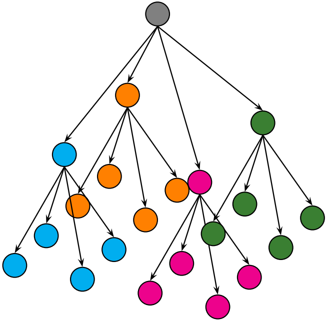
Image Denoising with Nonparametric Hidden Markov Trees

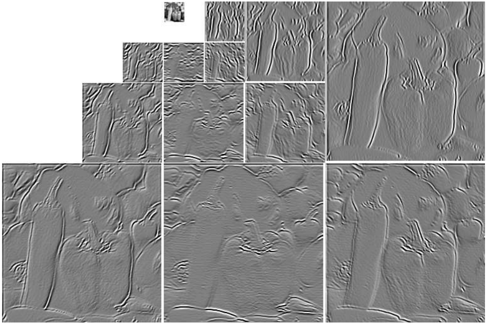
Jyri Kivinen   Erik Sudderth   Michael Jordan

Helsinki University of Technology   University of California, Berkeley

IEEE International Conference on Image Processing, 2007

Statistical Models for Wavelet Decompositions



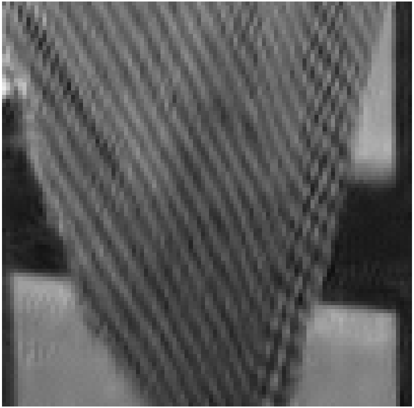


Goal: *Learn* good statistical models of natural images

Approach:
- Capture global statistics of wavelets via a *tree-structured* latent variable model
- Automatically adapt to data complexity via nonparametric, *Dirichlet process* priors

Denoising Images in Wavelet-Domain

Local Gaussian Scale Mixture
(Portilla et al. 2003)



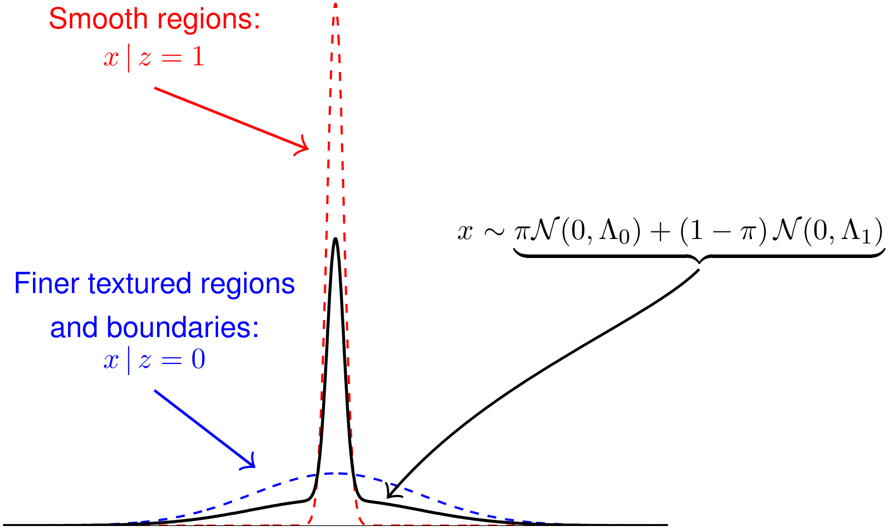
Hierarchical Dirichlet Process
Hidden Markov Tree

Apply learned statistics to *denoise* images

- model parameters are estimated *directly* from noisy image
- *global* statistics increase *robustness*

Modeling Wavelets Locally

A Two-State Mixture

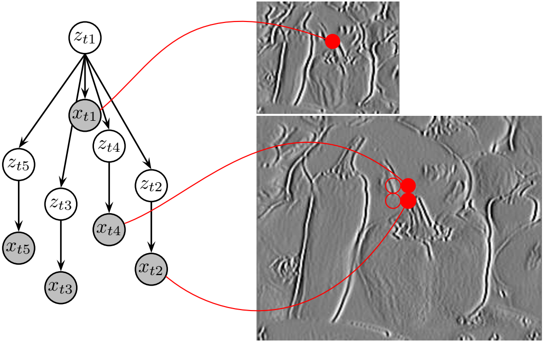


Modeling Wavelets Globally

Hidden Markov Tree Model (Crouse et al. 1998)

- Markov dependencies capture spatial relationships

- Models each orientation independently

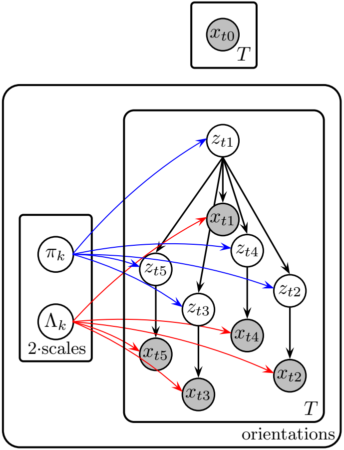


z_{ti}: hidden cluster assignment variable

x_{ti}: observed wavelet coefficient

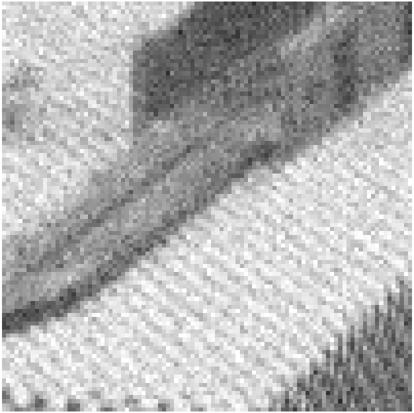
Modeling Wavelets Globally

Hidden Markov Tree Model (Crouse et al. 1998)

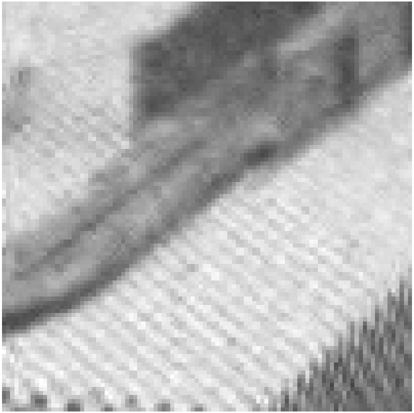


- Wavelet coefficients marginally distributed as mixtures of two zero-mean Gaussians
- Markov dependencies between hidden variables capture relationships among nearby coefficients
- Parameters chosen independently for each scale and orientation
- Successfully used for image denoising
- But...

Hidden Markov Tree (Crouse et al. 1998)



noisy



denoised

- Is two states per scale enough? How many is enough?
- Should I share states in the same pattern for all images?
- How should my model change for other wavelet decompositions?

Dirichlet Process Mixtures

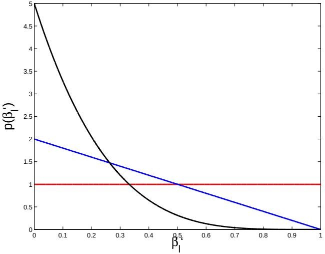
$$p(x \mid \beta, \Lambda_1, \Lambda_2, \ldots) = \sum_{k=1}^{\infty} \beta_k \mathcal{N}(x; 0, \Lambda_k)$$

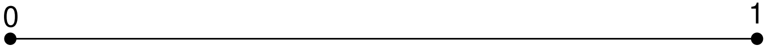
Control complexity using a *stick-breaking prior* for mixture weights:

$$\beta_k = \beta'_k \prod_{\ell=1}^{k-1} (1 - \beta'_\ell)$$

$$\beta'_\ell \sim \mathrm{Beta}(1, \gamma)$$

γ : concentration parameter





Dirichlet Process Mixtures

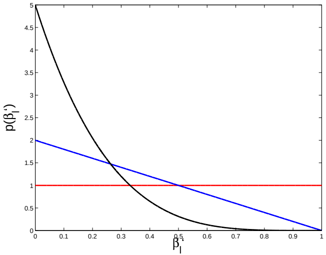
$$p(x \mid \beta, \Lambda_1, \Lambda_2, \ldots) = \sum_{k=1}^{\infty} \beta_k \mathcal{N}(x; 0, \Lambda_k)$$

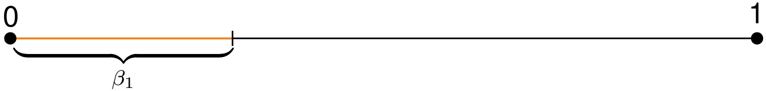
Control complexity using a *stick-breaking prior* for mixture weights:

$$\beta_k = \beta'_k \prod_{\ell=1}^{k-1} (1 - \beta'_\ell)$$

$$\beta'_\ell \sim \mathrm{Beta}(1, \gamma)$$

γ : concentration parameter





Dirichlet Process Mixtures

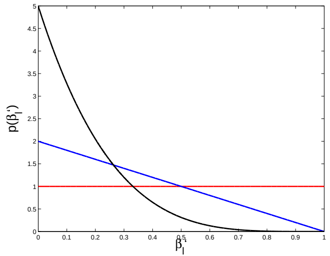
$$p(x \mid \beta, \Lambda_1, \Lambda_2, \ldots) = \sum_{k=1}^{\infty} \beta_k \mathcal{N}(x; 0, \Lambda_k)$$

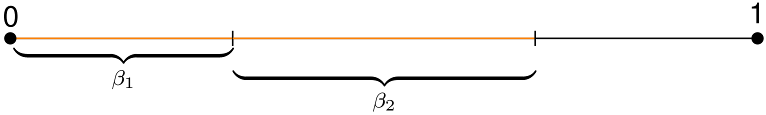
Control complexity using a *stick-breaking prior* for mixture weights:

$$\beta_k = \beta'_k \prod_{\ell=1}^{k-1} (1 - \beta'_\ell)$$

$$\beta'_\ell \sim \mathrm{Beta}(1, \gamma)$$

γ : concentration parameter





Dirichlet Process Mixtures

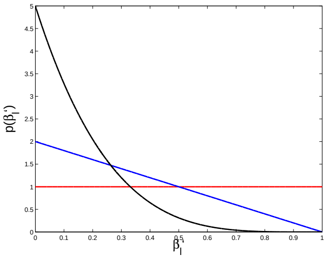
$$p(x \mid \beta, \Lambda_1, \Lambda_2, \ldots) = \sum_{k=1}^{\infty} \beta_k \mathcal{N}(x; 0, \Lambda_k)$$

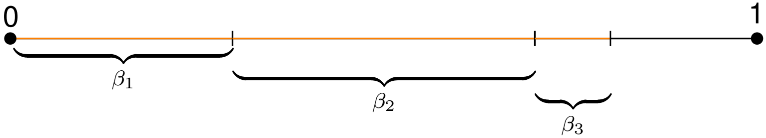
Control complexity using a *stick-breaking prior* for mixture weights:

$$\beta_k = \beta'_k \prod_{\ell=1}^{k-1} (1 - \beta'_\ell)$$

$$\beta'_\ell \sim \mathrm{Beta}(1, \gamma)$$

γ : concentration parameter





Dirichlet Process Mixtures

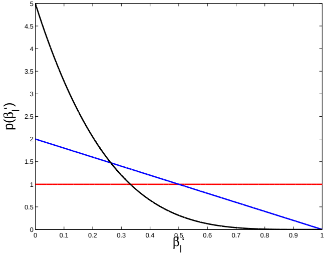
$$p(x \mid \beta, \Lambda_1, \Lambda_2, \ldots) = \sum_{k=1}^{\infty} \beta_k \mathcal{N}(x; 0, \Lambda_k)$$

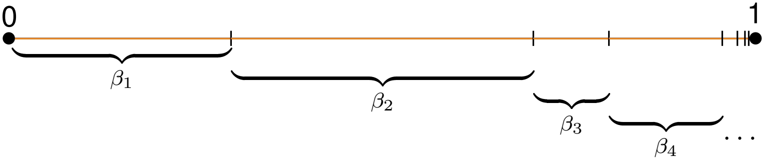
Control complexity using a *stick-breaking prior* for mixture weights:

$$\beta_k = \beta'_k \prod_{\ell=1}^{k-1} (1 - \beta'_\ell)$$

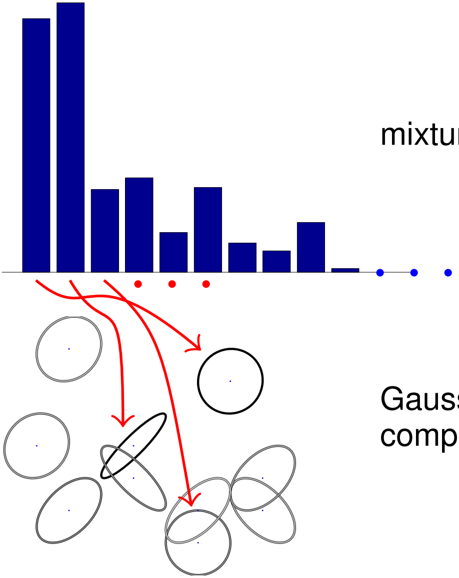
$$\beta'_\ell \sim \mathrm{Beta}(1, \gamma)$$

γ : concentration parameter





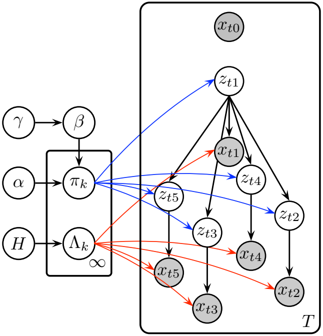
Dirichlet Process Mixtures

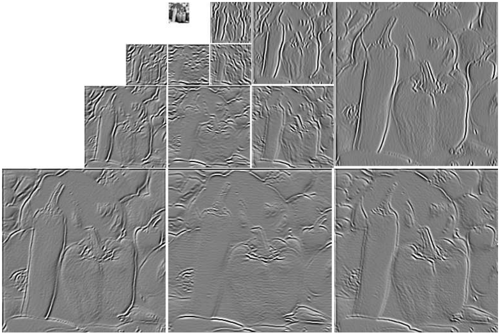


mixture weights (β_k)

Gaussian mixture components

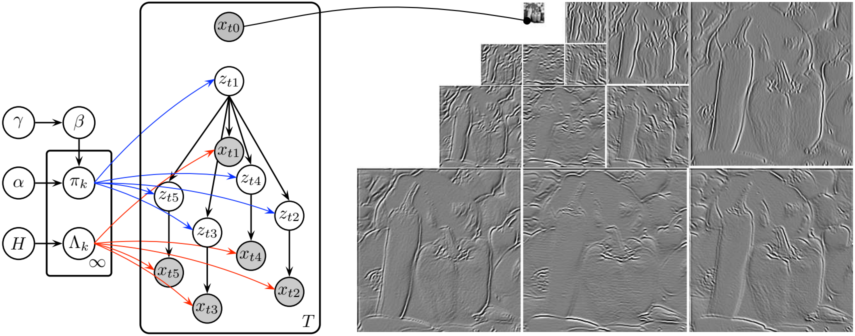
Hierarchical Dirichlet Process-Hidden Markov Tree





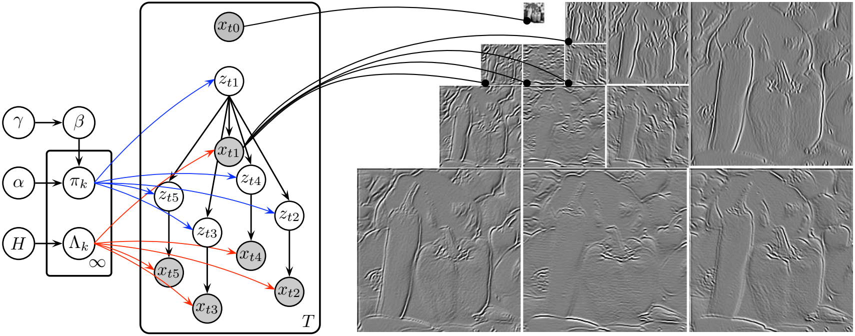
- Hidden states z_{ti} generate *vectors* of wavelet coefficients x_{ti} at *multiple* orientations.
- Wavelet coefficients marginally distributed as potentially *infinite* Dirichlet Process mixtures
- Hierarchical Dirichlet Process prior allows us to learn a potentially infinite set of *appearance patterns* from natural images

Hierarchical Dirichlet Process-Hidden Markov Tree



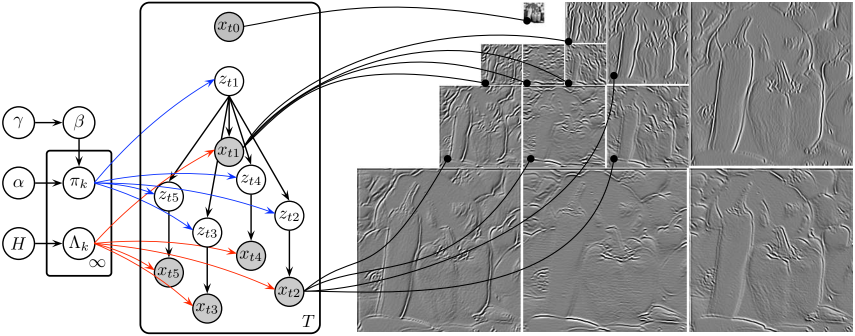
- Hidden states z_{ti} generate *vectors* of wavelet coefficients x_{ti} at *multiple* orientations.
- Wavelet coefficients marginally distributed as potentially *infinite* Dirichlet Process mixtures
- Hierarchical Dirichlet Process prior allows us to learn a potentially infinite set of *appearance patterns* from natural images

Hierarchical Dirichlet Process-Hidden Markov Tree



- Hidden states z_{ti} generate *vectors* of wavelet coefficients x_{ti} at *multiple* orientations.
- Wavelet coefficients marginally distributed as potentially *infinite* Dirichlet Process mixtures
- Hierarchical Dirichlet Process prior allows us to learn a potentially infinite set of *appearance patterns* from natural images

Hierarchical Dirichlet Process-Hidden Markov Tree



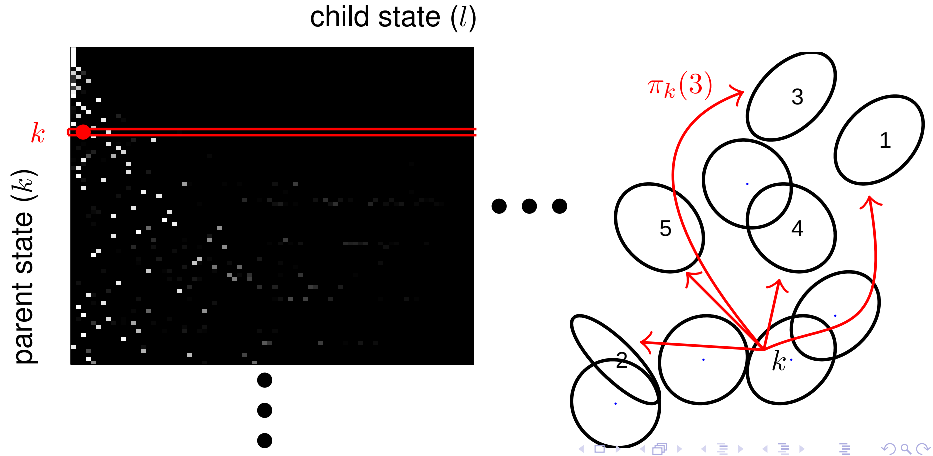
- Hidden states z_{ti} generate *vectors* of wavelet coefficients x_{ti} at *multiple* orientations.
- Wavelet coefficients marginally distributed as potentially *infinite* Dirichlet Process mixtures
- Hierarchical Dirichlet Process prior allows us to learn a potentially infinite set of *appearance patterns* from natural images

Need for *Hierarchical* Dirichlet Processes

- Each row of the transition matrix is a set of mixture proportions

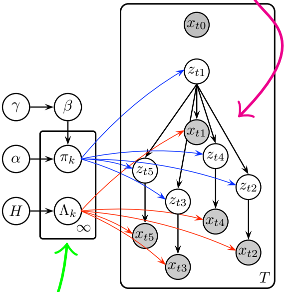
$$\pi_k(l) = Pr(z_{ti} = l \mid z_{\mathrm{Pa}(ti)} = k) \qquad k, l \in [1, \ldots, \infty)$$

- Hierarchical coupling ensures that child states reused by multiple parents



Learning HDP-HMT with a Collapsed Gibbs Sampler

Sample hidden assigments z



Consider

- states assigned to existing clusters
- a potential new state

(state cardinality determination)

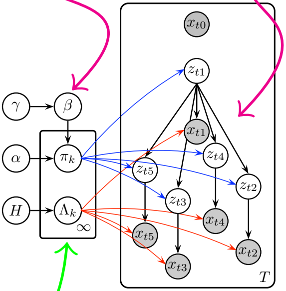
Marginalize state-specific parameters $\{\pi, \theta\}$

Rao–Blackwellization improves the efficiency and accuracy of MCMC methods.

Learning HDP-HMT with a Collapsed Gibbs Sampler

Sample global mixture proportions β

Sample hidden assigments z



Consider

- states assigned to existing clusters
- a potential new state

(state cardinality determination)

Marginalize state-specific parameters $\{\pi, \theta\}$

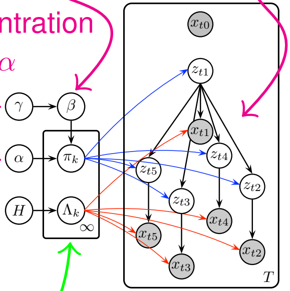
Rao–Blackwellization improves the efficiency and accuracy of MCMC methods.

Learning HDP-HMT with a Collapsed Gibbs Sampler

Sample global mixture proportions β

Sample hidden assigments z

Sample concentration parameters γ, α



Consider

- states assigned to existing clusters
- a potential new state

(state cardinality determination)

Marginalize state-specific parameters $\{\pi, \theta\}$

Rao–Blackwellization improves the efficiency and accuracy of MCMC methods.

The Overall Learning Algorithm to Denoise Images

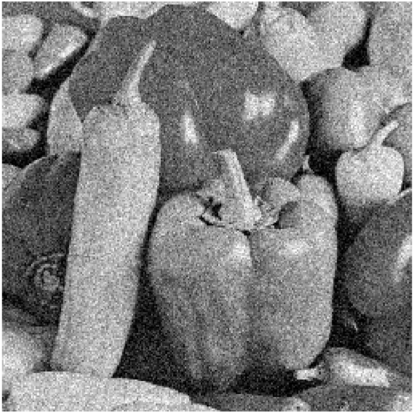
- Given a noisy image, learn HDP-HMT parameters by running the collapsed Gibbs sampler
- After burn-in, collect S samples $\theta^{(s)} = \{\pi_k^{(s)}, \Lambda_k^{(s)}\}_{k=1}^{K_s}$.
- Given each collected sample, a denoised image can be computed in closed form:

$$\mathbb{E}\big[w_{ti} \mid \mathbf{x}, \theta^{(s)}\big] = \sum_{k=1}^{K_s} \overbrace{p(z_{ti} = k \mid \mathbf{x}, \theta^{(s)})}^{\text{marginals from Belief Propagation}} \underbrace{\mathbb{E}\big[w_{ti} \mid x_{ti}, \Lambda_k^{(s)}\big]}_{\text{linear regression}}$$

- Average over samples of varying complexity to get posterior mean of detail coefficients

Results: Peppers, $\sigma = 25$

Binary HMT (Daubechies-4 Orthogonal Wavelets)

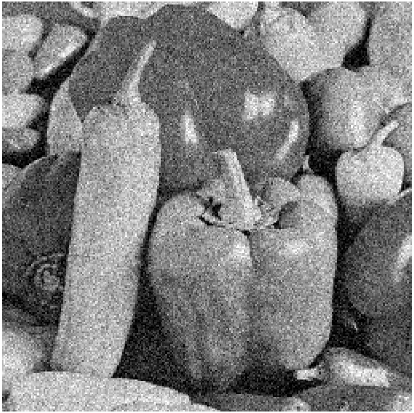


INPUT (20.18 dB)

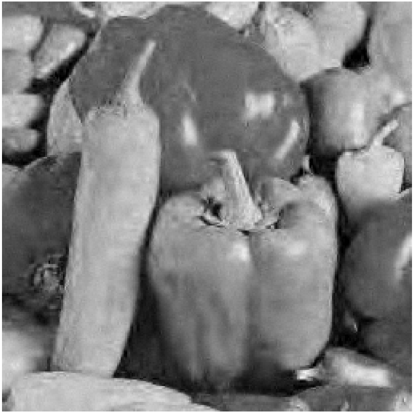
OUTPUT (27.36 dB)

Results: Peppers, $\sigma = 25$

HDP-HMT (Daubechies-4 Orthogonal Wavelets)



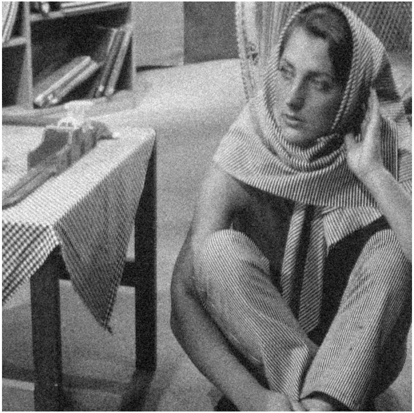
INPUT (20.18 dB)



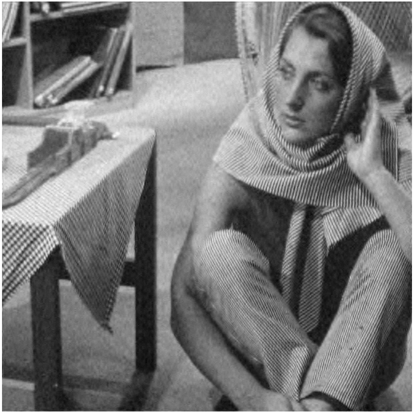
OUTPUT (28.09 dB)

Results: Barbara, $\sigma = 15$

Binary HMT (Daubechies-4 Orthogonal Wavelets)



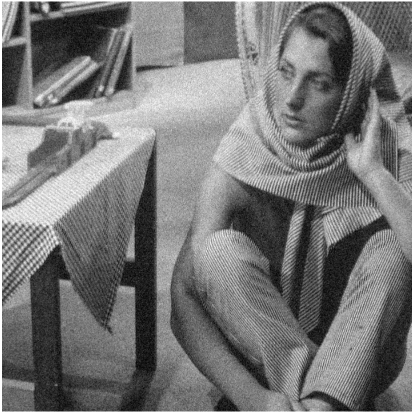
INPUT (24.61 dB)



OUTPUT (29.35 dB)

Results: Barbara, $\sigma = 15$

HDP-HMT (Daubechies-4 Orthogonal Wavelets)

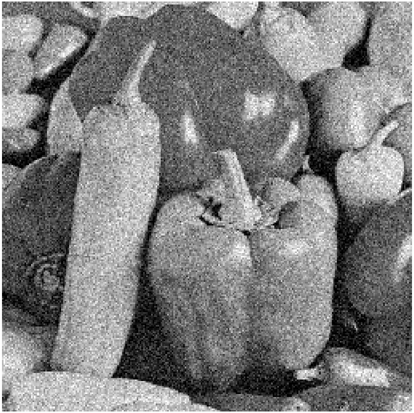


INPUT (24.61 dB)

OUTPUT (30.26 dB)

Results: Peppers, $\sigma = 25$

Local GSM by Portilla et. al. (2003) (7^{th}-Order Steerable Pyramids)

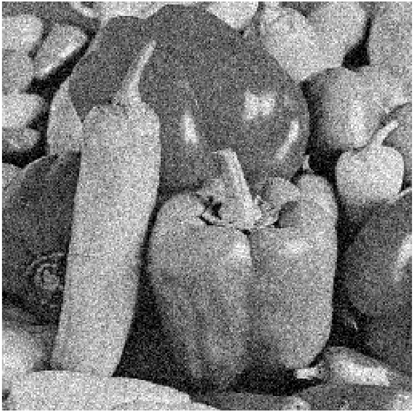


INPUT (20.18 dB)

OUTPUT (29.56 dB)

Results: Peppers, $\sigma = 25$

HDP-HMT (7^{th}-Order Steerable Pyramids)$\langle$Daubechies-4 Orthogonal Wavelets$\rangle$

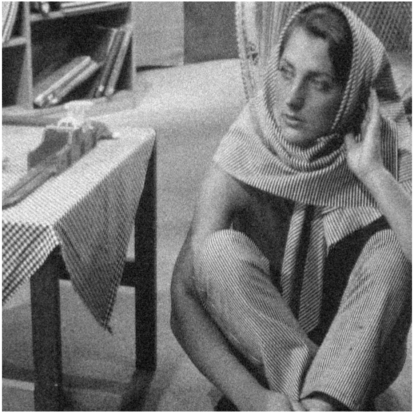


INPUT (20.18 dB)

OUTPUT (29.44 dB)$\langle$28.09 dB$\rangle$

Results: Barbara, $\sigma = 15$

Local GSM (7^{th}-Order Steerable Pyramids)



INPUT (24.61 dB)

OUTPUT (31.84dB)

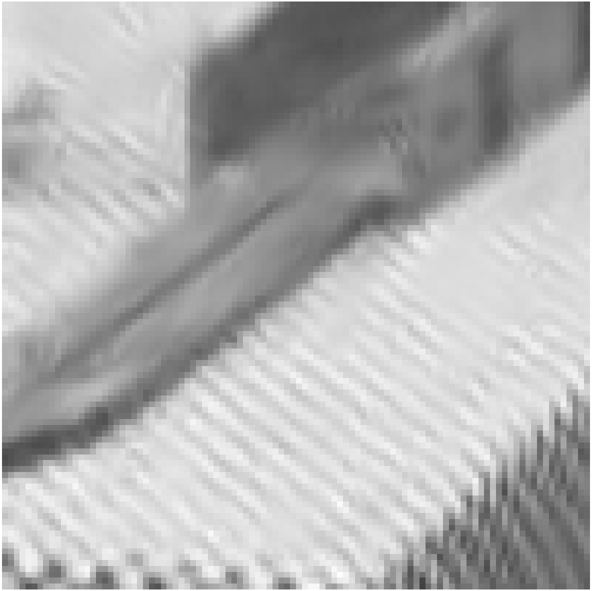
Results: Barbara, $\sigma = 15$

HDP-HMT (7^{th}-Order Steerable Pyramids)⟨Daubechies-4 Orthogonal Wavelets⟩

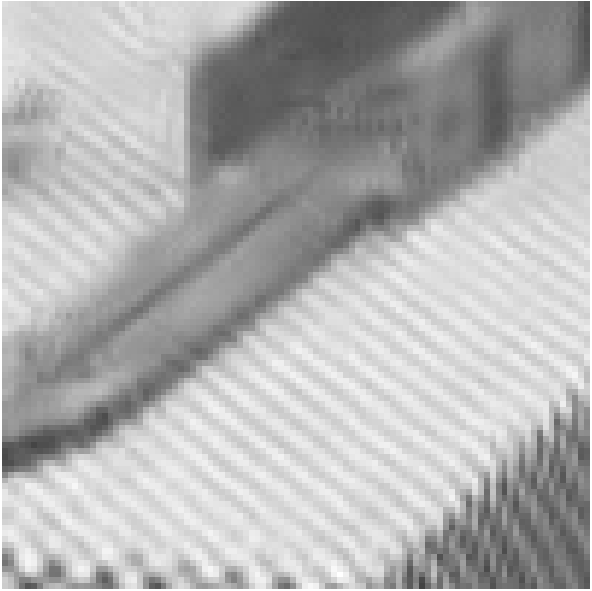
INPUT (24.61 dB)

OUTPUT (31.83 dB)⟨30.26 dB⟩

Local GSM

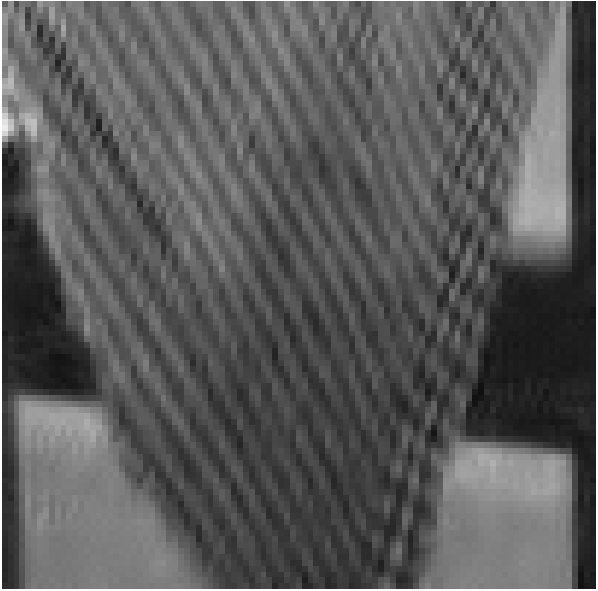


HDP-HMT



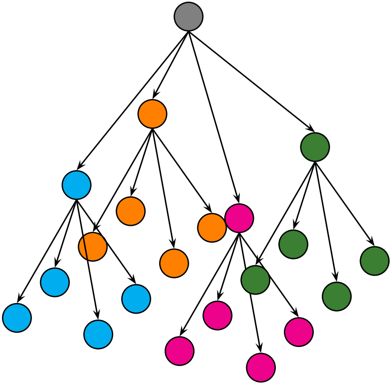
Local GSM

HDP-HMT

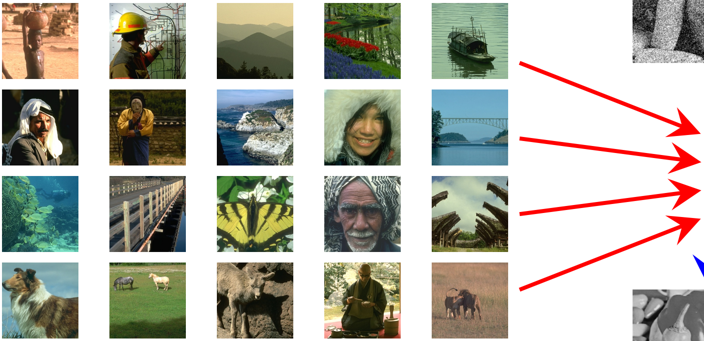


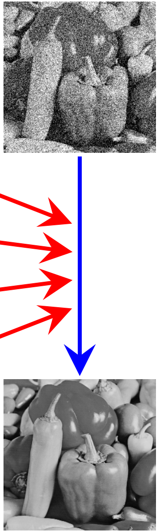
Summary

- Introduced a *nonparametric* Bayesian model for multiscale datasets with complex, non-local dependencies
- Developed an MCMC method which learns the potentially *infinite* set of HDP-HMT parameters
- Demonstrated effective *image denoising* via an empirical Bayesian approach



Ongoing Work





- Learn models from hundreds of natural images (to appear at ICCV 2007)
- Apply learned statistics of *clean* images to *denoise* images

http://www.cs.berkeley.edu/~kivinen

Joint Statistics of HDP-HMT Simulations

Peppers (Steerable Pyramid (Simoncelli et al. 1992) Detail Coefficients)