Learning High-Order MRF Priors of Color Images

Julian J. McAuley (Presenting)

National ICT Australia, Canberra, ACT 0200, Australia
University of New South Wales, Sydney, NSW 2052, Australia
j.mcauley@student.unsw.edu.au

Tibério S. Caetano

National ICT Australia, Canberra, ACT 0200, Australia
RSISE, Australian National University, Canberra, ACT 0200, Australia
tiberio.caetano@nicta.com.au

Alex J. Smola

National ICT Australia, Canberra, ACT 0200, Australia
RSISE, Australian National University, Canberra, ACT 0200, Australia
alex.smola@nicta.com.au

Matthias O. Franz

Max Planck Institute for Biological Cybernetics, Tuebingen 72012, Germany
matthias.franz@tuebingen.mpg.de

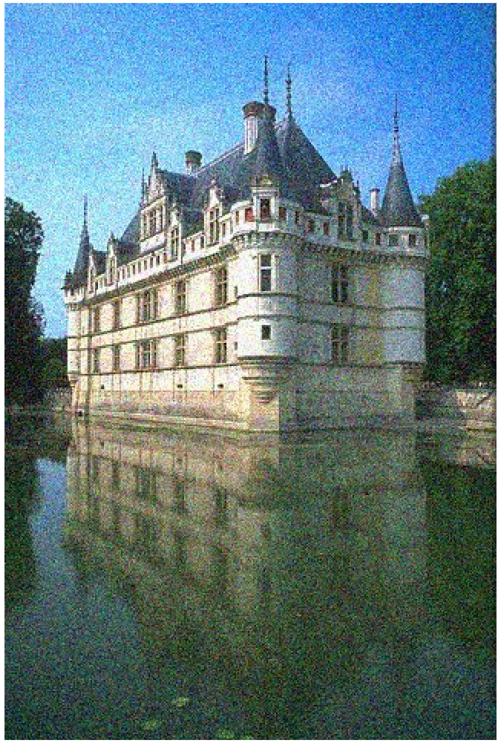
Image Denoising...

Image Denoising...

Original Image

Image Denoising...

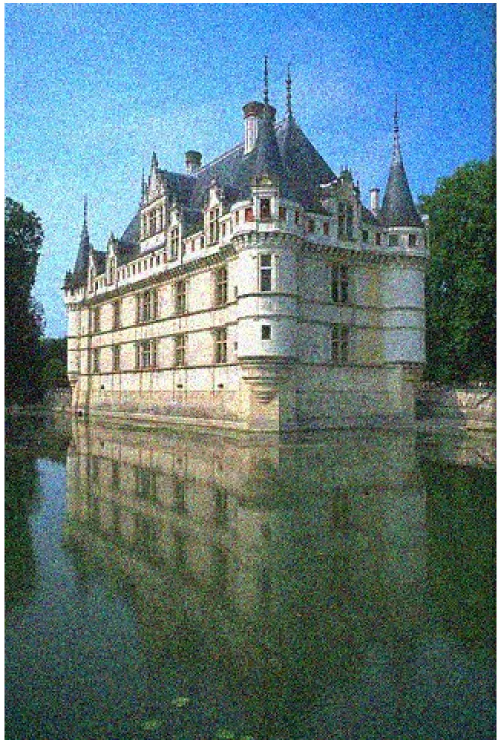
Original Image



Noisy Image

Image Denoising...

Original Image

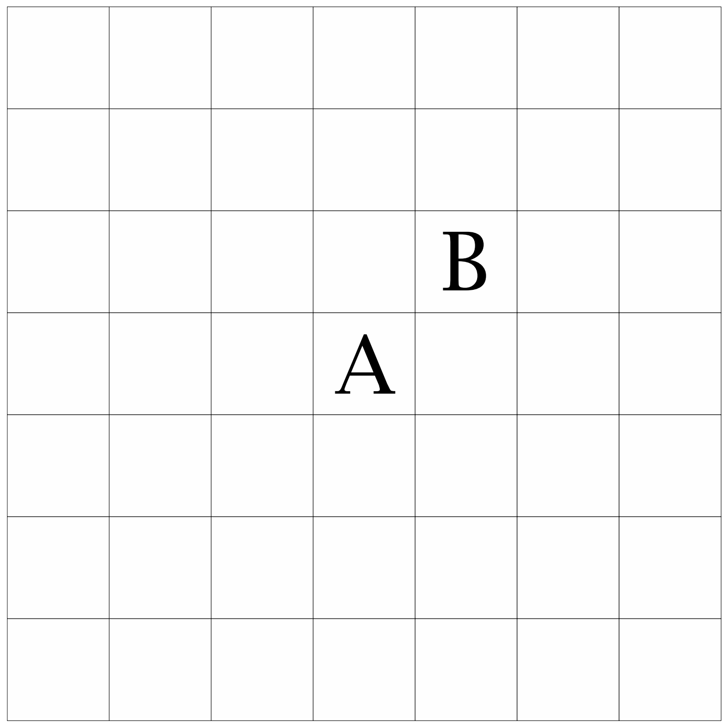


Noisy Image

Inferred Image

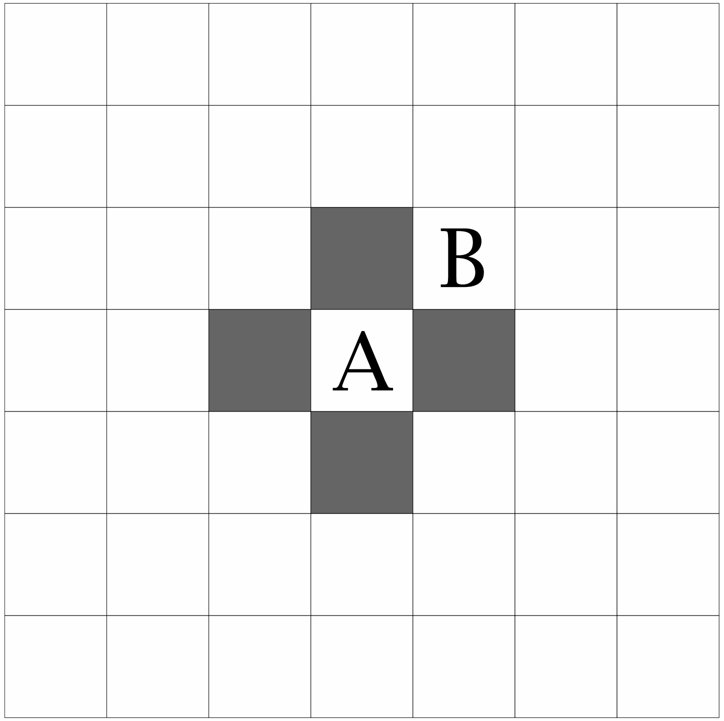
Conditional Independence...

Conditional Independence...



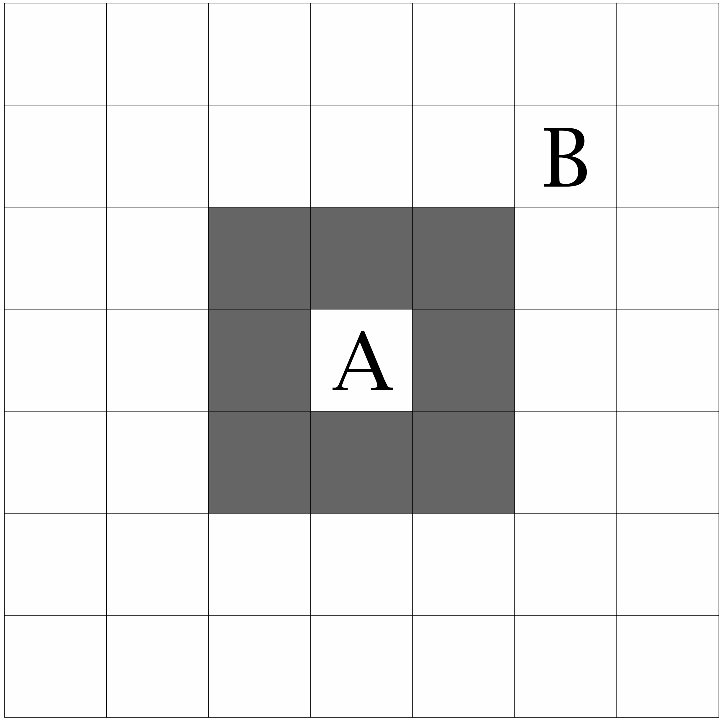
How does the pixel ‘A’ depend on the pixel ‘B’?

Conditional Independence...



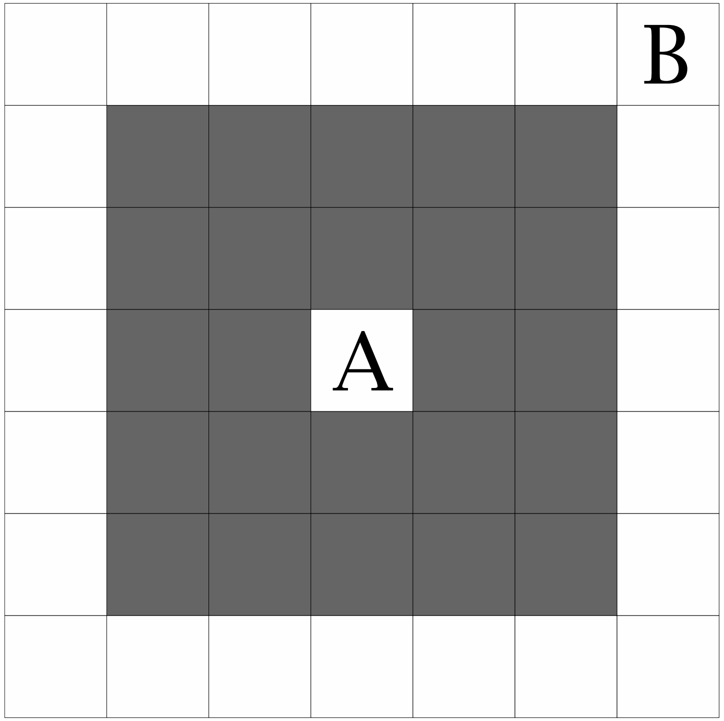
'A' and 'B' are *conditionally independent*, given the grey pixels.

Conditional Independence...



'A' and 'B' are *conditionally independent*, given the grey pixels.

Conditional Independence...



'A' and 'B' are *conditionally independent*, given the grey pixels.

Markov Random Fields...

Markov Random Fields...

- Our independence assumption allows us to define a MRF for images.

Markov Random Fields...

- Our independence assumption allows us to define a MRF for images.
- The maximal cliques are just 3x3 (or larger) regions, plus a pairwise clique for the noise.

Markov Random Fields...

- Our independence assumption allows us to define a MRF for images.
- The maximal cliques are just 3x3 (or larger) regions, plus a pairwise clique for the noise.
- If we have potentials, $\phi_c(\mathbf{x}_c)$ for each of these cliques, the Hammersley-Clifford theorem allows us to define the probability distribution for the whole field:

Markov Random Fields...

- Our independence assumption allows us to define a MRF for images.

- The maximal cliques are just 3x3 (or larger) regions, plus a pairwise clique for the noise.

- If we have potentials, $\phi_c(\mathbf{x}_c)$ for each of these cliques, the Hammersley-Clifford theorem allows us to define the probability distribution for the whole field:

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c).$$

Markov Random Fields...

- Our independence assumption allows us to define a MRF for images.
- The maximal cliques are just 3x3 (or larger) regions, plus a pairwise clique for the noise.
- If we have potentials, $\phi_c(\mathbf{x}_c)$ for each of these cliques, the Hammersley-Clifford theorem allows us to define the probability distribution for the whole field:

$$p(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c).$$

- The problem now is just to define the potential functions.

Roth and Black's Model...

Roth and Black's Model...

- In 2005, Stefan Roth and Michael J. Black defined a MRF model for natural images.

Roth and Black's Model...

- In 2005, Stefan Roth and Michael J. Black defined a MRF model for natural images.

- In their model, the potential functions take the form of a *product of experts*, in which each expert is the response to a particular filter.

Roth and Black's Model...

- In 2005, Stefan Roth and Michael J. Black defined a MRF model for natural images.

- In their model, the potential functions take the form of a *product of experts*, in which each expert is the response to a particular filter.

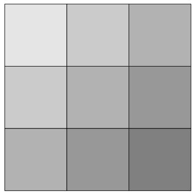
- Their 'experts' each take the form of a Student's T-distribution

$$\phi_c(\mathbf{x}_c; J, \alpha) = \prod_{f=1}^{F} (1 + \frac{1}{2} \langle J_f, \mathbf{x}_c \rangle^2)^{-\alpha_f}.$$

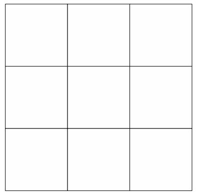
‘Filtering’...

'Filtering'...

- Given a filter, J_f, and a clique, $\mathbf{x}_c$,



filter



clique

‘Filtering’...

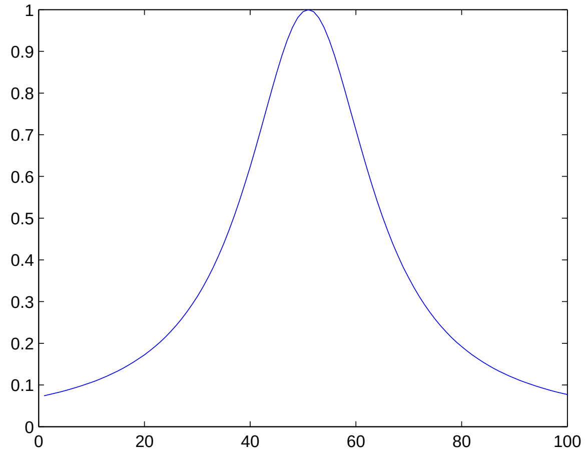
- Given a filter, J_f, and a clique, $\mathbf{x}_c$,

filter *clique*

- The ‘response’ of a clique to that filter is a function of their inner product.

'Filtering'...

- For the Student's T-distribution, we get something like:



The product of experts model...

- This value is high when the filter and the clique are **less** coincident, so the filters and the alphas must be selected accordingly.

The product of experts model...

- This value is high when the filter and the clique are **less** coincident, so the filters and the alphas must be selected accordingly.

- In theory, the number of filters we require should be equal to the dimensionality of our cliques – this way they are able to 'span' the entire range of pixel configurations.

The product of experts model...

- This value is high when the filter and the clique are **less** coincident, so the filters and the alphas must be selected accordingly.

- In theory, the number of filters we require should be equal to the dimensionality of our cliques – this way they are able to 'span' the entire range of pixel configurations.

- In practice, Roth and Black found that the 'most important' filter corresponds to a constant gray, which can be ignored.

The product of experts model...

- This value is high when the filter and the clique are **less** coincident, so the filters and the alphas must be selected accordingly.

- In theory, the number of filters we require should be equal to the dimensionality of our cliques – this way they are able to 'span' the entire range of pixel configurations.

- In practice, Roth and Black found that the 'most important' filter corresponds to a constant gray, which can be ignored.

- Hence, for cliques of size 3x3, we need to learn $8 \times 9 + 8 = 80$ parameters. For cliques of size 5x5, we need to learn $24 \times 25 + 24 = 624$ parameters.

The product of experts model...

- This value is high when the filter and the clique are **less** coincident, so the filters and the alphas must be selected accordingly.
- In theory, the number of filters we require should be equal to the dimensionality of our cliques – this way they are able to 'span' the entire range of pixel configurations.
- In practice, Roth and Black found that the 'most important' filter corresponds to a constant gray, which can be ignored.
- Hence, for cliques of size 3x3, we need to learn $8 \times 9 + 8 = 80$ parameters. For cliques of size 5x5, we need to learn $24 \times 25 + 24 = 624$ parameters.
- Roth and Black used a *Contrastive Divergence Learning* approach to learn the correct filters, based on the statistics of a large database of natural images.

Inference in the product of experts model...

Inference in the product of experts model...

- There are many inference techniques for MRFs, but for image denoising, gradient-ascent is a good choice.

Inference in the product of experts model...

- There are many inference techniques for MRFs, but for image denoising, gradient-ascent is a good choice.

- This makes sense for a denoising problem, since the noisy image is 'close to' the denoised image.

Inference in the product of experts model...

- There are many inference techniques for MRFs, but for image denoising, gradient-ascent is a good choice.

- This makes sense for a denoising problem, since the noisy image is 'close to' the denoised image.

- The gradient-ascent update equation is just

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \delta \frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}|\mathbf{y}),$$

  where $\mathbf{x}^{t+1}$ is the updated image, $\mathbf{x}^t$ is the previous image, δ is a learning rate, and $\mathbf{y}$ is the 'noisy' image.

Inference in the product of experts model...

- There are many inference techniques for MRFs, but for image denoising, gradient-ascent is a good choice.

- This makes sense for a denoising problem, since the noisy image is 'close to' the denoised image.

- The gradient-ascent update equation is just

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \delta \frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}|\mathbf{y}),$$

  where $\mathbf{x}^{t+1}$ is the updated image, $\mathbf{x}^t$ is the previous image, δ is a learning rate, and $\mathbf{y}$ is the 'noisy' image.

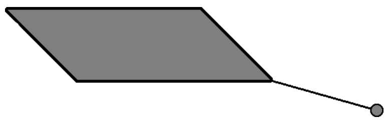
- And finally, the gradient of the log posterior is

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) = \sum_{f=1}^{F} \alpha_f J_f^- * \frac{(J_f * \mathbf{x})}{1 + \frac{1}{2}(J_f * \mathbf{x})^2} + \frac{\lambda}{\sigma^2}(\mathbf{y} - \mathbf{x}).$$

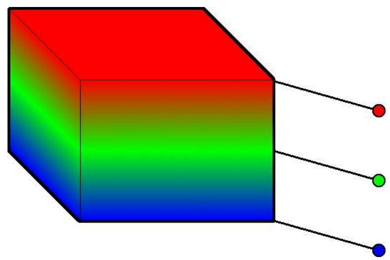
Extending the model to *color* images...

Extending the model to ***color*** images...

- Our 3x3 cliques now become 3x3x3 cliques, and our 5x5 cliques now become 5x5x3 cliques.



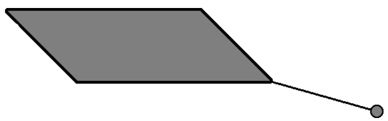
3x3 or 5x5 clique



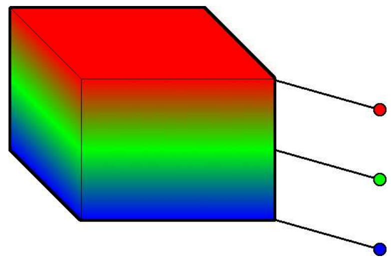
3x3x3 or 5x5x3 clique

Extending the model to *color* images...

- Our 3x3 cliques now become 3x3x3 cliques, and our 5x5 cliques now become 5x5x3 cliques.



3x3 or 5x5 clique

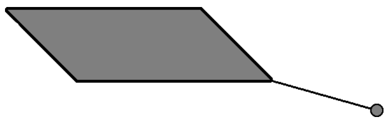


3x3x3 or 5x5x3 clique

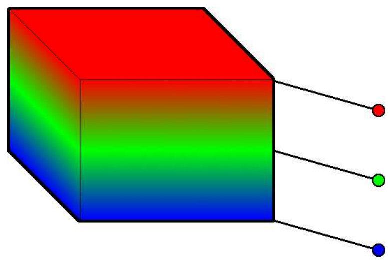
- For the 3x3 model, we now need to learn 26×27 dimensional filters, and for the 5x5 model, we need to learn 74×75 dimensional filters.

Extending the model to *color* images...

- Our 3x3 cliques now become 3x3x3 cliques, and our 5x5 cliques now become 5x5x3 cliques.



3x3 or 5x5 clique



3x3x3 or 5x5x3 clique

- For the 3x3 model, we now need to learn 26×27 dimensional filters, and for the 5x5 model, we need to learn 74×75 dimensional filters.

- A simpler learning approach is required.

Learning the color model...

Learning the color model...

- Rather than using contrastive divergence learning, we simply applied principal component analysis to a large sample of natural image patches.

Learning the color model...

- Rather than using contrastive divergence learning, we simply applied principal component analysis to a large sample of natural image patches.
- These image patches were found by randomly cropping 3x3 and 5x5 regions from images in the Berkeley Segmentation Database.

a training image

randomly cropped patches

Learning the color model...

- This tells us the components in natural images that *vary the most*.

Learning the color model...

- This tells us the components in natural images that *vary the most*.

- Although we ourselves were surprised that this worked well, it seems like a sensible choice in lieu of the true Maximum-Likelihood solution.

Learning the color model...

- As well as learning the filters themselves, we had to learn the alphas. These are critical components that gauge the 'shape' of the Student's T-distribution – i.e. the 'importance' of each filter.

Learning the color model...

- As well as learning the filters themselves, we had to learn the alphas. These are critical components that gauge the 'shape' of the Student's T-distribution – i.e. the 'importance' of each filter.

- Again, we tried to use gradient-based approaches, to find the most likely alphas, given our filters, and our database of training images.

Learning the color model...

- As well as learning the filters themselves, we had to learn the alphas. These are critical components that gauge the 'shape' of the Student's T-distribution – i.e. the 'importance' of each filter.

- Again, we tried to use gradient-based approaches, to find the most likely alphas, given our filters, and our database of training images.

- Unfortunately, gradient-ascent in MRFs requires sampling from the posterior distribution, which is a *very* costly procedure.

Learning the color model...

- As well as learning the filters themselves, we had to learn the alphas. These are critical components that gauge the 'shape' of the Student's T-distribution – i.e. the 'importance' of each filter.

- Again, we tried to use gradient-based approaches, to find the most likely alphas, given our filters, and our database of training images.

- Unfortunately, gradient-ascent in MRFs requires sampling from the posterior distribution, which is a *very* costly procedure.

- However, during the *first* iteration, the posterior distribution is flat, and sampling is easy.

Learning the color model...

- Surprisingly, we found that after the first iteration, the *relative* weights of the alphas appeared to remain the same.

Learning the color model...

- Surprisingly, we found that after the first iteration, the *relative* weights of the alphas appeared to remain the same.

- This meant that one iteration of gradient ascent was sufficient, and there was no need to perform sampling.

Learning the color model...

- Surprisingly, we found that after the first iteration, the *relative* weights of the alphas appeared to remain the same.

- This meant that one iteration of gradient ascent was sufficient, and there was no need to perform sampling.

- With a learning procedure this simple, it is now trivial to learn monochromatic or color models of natural images, even if the clique size is extremely large.

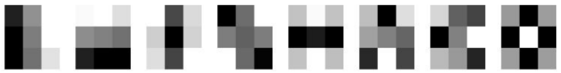
What we learnt...

What we learnt...

- 3x3 monochromatic model:

What we learnt...

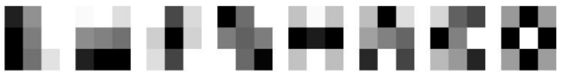
- 3x3 monochromatic model:



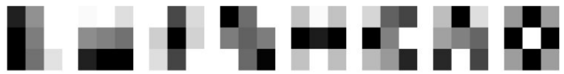
Filters, sorted according to eigenvalue.

What we learnt...

- 3x3 monochromatic model:



Filters, sorted according to eigenvalue.



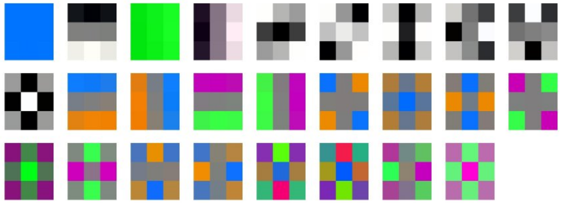
Filters, sorted by importance, after learning.

What we learnt...

- 3x3 color model:

What we learnt...

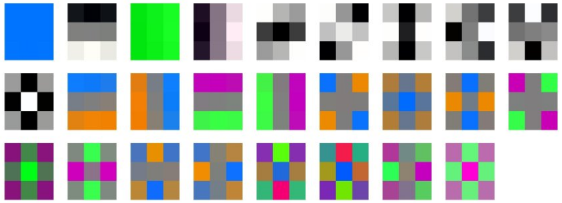
- 3x3 color model:



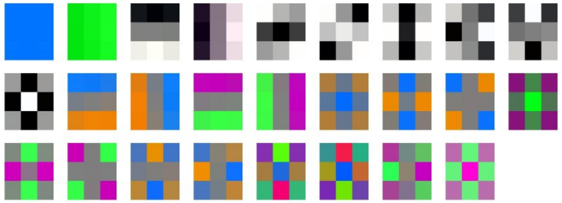
Filters, sorted according to eigenvalue.

What we learnt...

- 3x3 color model:



Filters, sorted according to eigenvalue.



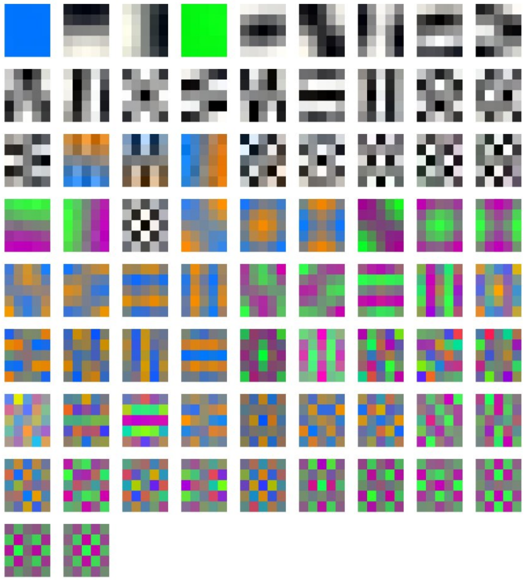
Filters, sorted by importance, after learning.

What we learnt...

- 5x5 color model:

What we learnt...

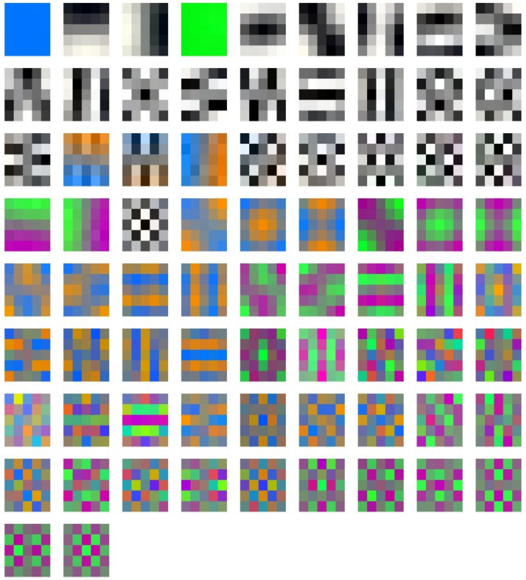
- 5x5 color model:



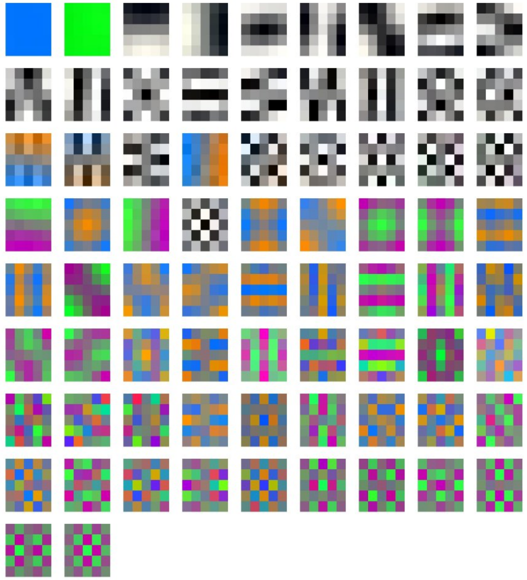
Filters, sorted according to eigenvalue.

What we learnt...

- 5x5 color model:



Filters, sorted according to eigenvalue.



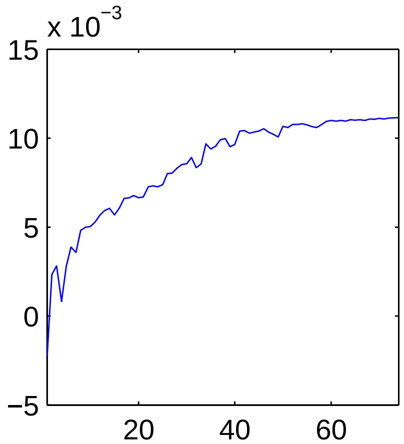
Filters, sorted by importance, after learning.

What we learnt...

- 5x5 color model:

What we learnt...

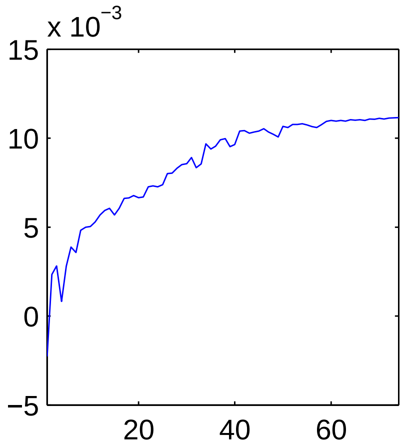
- 5x5 color model:



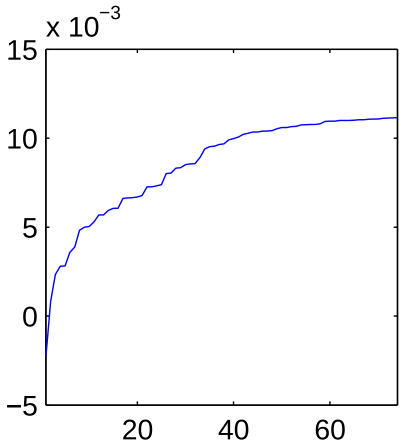
A plot of the alphas, sorted by eigenvalue.

What we learnt...

- 5x5 color model:



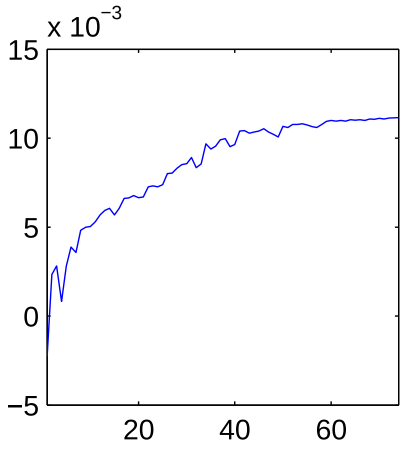
A plot of the alphas, sorted by eigenvalue.



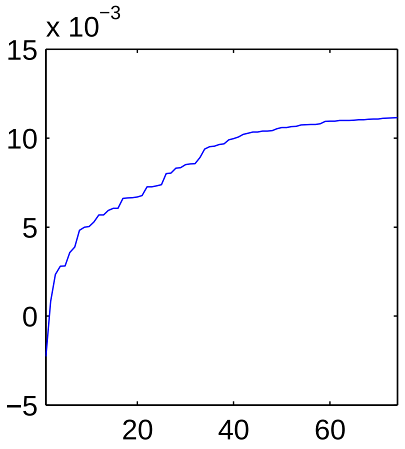
Sorted by importance, after learning.

What we learnt...

- 5x5 color model:



A plot of the alphas, sorted by eigenvalue.



Sorted by importance, after learning.

- The fact that these two plots are different demonstrates the need to perform learning by maximum-likelihood.

Results...

Results...

- Our first experiment involved applying an unequal amount of noise to each channel.

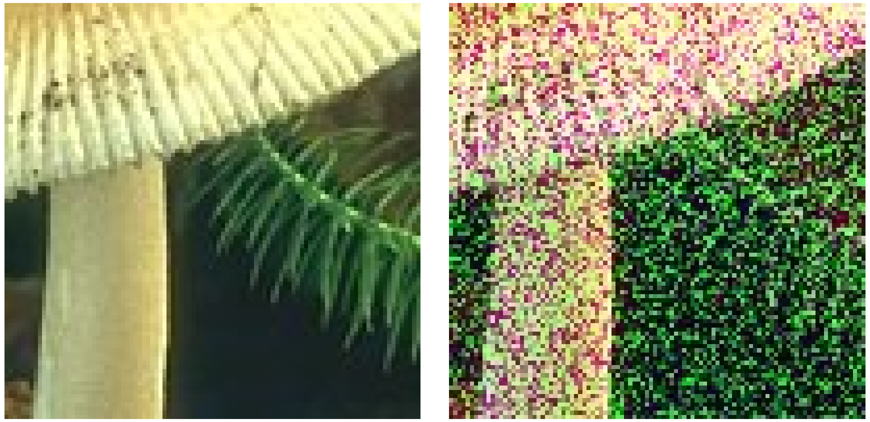
The original image...

...Is corrupted with $\sigma = 128$, in the **green** channel only.

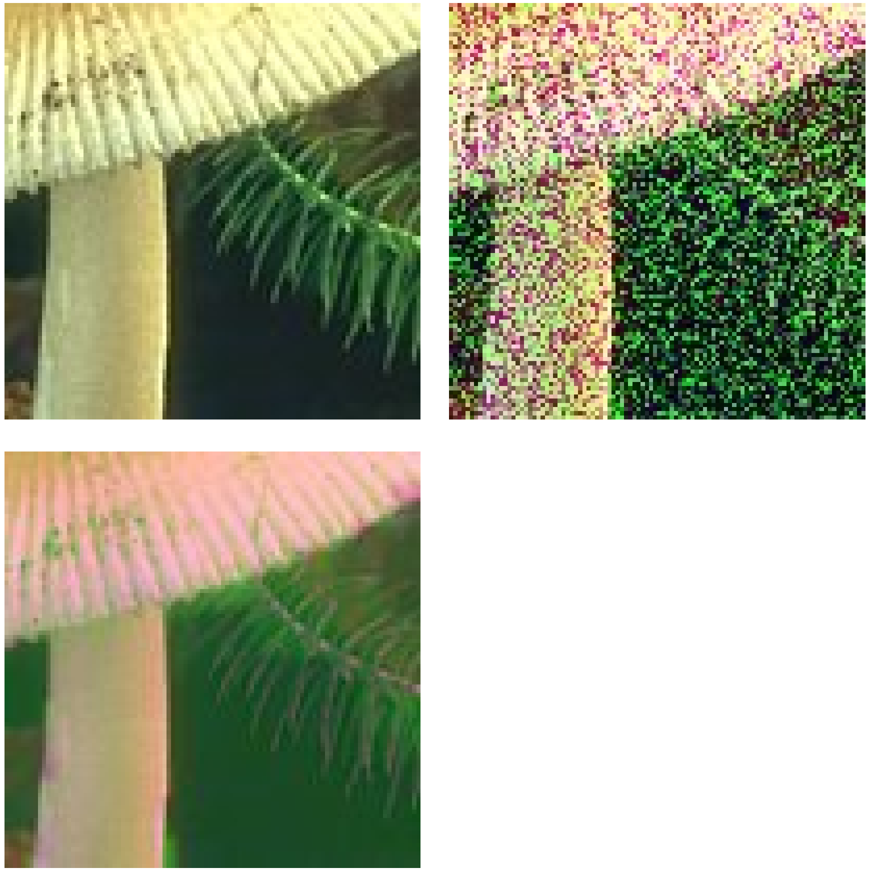
Denoised, using a model trained independently on each channel.

Denoised, using a model trained on all channels simultaneously.

The original image...

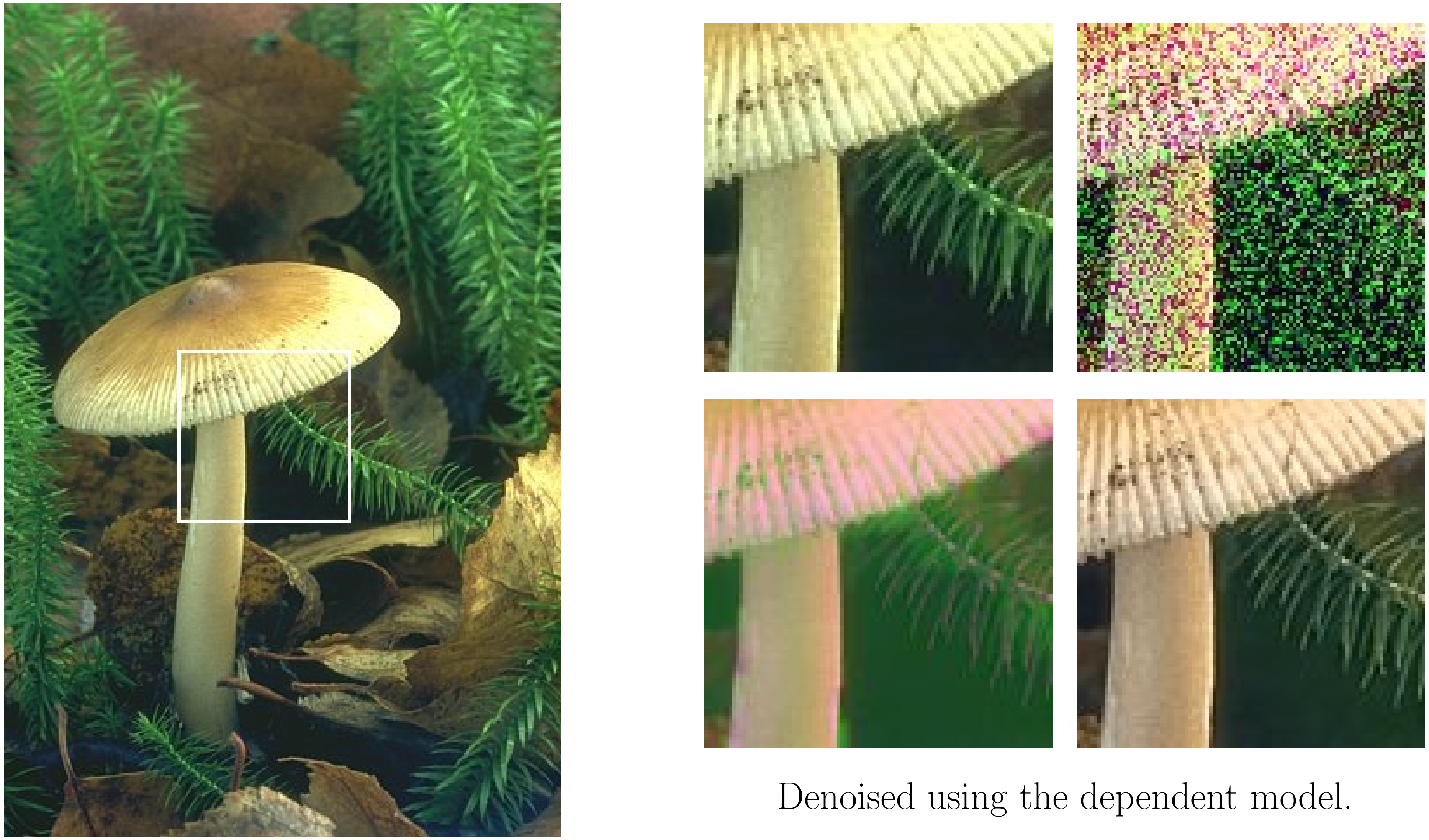


The corrupted image.



Denoised using the independent model.

Learning High-Order MRF Priors of Color Images



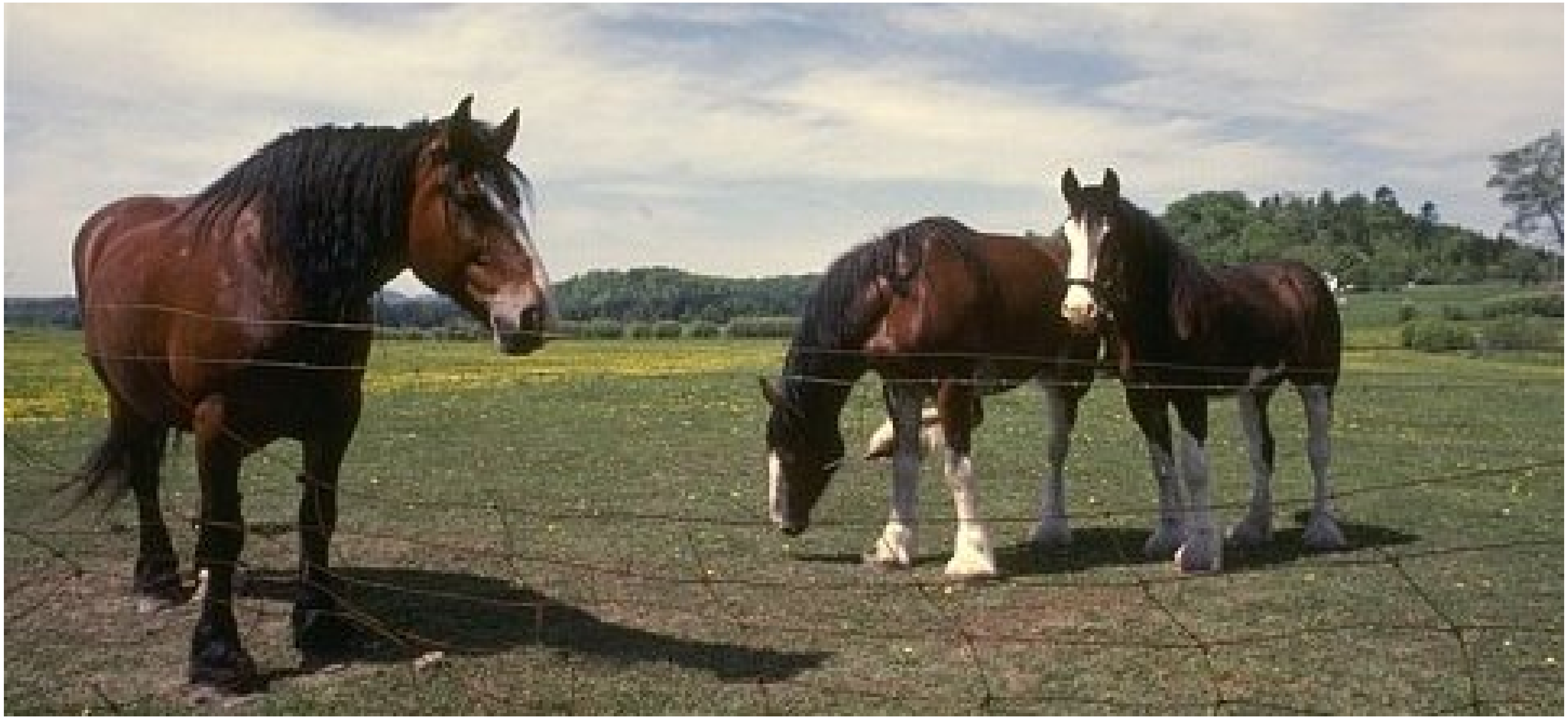
Denoised using the dependent model.

Results...

- Similarly, we can apply a different amount of noise to *every* channel.

Results...

- Similarly, we can apply a different amount of noise to *every* channel.
- The next image has $\sigma = 128$ in the red channel, $\sigma = 15$ in the green channel, $\sigma = 5$ in the blue channel.



The original image...

Learning High-Order MRF Priors of Color Images

...Is corrupted with $\sigma = 128$ (red), 15 (green), and 5 (blue).

Denoised, using a model trained independently on each channel.

Denoised, using a model trained on all channels simultaneously.

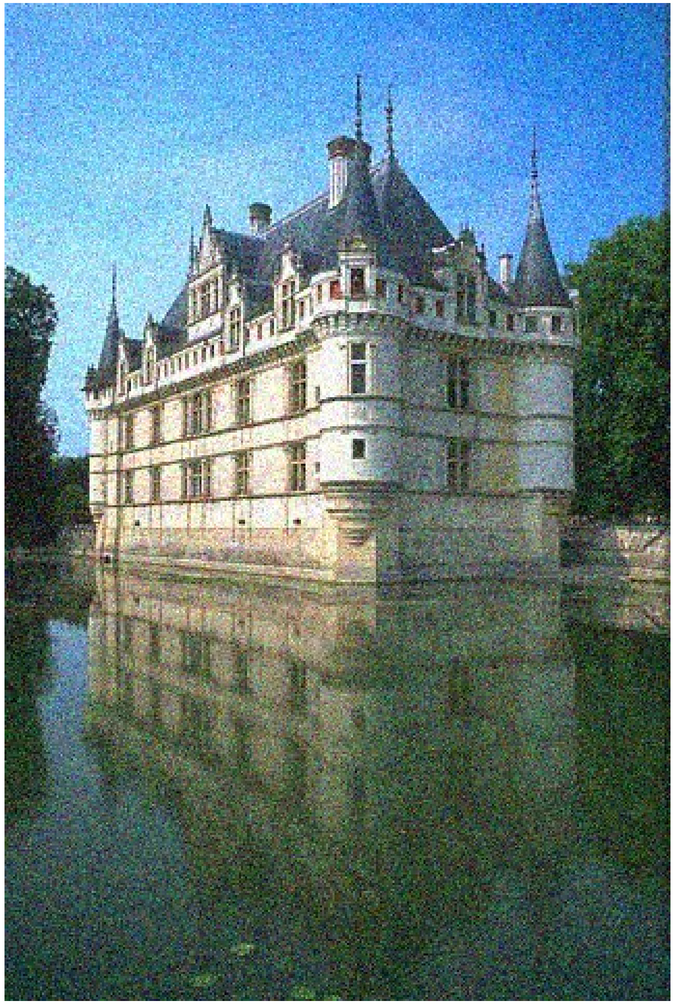
Results...

- Although that worked pretty well, it is now very difficult to tune the gradient-ascent parameters (e.g. learning rate), so the results may not be optimal.

Results...

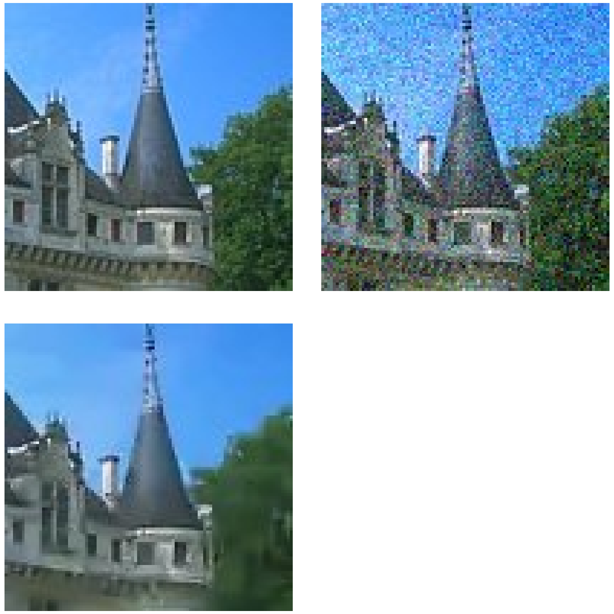
- Finally, we apply *equal* noise to each channel, and compare our results to the state-of-the-art.

The original image...

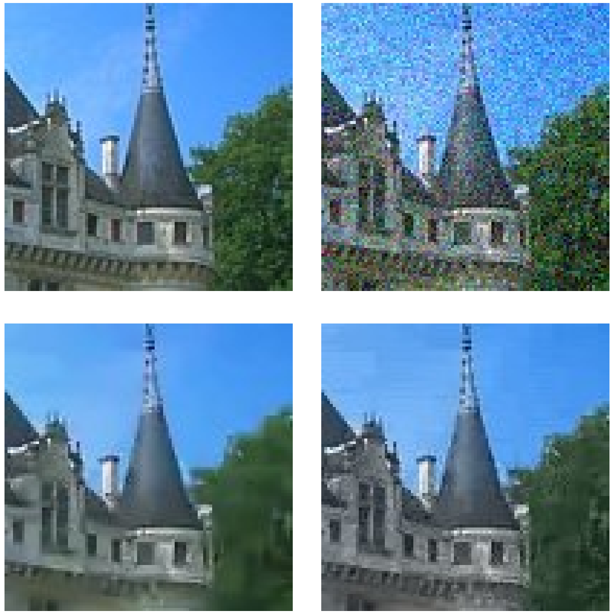


...Is corrupted with $\sigma = 25$ in all channels.

Learning High-Order MRF Priors of Color Images

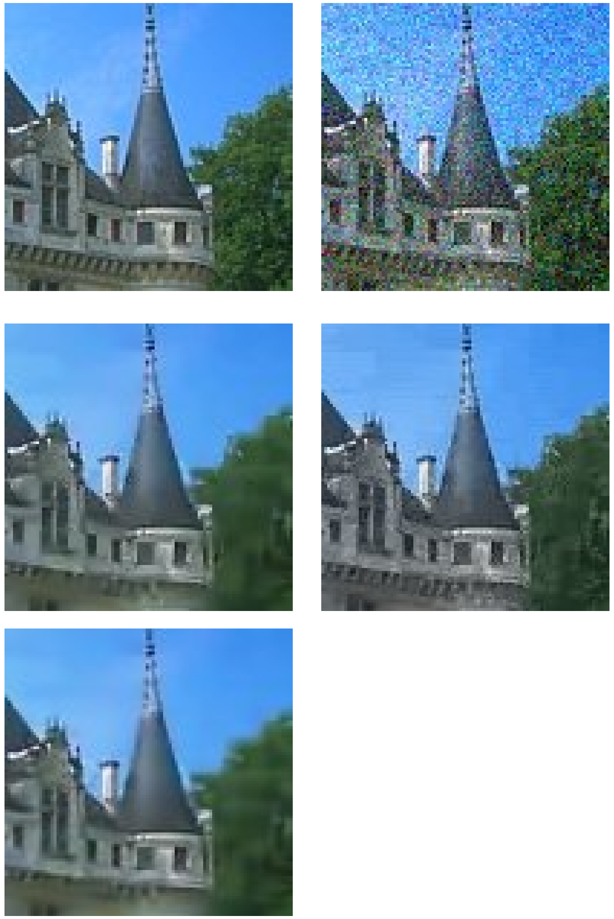


Using Roth and Black's 3x3 model (PSNR = 29.91).



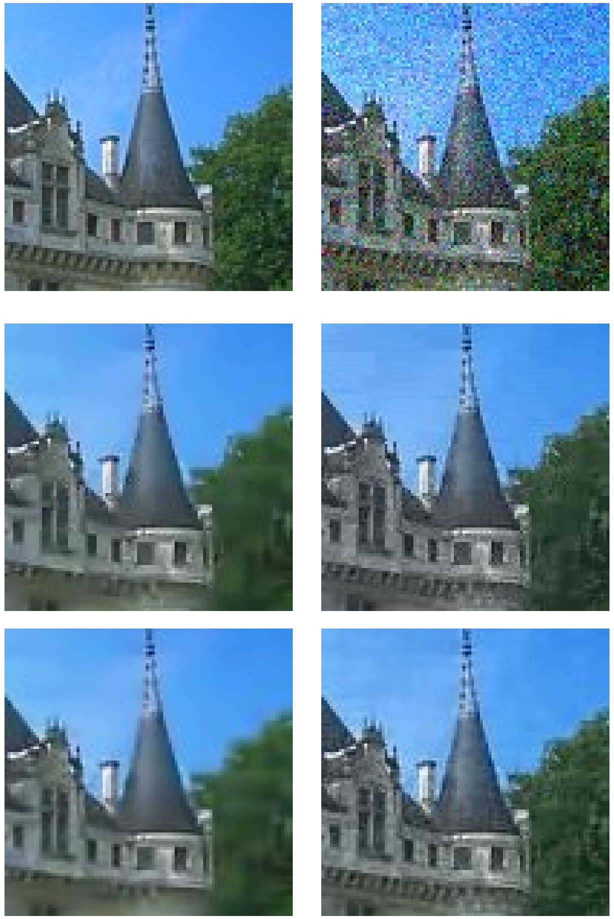
Using our 3x3 model (PSNR = 29.98).

Learning High-Order MRF Priors of Color Images



Using Roth and Black's 5x5 model (PSNR = 29.82).

Learning High-Order MRF Priors of Color Images



Using our 5x5 model (PSNR = 30.41).

Conclusions...

Conclusions...

- Even though our 5x5 model produces the best results, the time taken to perform inference is very high.

Conclusions...

- Even though our 5x5 model produces the best results, the time taken to perform inference is very high.

- Fortunately, even the 3x3 color model produces results superior to the 5x5 monochromatic model. This is an interesting result, since the inference time is approximately the same in both cases (27 dimensional filters, as opposed to 25 dimensional filters).

Conclusions...

- Even though our 5x5 model produces the best results, the time taken to perform inference is very high.

- Fortunately, even the 3x3 color model produces results superior to the 5x5 monochromatic model. This is an interesting result, since the inference time is approximately the same in both cases (27 dimensional filters, as opposed to 25 dimensional filters).

- This tells us that we are gaining *more* by moving to color than we gain by increasing the neighborhood size.

Questions?