical results

Iteration 1



# Numerical results

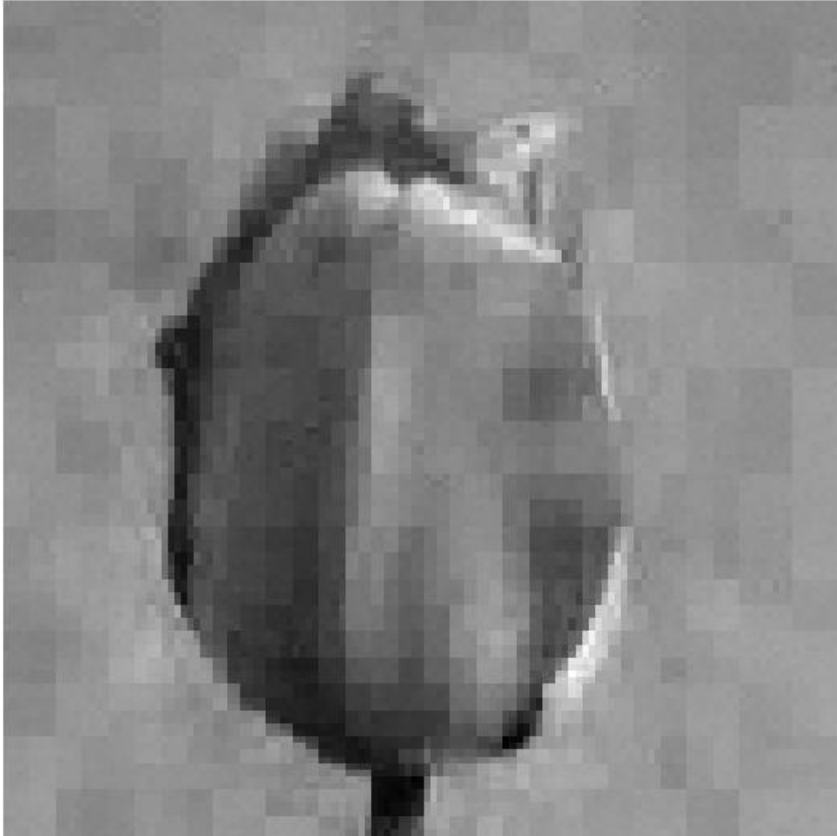
Iteration 3





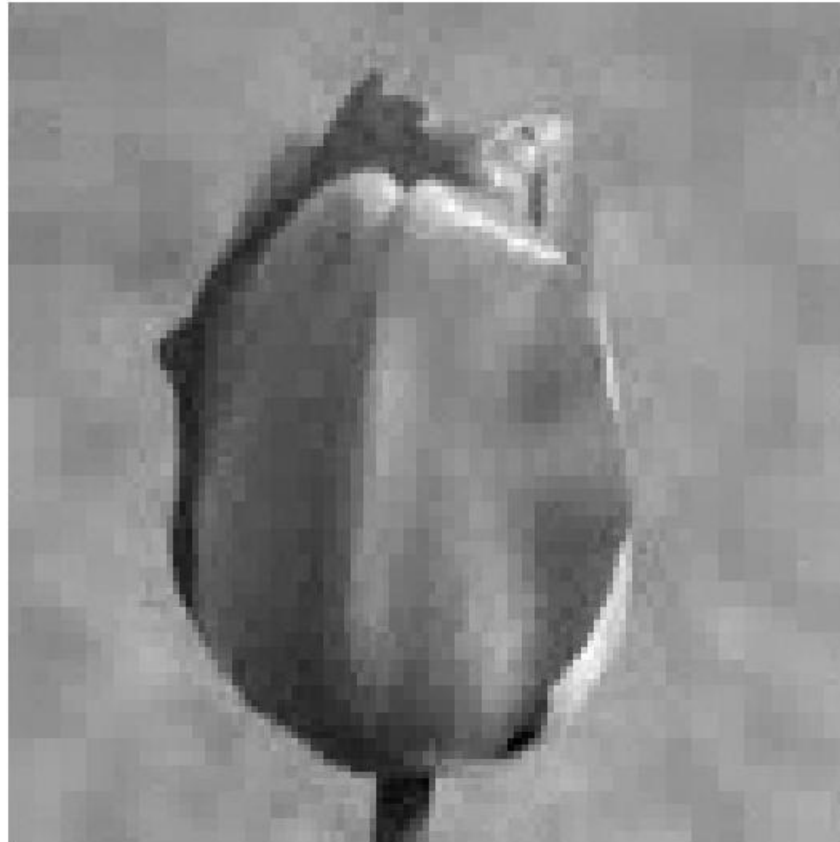
# Numerical results

Iteration 7



# Numerical results

Iteration 30



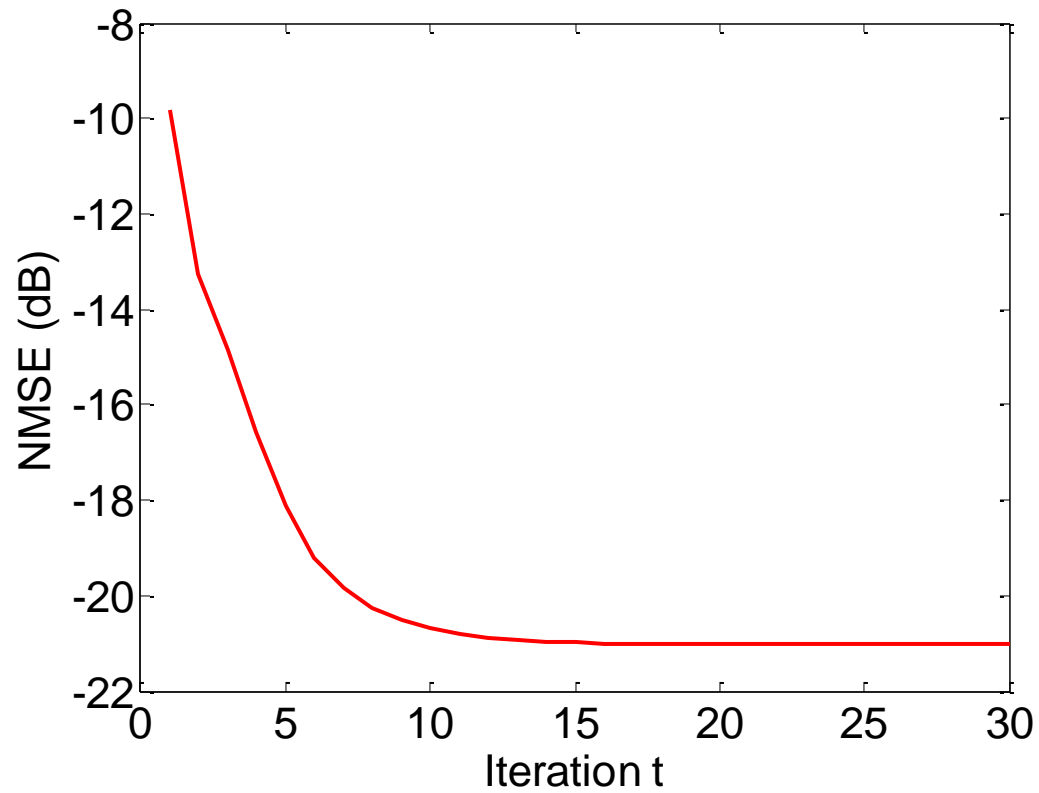
# Numerical results

A: i.i.d. zero-mean Gaussian, Measurement rate 0.3  
Average over 591 images

<b>Algorithm</b>	<b>NMSE(dB)</b>	<b>Runtime(sec)</b>
Turbo-BG	-20.37	12.39
Turbo-GM	-20.72	12.47
MCMC	-20.31	>400
AMP-Wiener	<b>-21.00</b>	<b>3.34</b>

[Turbo-BG/GM: Som & Schniter 2012]  
[MCMC: He & Carin 2009]

# Numerical results



# Numerical results

AMP-Wiener [Tan et al., May 2014]

Algorithm	NMSE(dB)	Runtime(sec)
Turbo-BG	-20.37	12.39
Turbo-GM	-20.72	12.47
MCMC	-20.31	>400
AMP-Wiener	-21.00	3.34

$$r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$$



$$= \eta_t \left( \text{noisy image} \right)$$



# Numerical results

AMP-Wiener [Tan et al., May 2014]

Algorithm	NMSE(dB)	Runtime(sec)
Turbo-BG	-20.37	12.39
Turbo-GM	-20.72	12.47
MCMC	-20.31	>400
AMP-Wiener	-21.00	3.34

$$r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1} (x^{t-1} + A^T r^{t-1}) \rangle$$



## AMP-BM3D with Monte Carlo

[Metzler et al., June 2014]

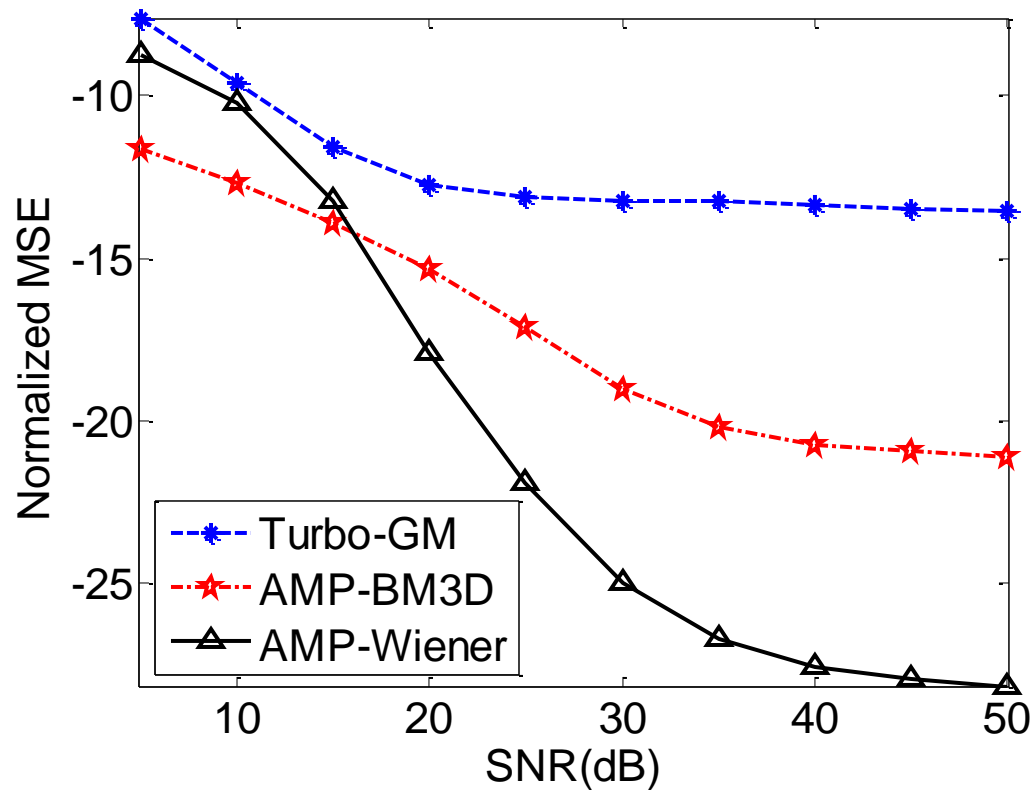
# Numerical results

<b>Algorithm</b>	<b>NMSE(dB)</b>	<b>Runtime(sec)</b>
Turbo-BG	-20.37	12.39
Turbo-GM	-20.72	12.47
MCMC	-20.31	>400
AMP-Wiener	-21.00	<b>3.34</b>
AMP-BM3D	<b>-25.27</b>	16.06

***A***: *i.i.d. zero mean Gaussian*

# Radio astronomy imaging system

$$y = \text{blurring kernel} * x + z$$

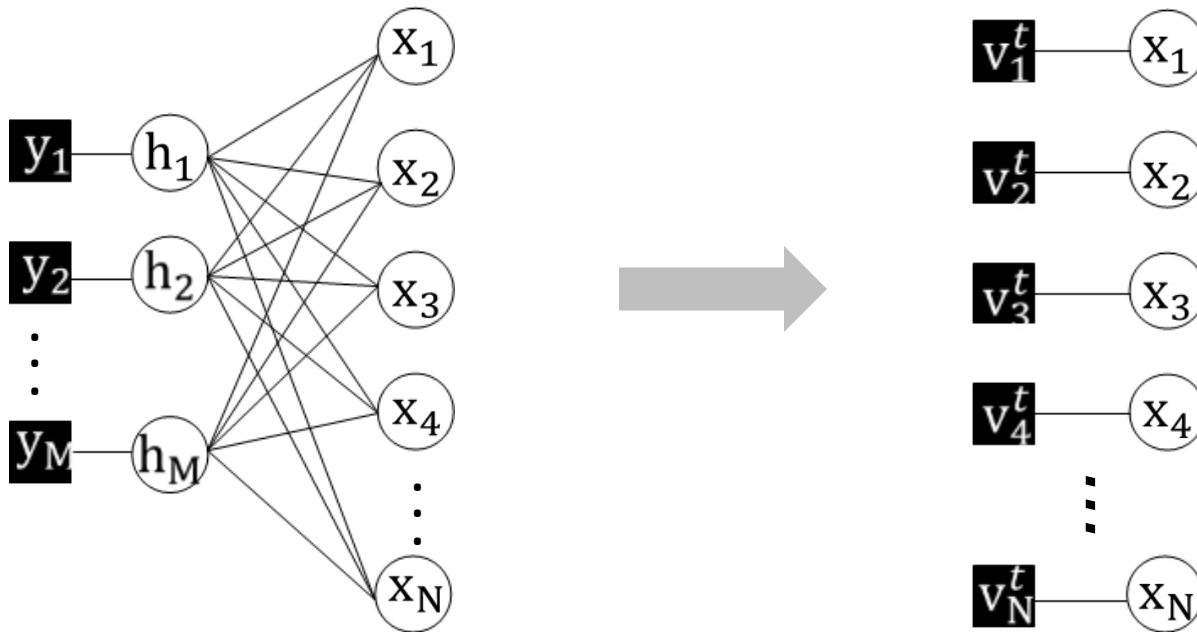




# Conclusion

# Conclusion

Approximate message passing:  
Convert matrix channel problem to  
scalar denoising problem



# Conclusion

Denoising via adaptive Wiener filter:

$$x^{t+1} = \eta_t(v^t)$$



# Conclusion

## Adaptive Wiener filter:

A robust denoiser with simple derivative

$$r^t = y - Ax^t + \frac{r^{t-1}}{M/N} \langle \eta'_{t-1}(x^{t-1} + A^T r^{t-1}) \rangle$$

*Onsager term*

Questions?