Image Restoration

14.05.2014

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

Image Restoration

TECHNISCHE UNIVERSITÄT DARMSTADT

- Now we will shift gears and look at another problem domain: **Image restoration**
  - We will see that we can use a number of techniques that are very similar to what we have used in stereo.
- We will look at 3 applications:
  - Image denoising.
  - Image inpainting.
  - Super-resolution.

Image Denoising

TECHNISCHE UNIVERSITÄT DARMSTADT

- Essentially all digital images exhibit **image noise** to some degree.
  - Even very high quality images contain some noise.
  - Really noisy images may look like they are dominated by noise.
- Image noise is both:
  - Visually disturbing for the human viewer.
  - Distracting for vision algorithms.
- Image denoising aims at removing or reducing the amount of noise in the image.

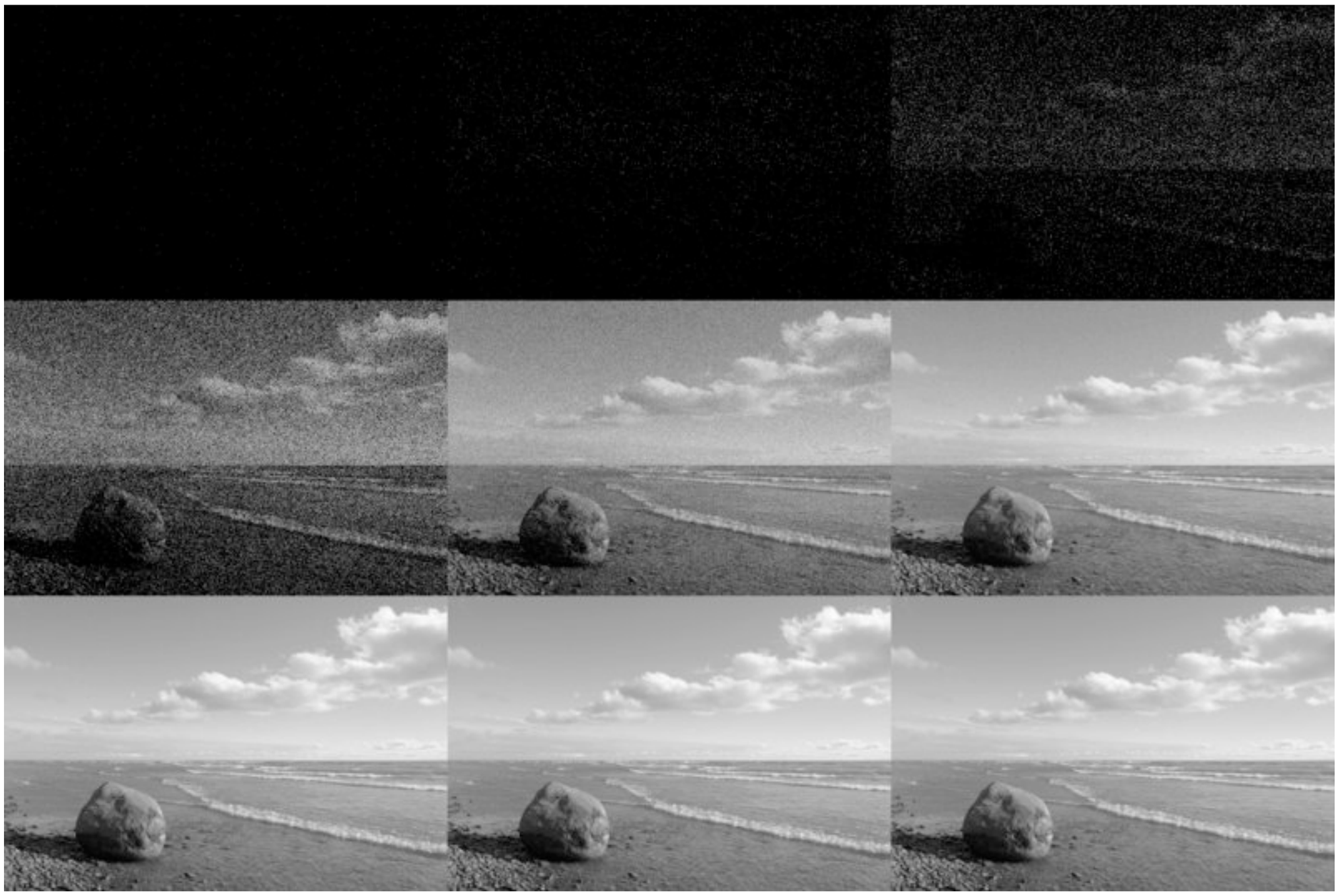
Image Noise

TECHNISCHE UNIVERSITÄT DARMSTADT

- Image noise is a very common phenomenon that can come from a variety of sources:
  - Shot noise or photon noise due to the stochastic nature of the photons arriving at the sensor.
    - Cameras actually count photons!
  - Thermal noise (“fake” photon detections).
  - Processing noise within CMOS or CCD, or in camera electronics.
  - If we use “analog” film, there are physical particles of a finite size that cause noise in a digital scan.

Shot Noise

TECHNISCHE UNIVERSITÄT DARMSTADT



Simulated shot noise (Poisson process)

From Wikipedia

Thermal Noise or “Dark Noise”

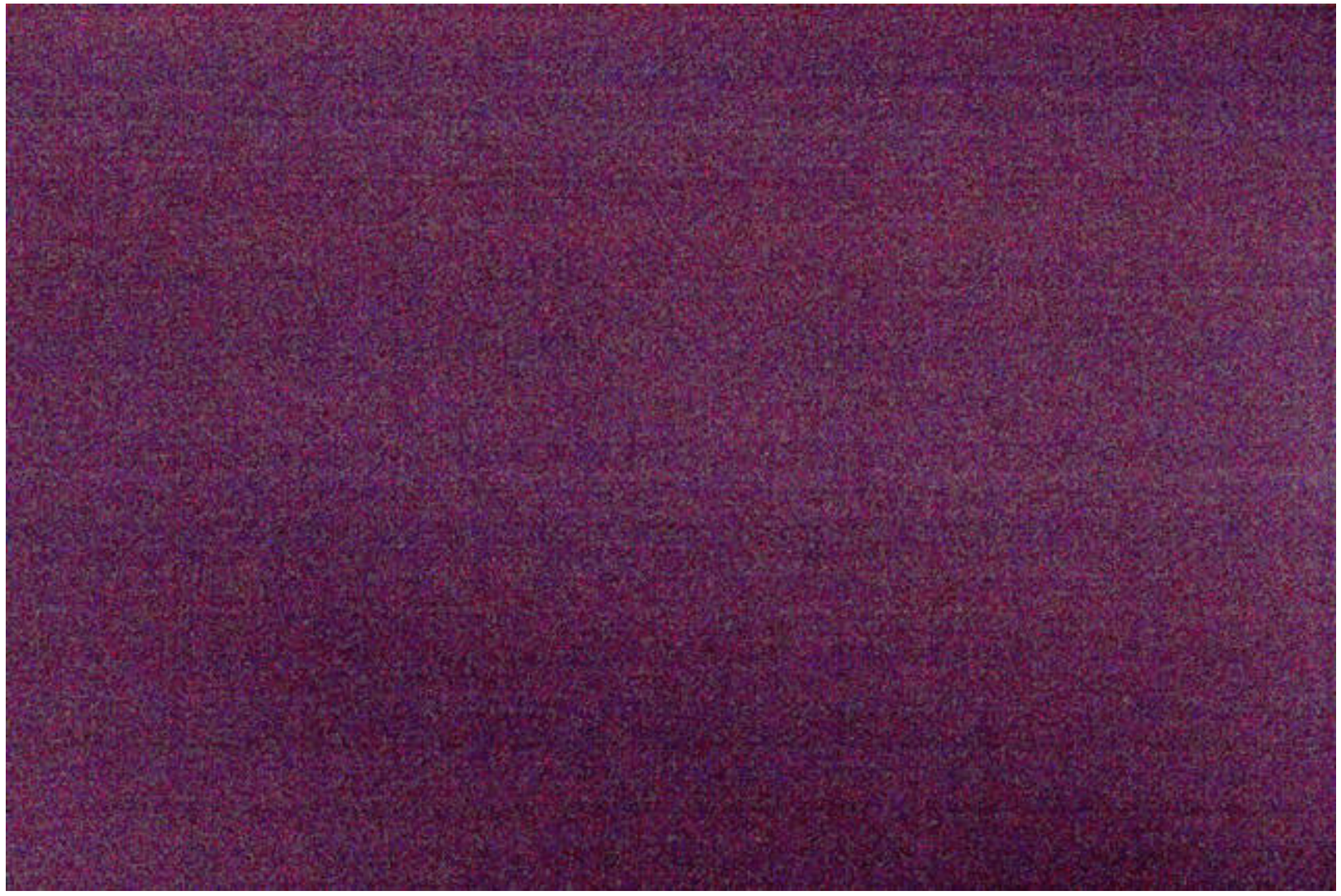
TECHNISCHE UNIVERSITÄT DARMSTADT

62 minute exposure with no incident light

From Jeff Medkeff (photo.net)

Bias Noise from Amplifiers

TECHNISCHE UNIVERSITÄT DARMSTADT



High ISO exposure (3200)

From Jeff Medkeff (photo.net)

Noise in Real Digital Photographs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Combination of many different noise sources:

From dcresource.com

Film Grain

TECHNISCHE UNIVERSITÄT DARMSTADT

- If we make a digital scan of a film, we get noise from the small silver particles on the film strip:

Das Testament des Dr. Mabuse, 1933

How do we remove image noise?

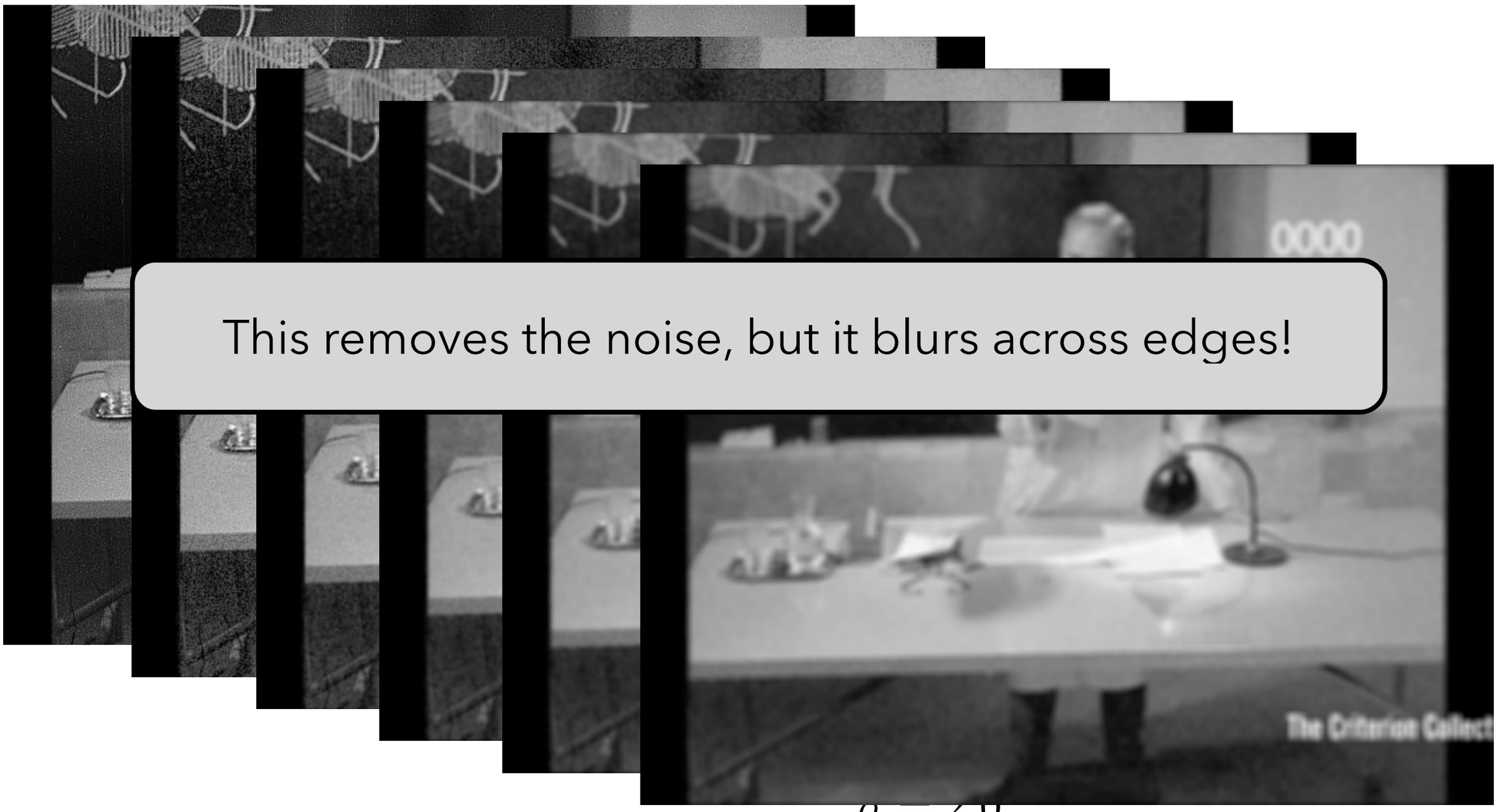
TECHNISCHE UNIVERSITÄT DARMSTADT

- Classical techniques:
  - Linear filtering, e.g. Gauss filtering.
  - Median filtering
  - Wiener filtering
  - Etc.
- Modern techniques:
  - PDE-based techniques
  - Wavelet techniques
  - **MRF-based techniques**
  - Etc.

Linear Filtering

TECHNISCHE UNIVERSITÄT DARMSTADT

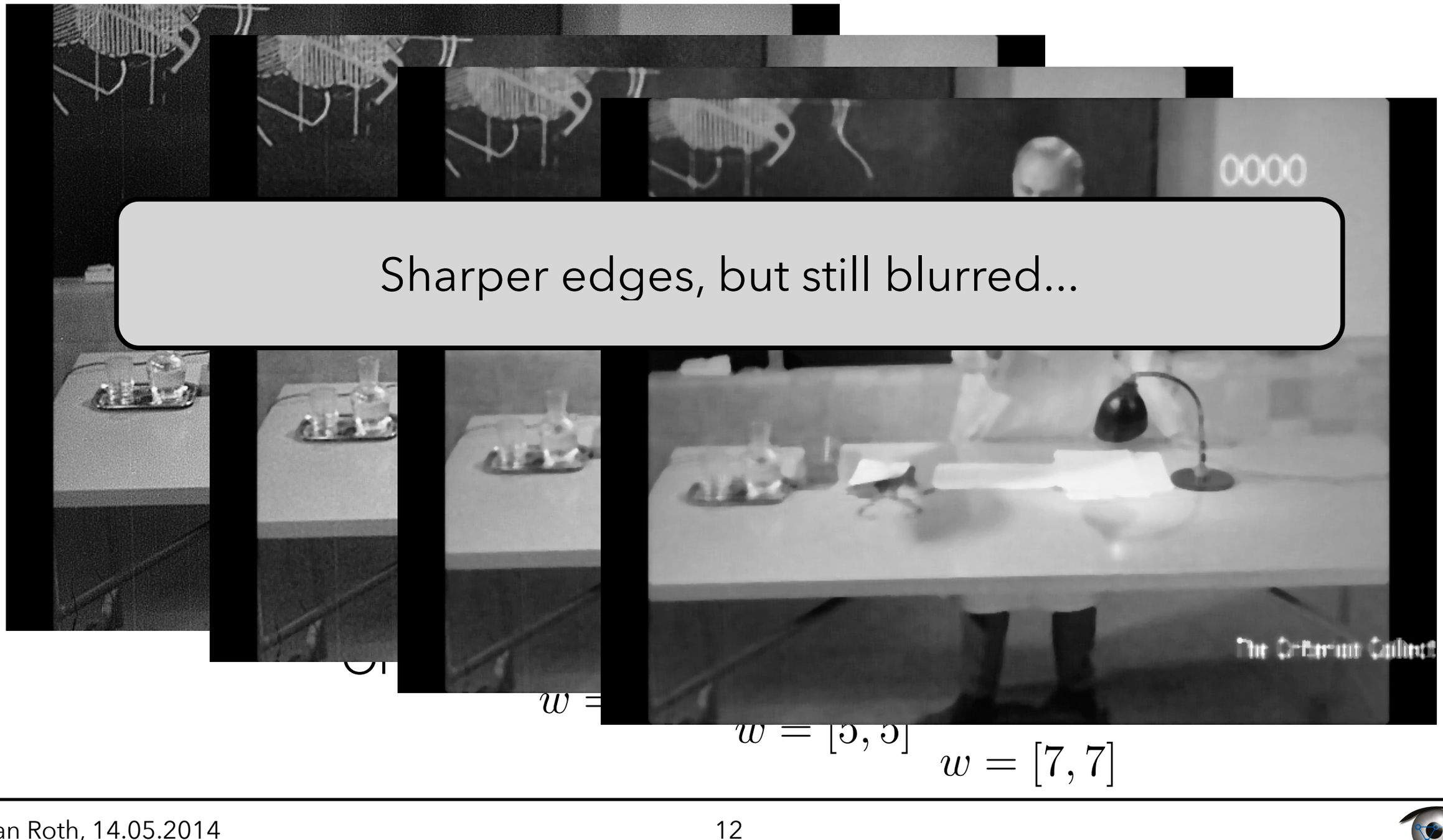
- The simplest idea is to use a Gaussian filter:



This removes the noise, but it blurs across edges!

Median Filter

- Replace each pixel by the median of the pixel values in a window around it:



How do we improve on this?

TECHNISCHE UNIVERSITÄT DARMSTADT

- We need denoising techniques that **better preserve boundaries** in images and do not blur across them.
- There is a whole host of modern denoising techniques:
  - We would need a whole semester to go through the important ones in detail.
  - So we will do the area of denoising some major injustice and restrict ourselves to what we can easily understand with what we have learned so far.

Denoising as Probabilistic Inference

TECHNISCHE UNIVERSITÄT DARMSTADT

- We formulate the problem of image denoising in a Bayesian fashion as a problem of probabilistic inference.
- For that we model denoising using a suitable posterior distribution:

$$p(\text{true image}|\text{noisy image}) = p(\mathbf{T}|\mathbf{N})$$

- If we model this posterior appropriately, we can use the inference techniques from last class to estimate the true image that we want to recover.

Modeling the Posterior

TECHNISCHE UNIVERSITÄT DARMSTADT

- We could model the posterior directly:
  - Again, this is a little more complicated.
  - Secondly, we would need to change the whole model if we want to make it work for a different type or even strength of noise.
- So instead, we apply Bayes' rule again and obtain:

likelihood of noisy given true image (observation model)

image prior for all true images

$$p(\mathbf{T}|\mathbf{N}) = \frac{p(\mathbf{N}|\mathbf{T}) \cdot p(\mathbf{T})}{p(\mathbf{N})}$$

posterior

normalization term (constant)

Modeling the Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Like in stereo, the likelihood expresses a model of the observation:
  - Given the true, noise free image, we assess how likely it is to observe a particular noisy image.
  - If we wanted to model particular real noise phenomena, we could model the likelihood based on real physical properties of the world.
  - Here, we will simplify things and only use a simple, generic noise model.
  - Nevertheless, our formulation allows us to easily adapt the noise model without having to change everything...

Modeling the Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Simplification: We assume that the noise at one pixel is **independent** of the others.

$$p(\mathbf{N}|\mathbf{T}) = \prod_{i,j} p(N_{ij}|T_{ij})$$

  - This is a pretty reasonable assumption, for example since sites of a CCD sensor are relatively independent.

- Then we will assume that the noise at each pixel is **additive and Gaussian distributed**:

$$p(N_{ij}|T_{ij}) = \mathcal{N}(N_{ij} - T_{ij}|0, \sigma^2)$$

  - The variance σ^2 controls the amount of noise.

Gaussian Image Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- We can thus write the Gaussian image likelihood as:

$$p(\mathbf{N}|\mathbf{T}) = \prod_{i,j} \mathcal{N}(N_{ij} - T_{ij}|0, \sigma^2)$$

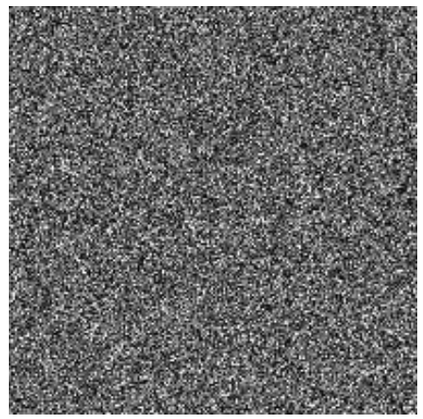
- While this may seem a bit hacky, it works well in many applications.
  - It is suboptimal when the noise is not really independent, such as in some high definition (HD) images.
  - It also is suboptimal when the noise is non-additive, or not really Gaussian, for example as with film grain noise.

Modeling the Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- How do we model the prior distribution of true images?
- What does that even mean?
  - We want the prior to describe how probable it is (a-priori) to have a particular true image among the set of all possible images.

probable

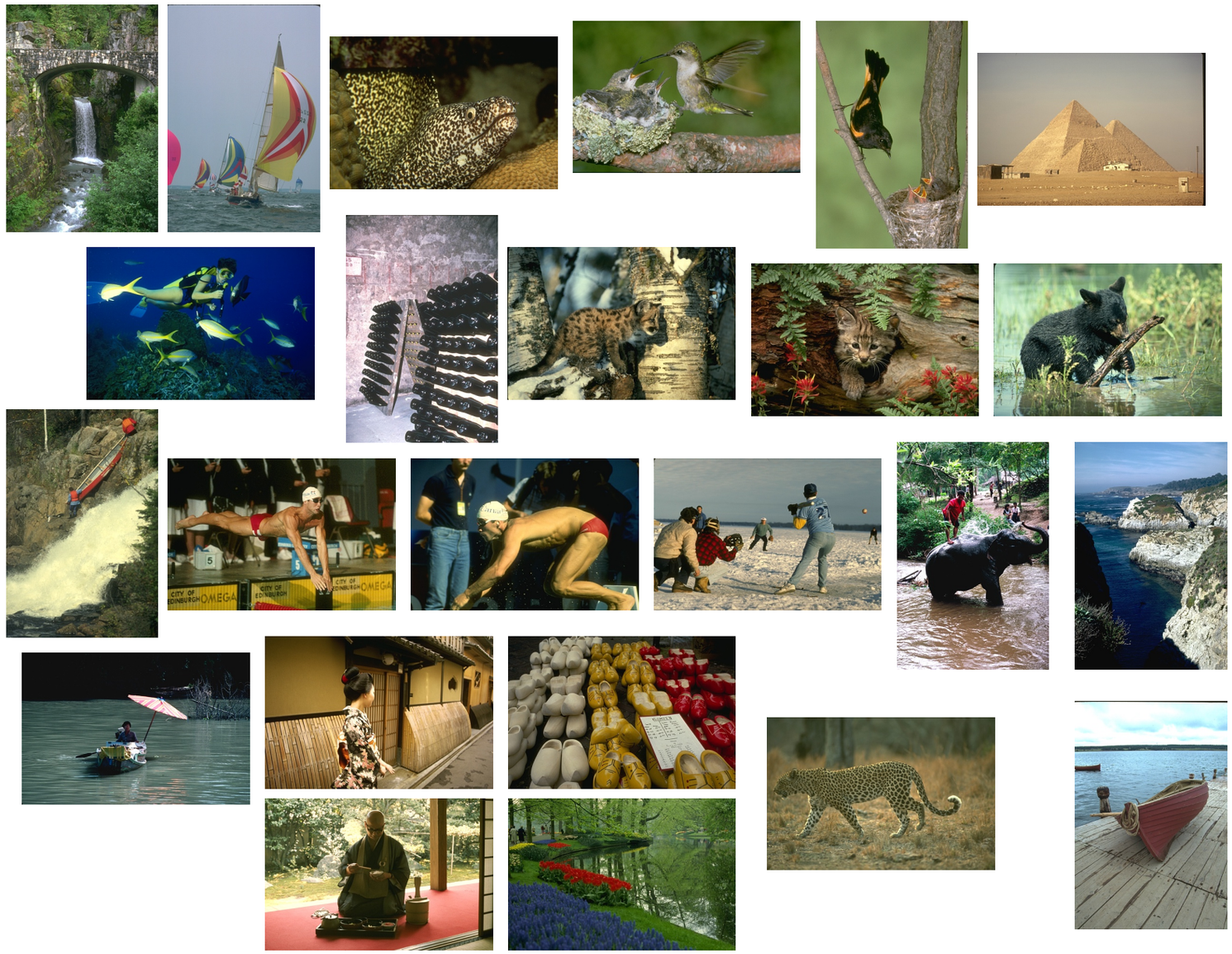


improbable

Natural Images

TECHNISCHE UNIVERSITÄT DARMSTADT

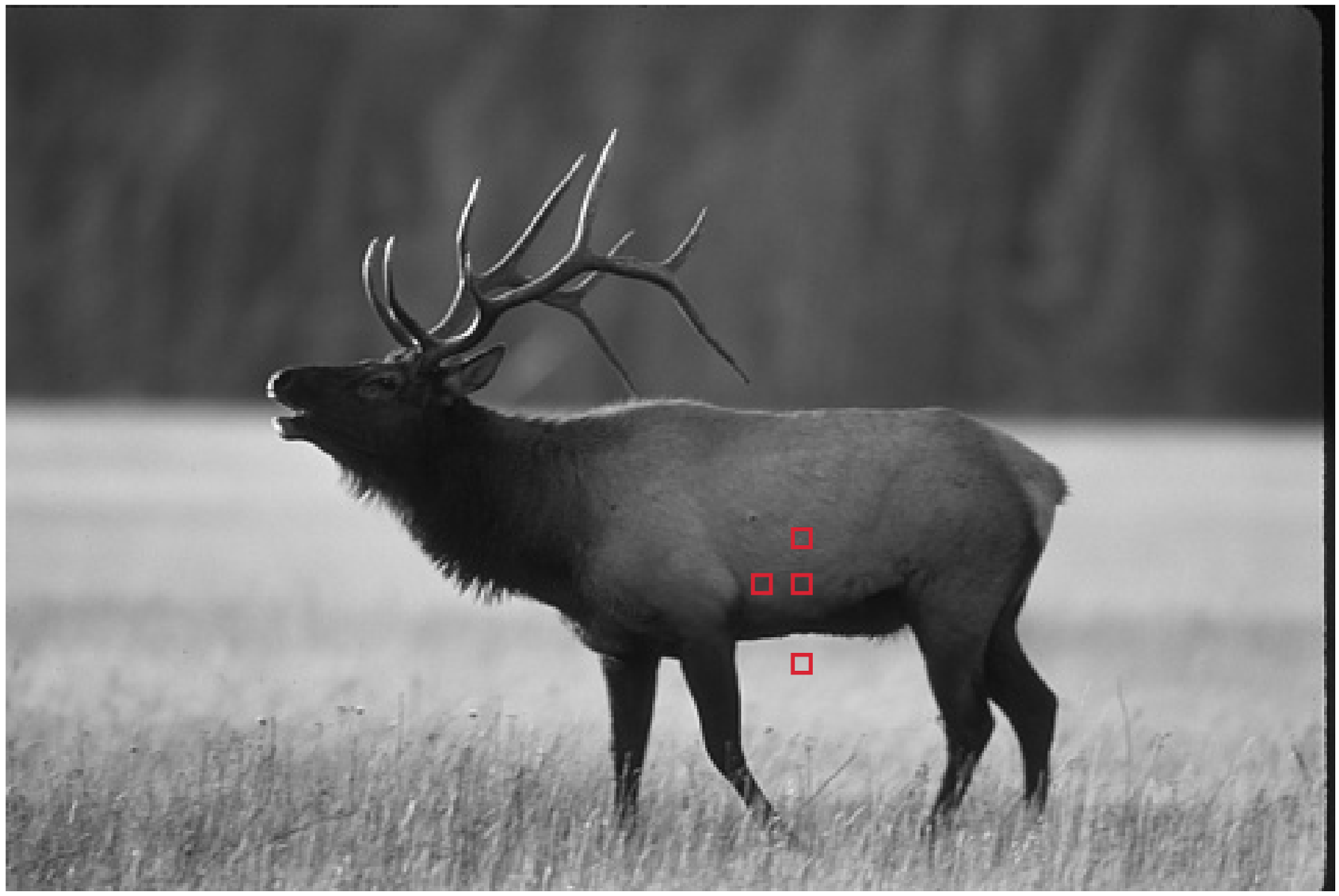
- What distinguishes “natural” images from “fake” ones?
  - We can take a large database of natural images and study them.



Simple Observation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nearby pixels often have a similar intensity:

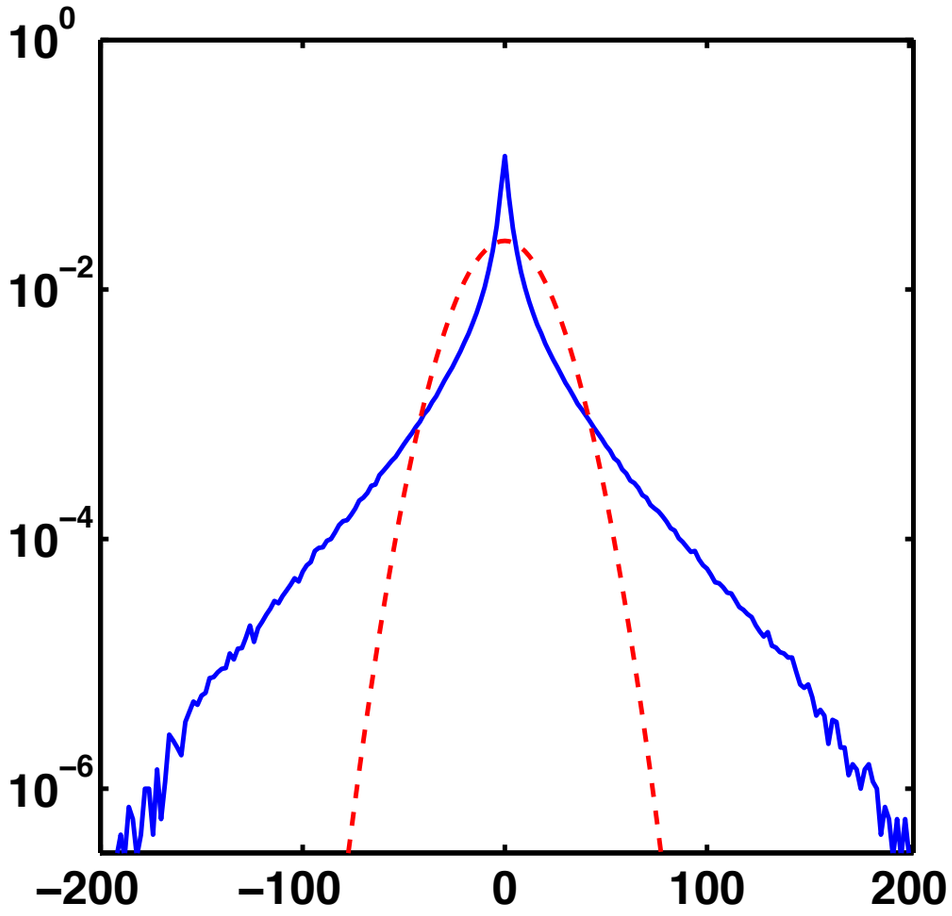


- But sometimes there are large intensity changes.

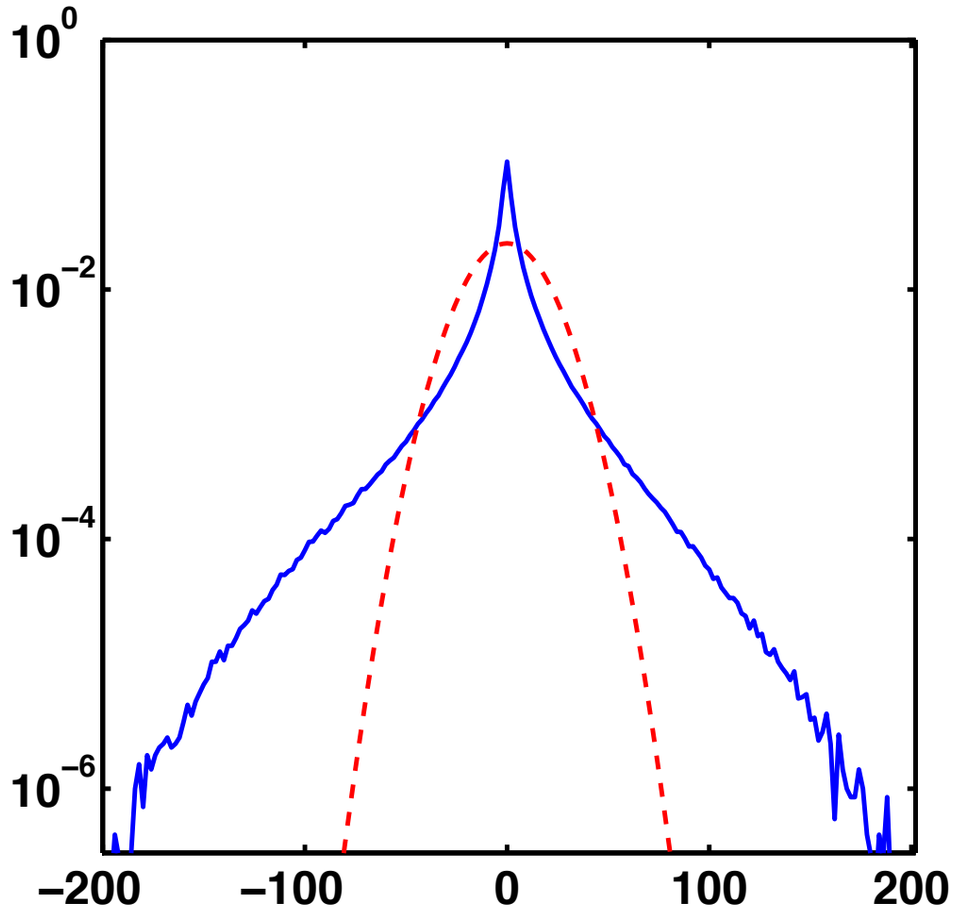
Statistics of Natural Images

TECHNISCHE UNIVERSITÄT DARMSTADT

- Compute the image derivative of all images in an image database and plot a histogram:



x-derivative



y-derivative

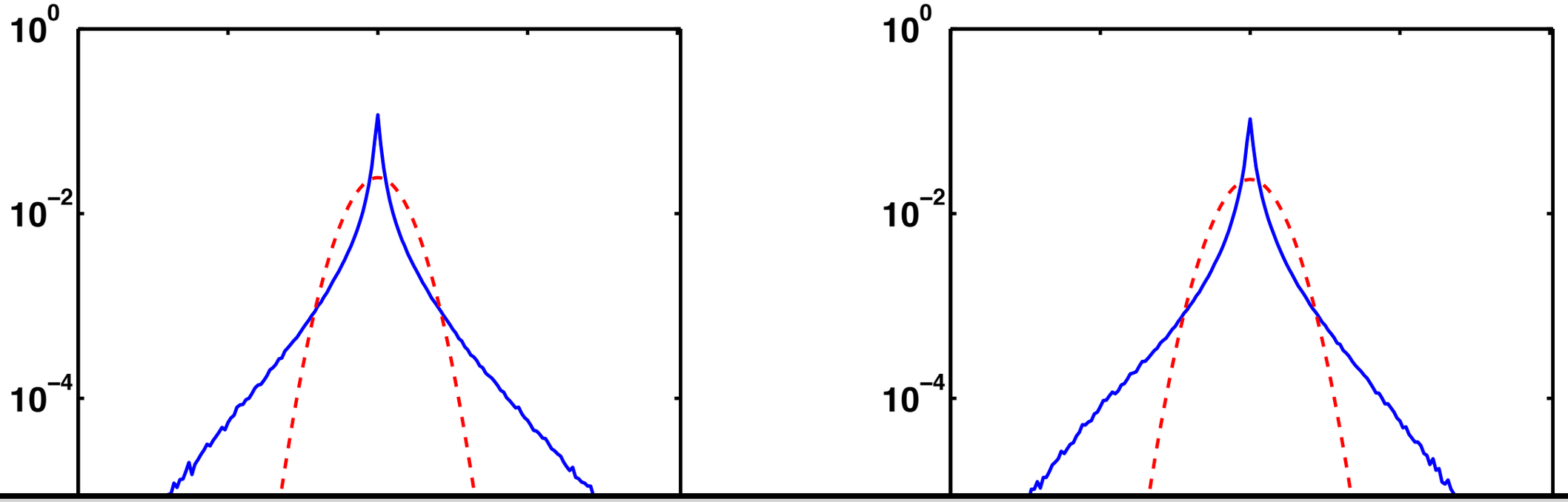
empirical histogram

fit with a Gaussian

Statistics of Natural Images

TECHNISCHE UNIVERSITÄT DARMSTADT

- Compute the image derivative of all images in an image database and plot a histogram:



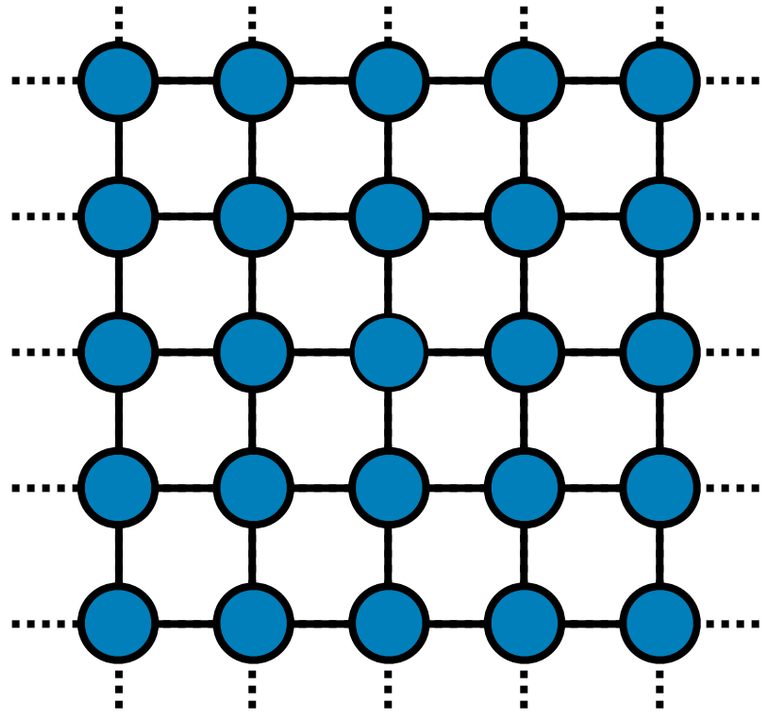
- Sharp peak at zero: Neighboring pixels most often have identical intensities.
- Heavy tails: Sometimes, there are strong intensity differences due to discontinuities in the image.

MRF Model of Image Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- We have seen something like this before!
- In stereo, we had almost the same challenge in modeling disparity.
  - So why don't we use the same formalism?
- We model the image prior as a Markov random field:

$$p(\mathbf{T}) = \prod_{i,j} f_H(T_{i,j}, T_{i+1,j}) \cdot f_V(T_{i,j}, T_{i,j+1})$$

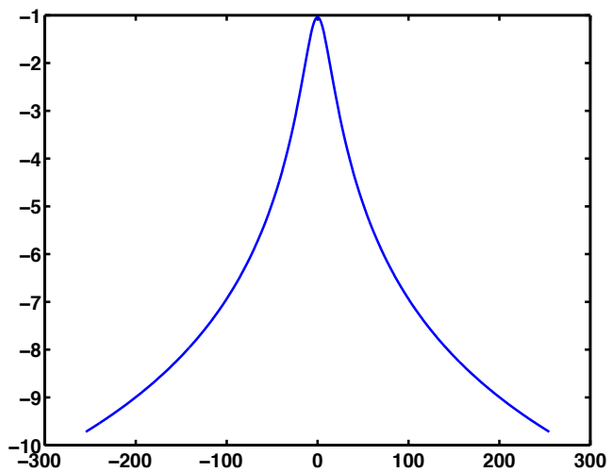


Modeling the Potentials

TECHNISCHE UNIVERSITÄT DARMSTADT

- We face a similar challenge in modeling the potentials (or compatibility functions), e.g.: $f_H(T_{i,j}, T_{i+1,j})$
- Gaussian distributions are inappropriate:
  - They do not match the statistics of natural images well.
  - They would lead to blurred discontinuities.
- We need **discontinuity-preserving potentials**:
  - One possibility: Student-t distribution.

$$f_H(T_{i,j}, T_{i+1,j}) = \left(1 + \frac{1}{2\sigma^2}(T_{i,j} - T_{i+1,j})^2\right)^{-\alpha}$$

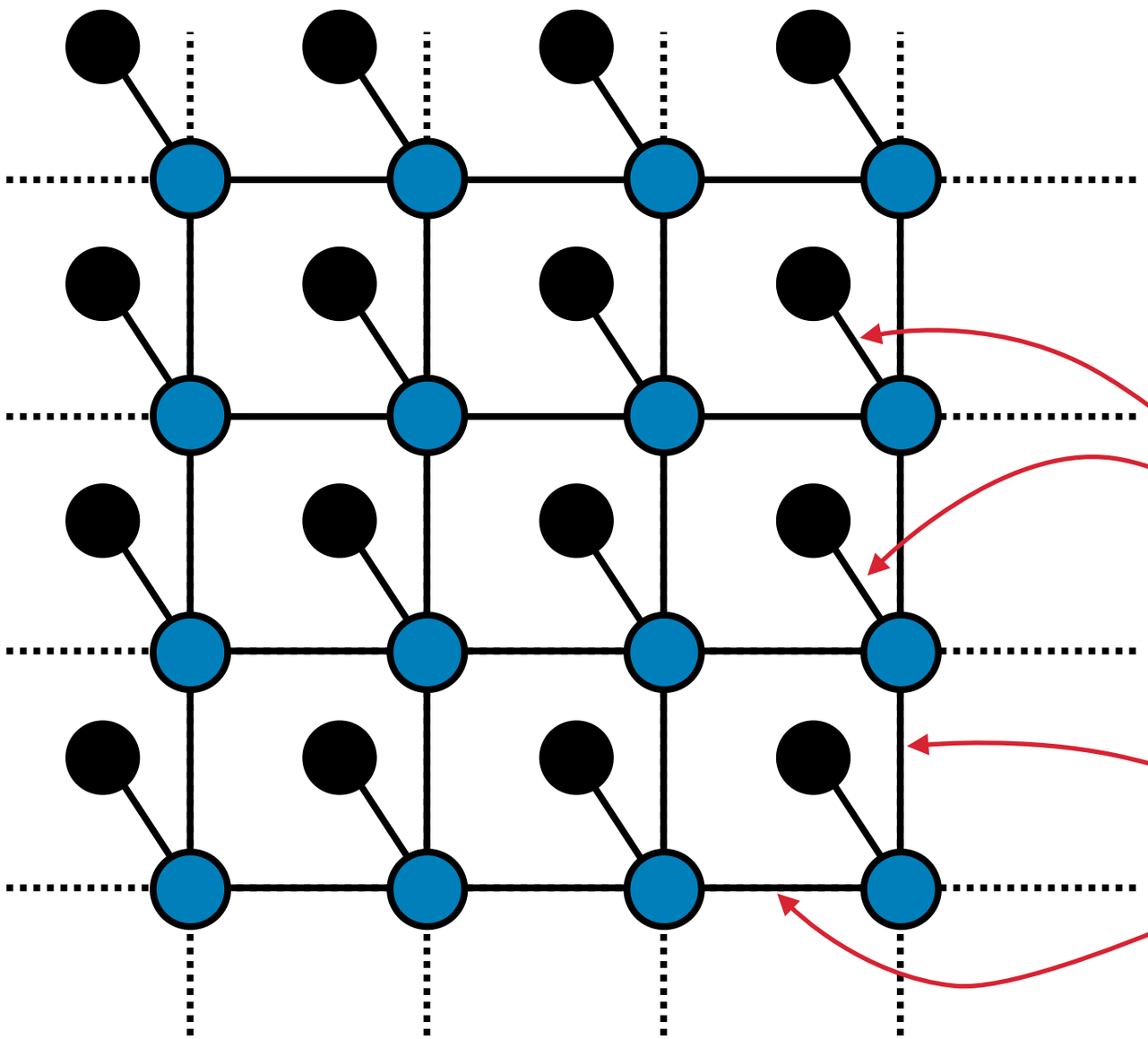


log-density

MRF Model of the Posterior

TECHNISCHE UNIVERSITÄT DARMSTADT

- We can now put the likelihood and the prior together in a single MRF model:



$T_{i,j}$ pixels of the true image (hidden)

$N_{i,j}$ pixels of the noisy image (observed)

Edges representing the likelihood

Edges representing the prior

Denoising as Inference

TECHNISCHE UNIVERSITÄT DARMSTADT

- Because the likelihood is simpler here than in the stereo case, gradient techniques work quite well for inference.
  - Very simple to implement.
- Of course, we can also use graph cuts or belief propagation.
  - Can lead to slightly better results (in case of a pairwise prior).

Denoising Results

TECHNISCHE UNIVERSITÄT DARMSTADT

original image

noisy image,
σ=20

PSNR 22.49dB
SSIM 0.528

denoised using
gradient ascent

PSNR 27.60dB
SSIM 0.810

Denoising Results

- Very sharp discontinuities. No blurring across boundaries.
- Noise is removed quite well nonetheless.

Denoising Results

TECHNISCHE UNIVERSITÄT DARMSTADT

- Because the noisy image is based on synthetic noise, we can measure the performance:
  - PSNR: Peak signal-to-noise ratio
  - SSIM [Wang et al., 04]: Perceptual similarity
    - Gives an estimate of how humans would assess the quality of the image.

original image

noisy image, σ=20

PSNR 22.49dB
SSIM 0.528

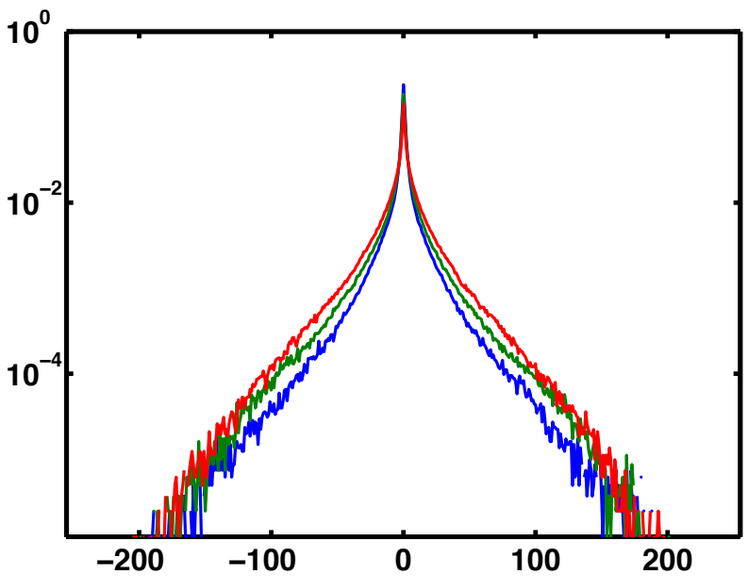
denoised using gradient ascent

PSNR 27.60dB
SSIM 0.810

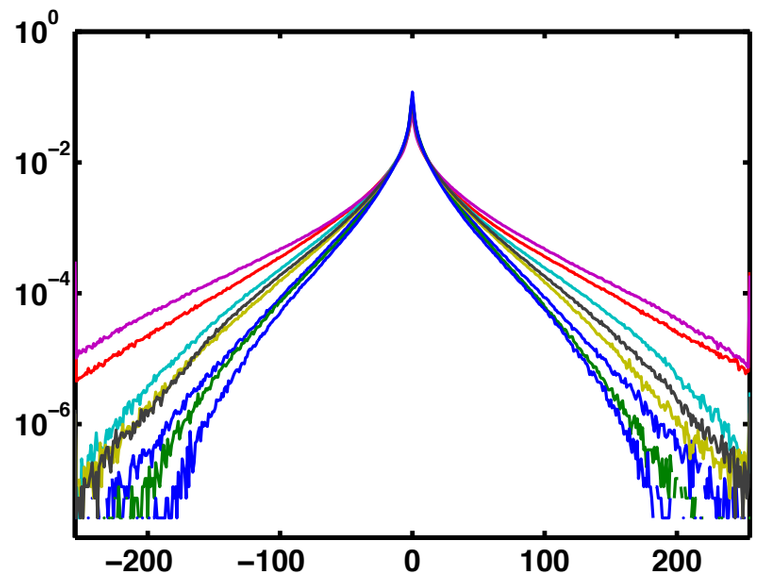
Is this the end of the story?

TECHNISCHE UNIVERSITÄT DARMSTADT

- No, natural images have many complex properties that we have not modeled.
  - For example, they have complex structural properties that are not modeled with a simple MRF based on a 4-connected grid.
  - Natural images have scale-invariant statistics, our model does not.
  - Responses to random linear filters are heavy-tailed.
  - Etc.



Derivative histogram on 4 spatial scales

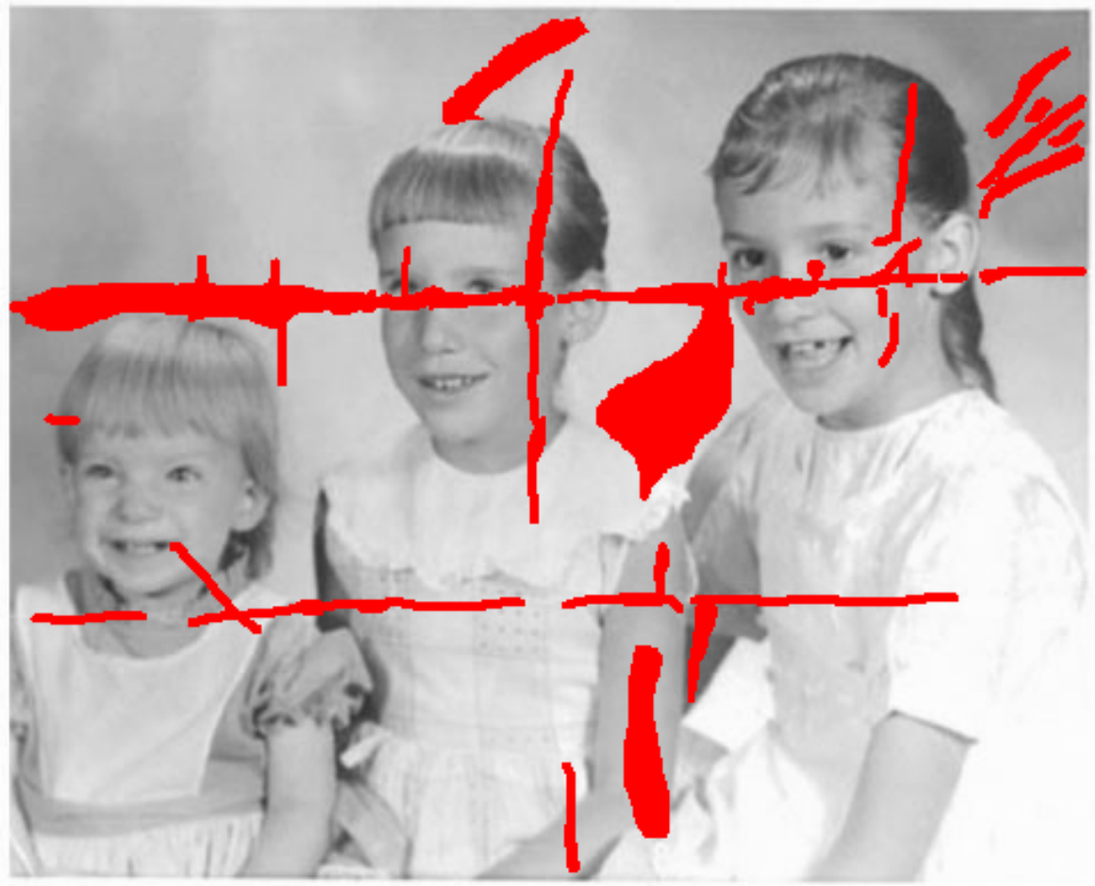


Histograms of random (zero-mean) linear filters

Image Inpainting

- In image inpainting the goal in to fill in a "missing" part of an image:
  - Restoration of old photographs, e.g. scratch removal...

old photograph



user-supplied mask

[Bertalmio et al., 2000]

Image Inpainting

TECHNISCHE UNIVERSITÄT DARMSTADT

- There are many different ways to do inpainting:
  - PDEs: [Bertalmio et al, 2000], ...
  - Exemplar-based: [Criminisi et al., 2003], ...
  - And many more.
- But, we can also apply what we already know:
  - We model the problem in a Bayesian fashion, where we regard the inpainted image as the true image. We are given the corrupted image with missing pixels.

$$p(\text{true image}|\text{corrupted image}) = p(\mathbf{T}|\mathbf{C})$$

  - Then we apply probabilistic inference...

Image Inpainting

- We apply Bayes' rule:

$$p(\mathbf{T}|\mathbf{C}) = \frac{p(\mathbf{C}|\mathbf{T}) \cdot p(\mathbf{T})}{p(\mathbf{C})}$$

  - I know this may be boring, but this general approach really is this versatile...
- Modeling the prior:
  - **Important observation:** This is the very same prior that we needed in order to do denoising!
  - We can re-use the prior model from denoising here.
  - Once we have a good probabilistic model of images, we can use it in many applications!

Inpainting Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Again, we assume independence of the pixels:

$$p(\mathbf{C}|\mathbf{T}) = \prod_{i,j} p(C_{ij}|T_{ij})$$

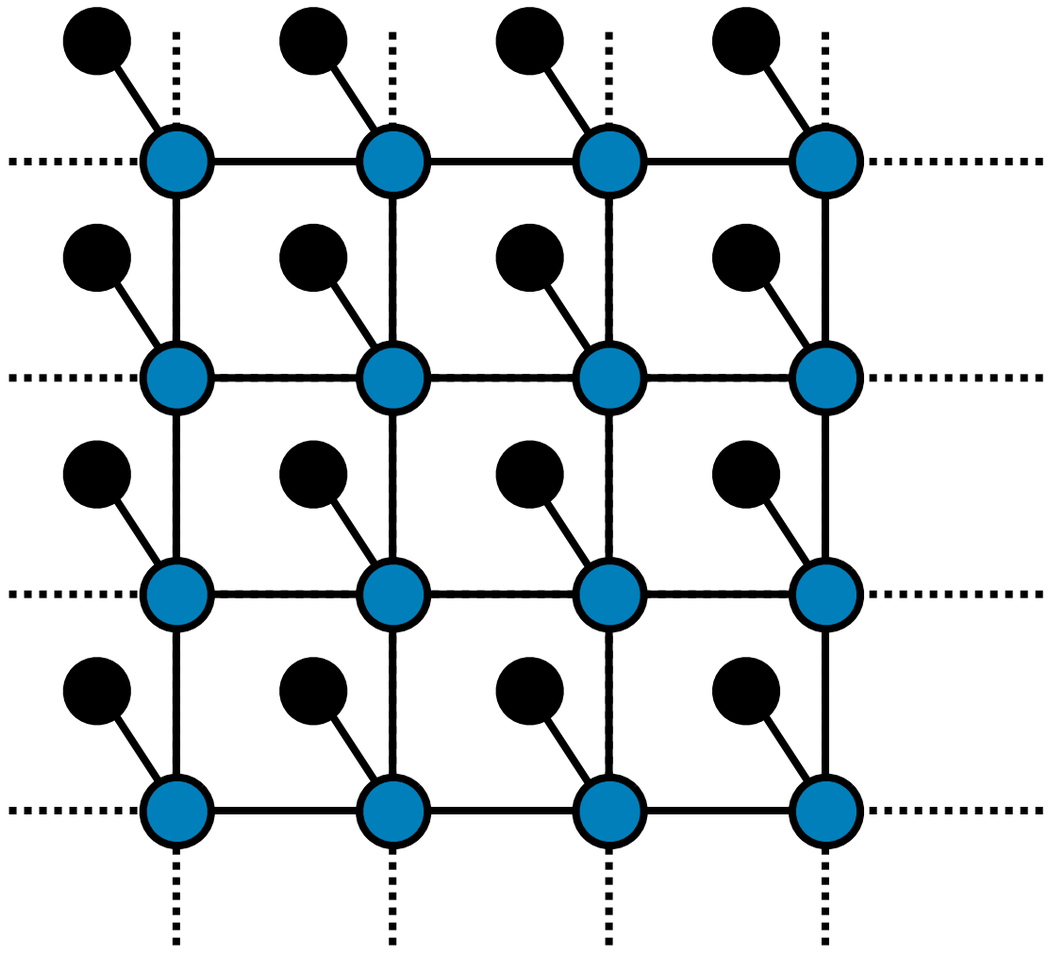
- Desiderata:
  - We want to keep all known pixels fixed.
  - For all unknown pixels all intensities should be equally likely.
- Simple likelihood model:

$$p(C_{ij}|T_{ij}) = \begin{cases} \text{const}, & C_{ij} \text{ is corrupted} \\ \delta(T_{ij} - C_{ij}), & C_{ij} \text{ is not corrupted} \end{cases}$$

MRF Model of the Posterior

TECHNISCHE UNIVERSITÄT DARMSTADT

- The posterior for inpainting has the same graphical model structure as the one for denoising.
- Nonetheless, the potentials representing the likelihood are different.



Inpainting Results

"Corrupted" image (artificially corrupted for benchmarking)

Inpainted image obtained using gradient ascent

Other Inpainting Results

TECHNISCHE UNIVERSITÄT DARMSTADT

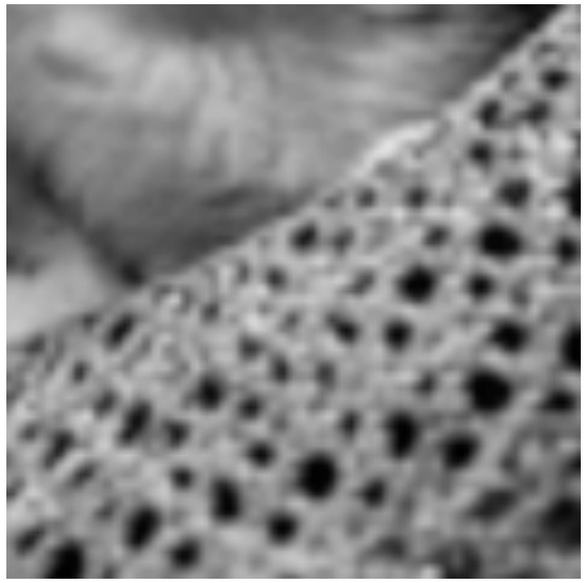
From [Bertalmio et al., 2000]

Image Super-Resolution

TECHNISCHE UNIVERSITÄT DARMSTADT

- In super-resolution, the goal is to increase the resolution of an image, while making the high-resolution image seem sharp and natural.

64x64 original image



128x128 super-resolved image (bicubic interpolation)

From [Tappen et al., 2003]

Image Super-Resolution

TECHNISCHE UNIVERSITÄT DARMSTADT

- In super-resolution, the goal is to increase the resolution of an image, while making the high-resolution image seem sharp and natural.

- Many applications:
  - Resizing of old, low-res digital imagery.
  - Improving resolution of images from cheap cameras.
  - Up-conversion of standard definition TV and movie footage to high-definition (720p or 1080p).

Image Super-Resolution

TECHNISCHE UNIVERSITÄT DARMSTADT

- The story parallels what we have seen before.
- There are many special purpose super-resolution techniques:
  - The simplest ones are bilinear and bicubic interpolation.
  - Example-based methods.
  - Etc.
- But once again, we can formulate the problem in a Bayesian way and super-resolve images using probabilistic inference:

$$p(\text{high res. image}|\text{low res. image}) = p(\mathbf{H}|\mathbf{L})$$

Image Super-Resolution

TECHNISCHE UNIVERSITÄT DARMSTADT

- Yet again apply Bayes rule:

$$p(\mathbf{H}|\mathbf{L}) = \frac{p(\mathbf{L}|\mathbf{H}) \cdot p(\mathbf{H})}{p(\mathbf{L})}$$

  - I know, this may really get boring by now...

- Modeling the prior of high-resolution images:
  - We can again use the prior model that we already have!
- Modeling the likelihood:
  - The likelihood has to model how the low-resolution pixels correspond to the high-resolution pixels.
  - For simplicity, assume a fixed zooming factor of 2x.

Super-Resolution Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- If we have a zoom factor of 2x, there is a 2x2 patch of high-resolution pixels that corresponds to a single low-resolution pixel.
- Again assume conditional independence of the low-resolution pixels given the high-resolution image.
- We can thus write the likelihood as:

$$p(\mathbf{L}|\mathbf{H}) = \prod_{i,j} p(L_{i,j}|H_{2i,2j}, H_{2i+1,2j}, H_{2i,2j+1}, H_{2i+1,2j+1})$$

- We may for example assume that a low-resolution pixel is the average of the 4 high-resolution pixels plus a small amount of Gaussian noise:

$$p(L_{i,j}|H_{2i,2j}, H_{2i+1,2j}, H_{2i,2j+1}, H_{2i+1,2j+1}) = \mathcal{N}\left(L_{i,j} - \frac{1}{4}(H_{2i,2j} + H_{2i+1,2j} + H_{2i,2j+1} + H_{2i+1,2j+1}); 0, \sigma^2\right)$$

Super-resolution Results

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gradient ascent does not work very well here, because the likelihood is more complex than in denoising or inpainting.
- But belief propagation can be made to work (with some tricks...):

original

MRF + BP

bicubic interpolation

From [Tappen et al., 2003]

Super-resolution Results

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gradient ascent does not work very well here, because the likelihood is more complex than in denoising or inpainting.
- But belief propagation can be made to work (with some tricks...):

original

MRF + BP

bicubic interpolation

From [Tappen et al., 2003]

Super-resolution Results

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gradient ascent does not work very well here, because the likelihood is more complex than in denoising or inpainting.
- But belief propagation can be made to work (with some tricks…):

original

MRF + BP

bicubic interpolation

From [Tappen et al., 2003]

Summary

- Most relevant image processing problems can be formulated as problems of probabilistic inference.
  - This is only one of many ways of approaching these problems!
- Advantages:
  - Unified approach to many different problems, in which important components (prior) may be re-used.
  - Is is relatively easy to understand what the various parts do.
  - Good application performance, despite generality.
- Disadvantages:
  - Computationally often expensive.
  - Special purpose techniques often have somewhat better application performance.

High-order MRFs & Learning

14.05.2014

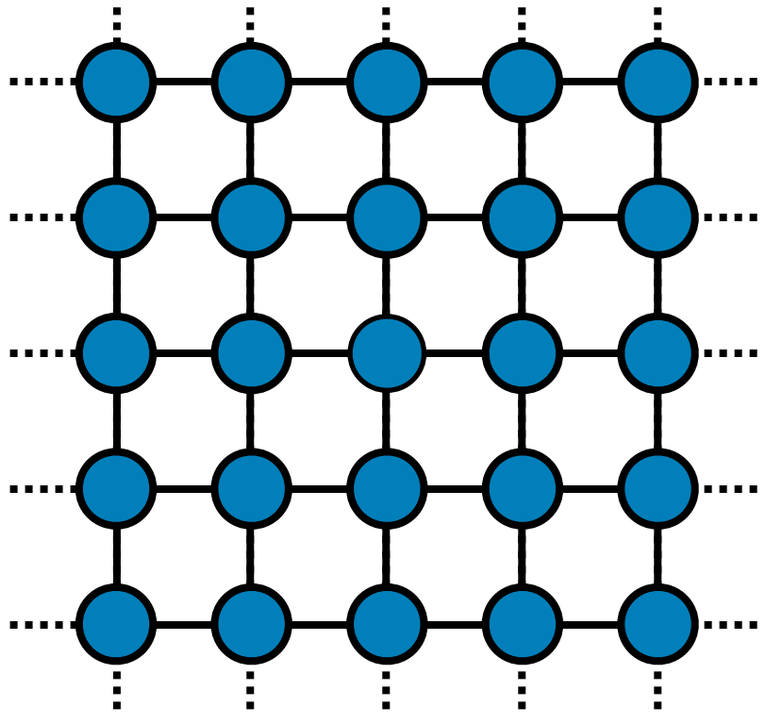
TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

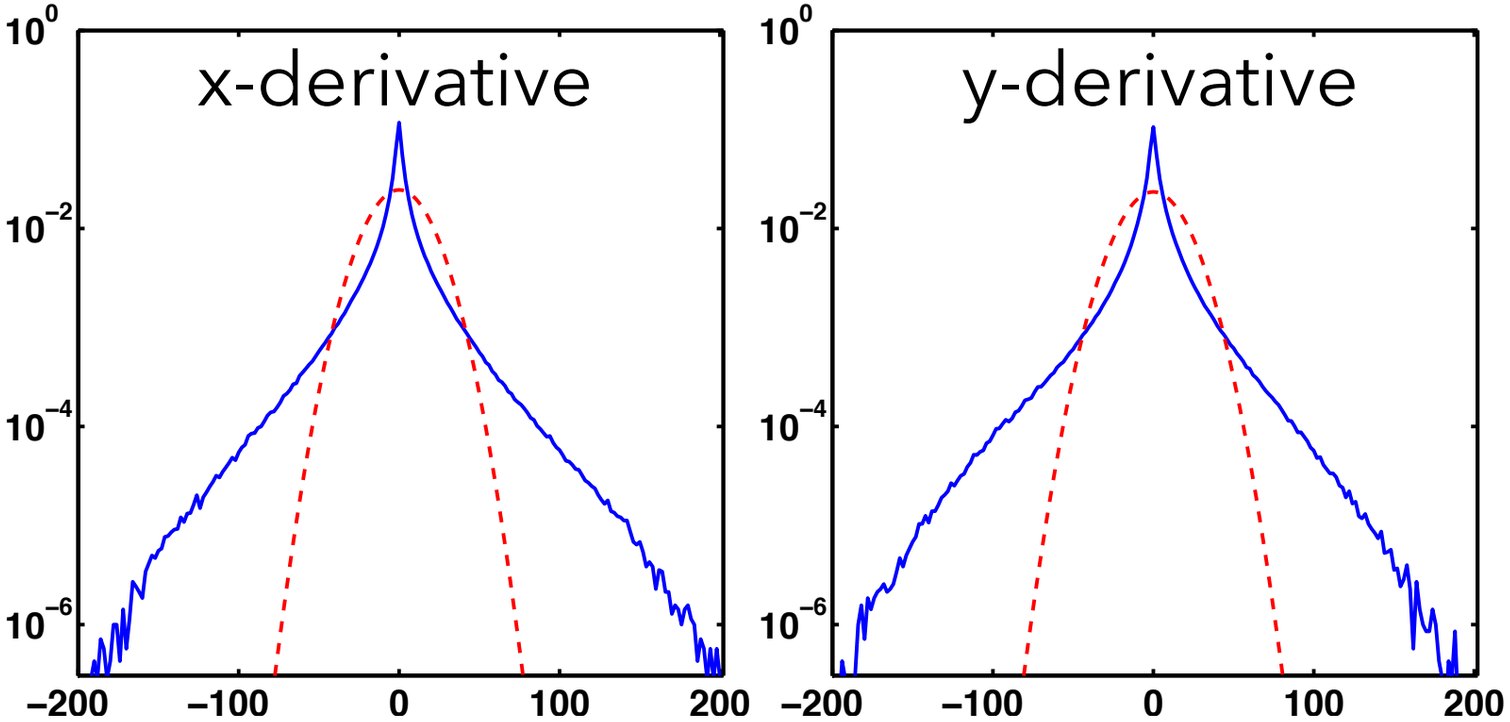
"Pairwise" MRF Image Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- 4-connected Markov random field



$$p(\mathbf{T}) = \prod_{i,j} f_H(T_{i,j}, T_{i+1,j}) \cdot f_V(T_{i,j}, T_{i,j+1})$$

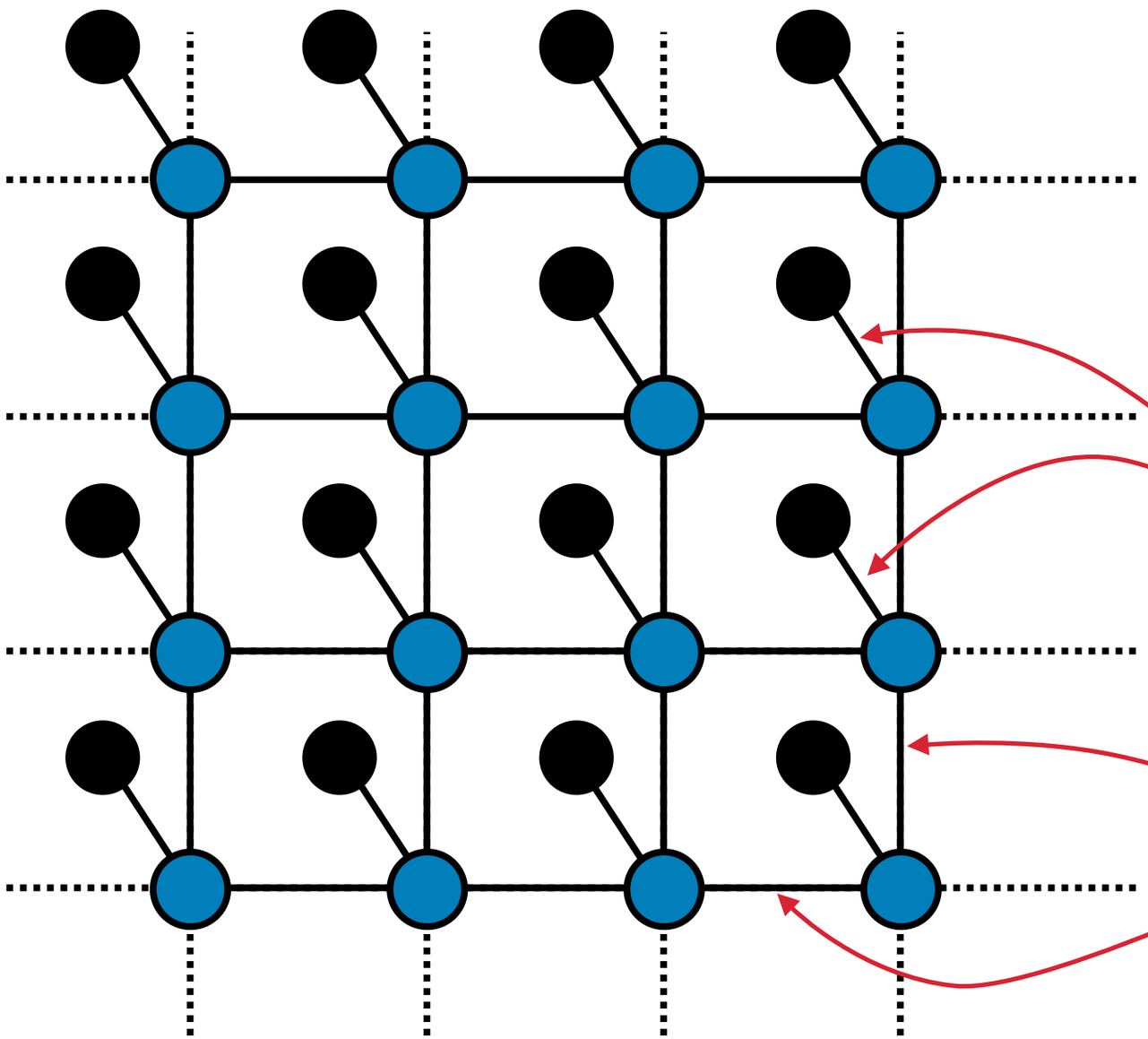


empirical histogram — fit with a Gaussian

MRF Model of the Posterior

TECHNISCHE UNIVERSITÄT DARMSTADT

- Likelihood and the prior together in a single MRF model:



$T_{i,j}$ pixels of the true image (hidden)

$N_{i,j}$ pixels of the noisy image (observed)

Edges representing the likelihood

Edges representing the prior

Denoising Results

TECHNISCHE UNIVERSITÄT DARMSTADT

original image

noisy image,
σ=20

PSNR 22.49dB
SSIM 0.528

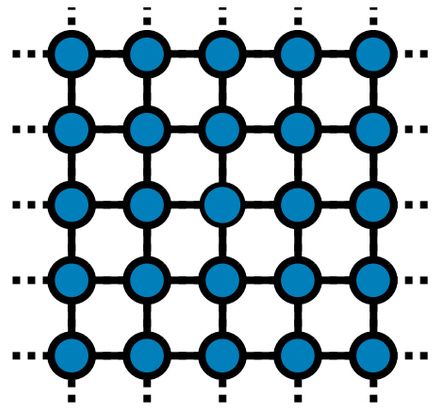
denoised using
gradient ascent

PSNR 27.60dB
SSIM 0.810

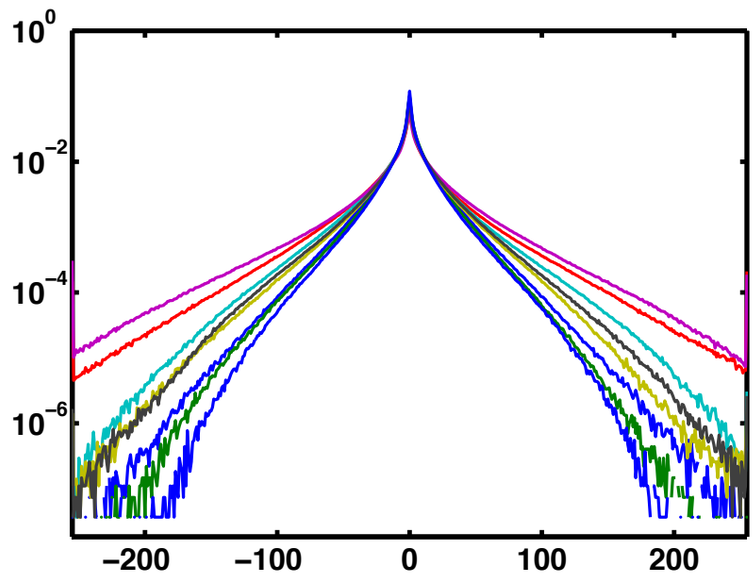
Can we do better?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Relax strong conditional independence (Markov) assumption
  - MRF based on a 4-connected grid restrictive (more complex properties cannot be modeled)

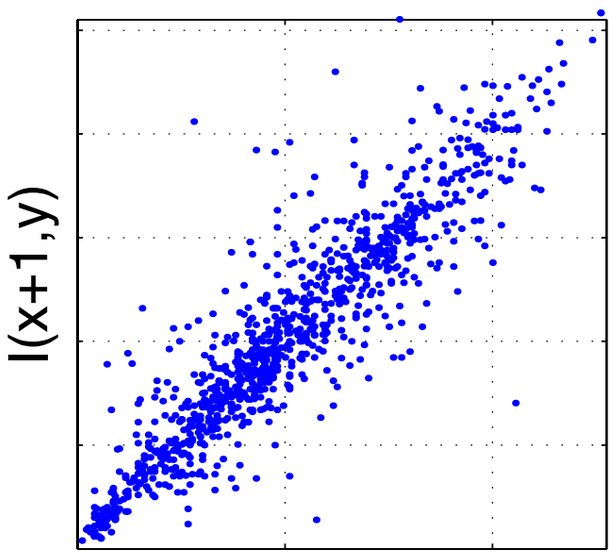


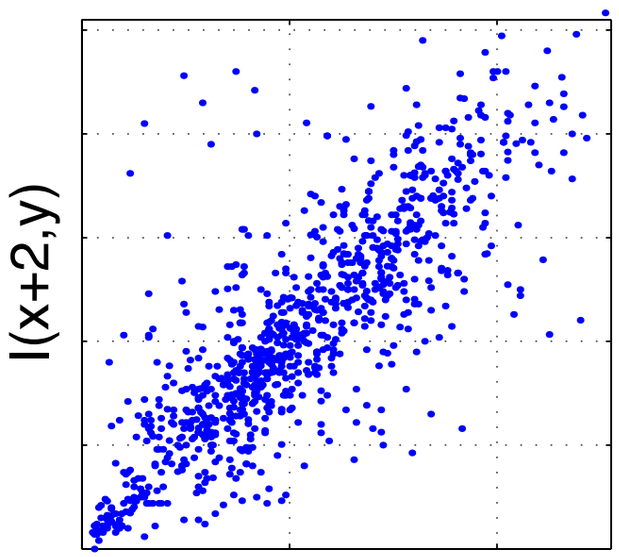
- Motivated by image statistics
  - Responses to larger (random) linear filters are also heavy-tailed

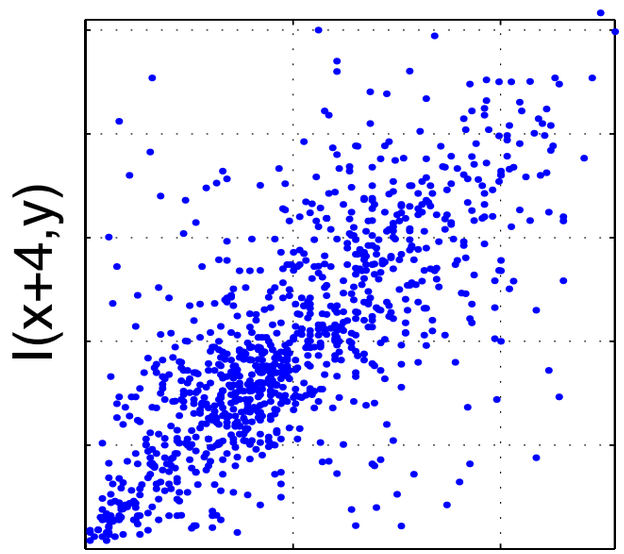


Histograms of random (zero-mean) linear filters

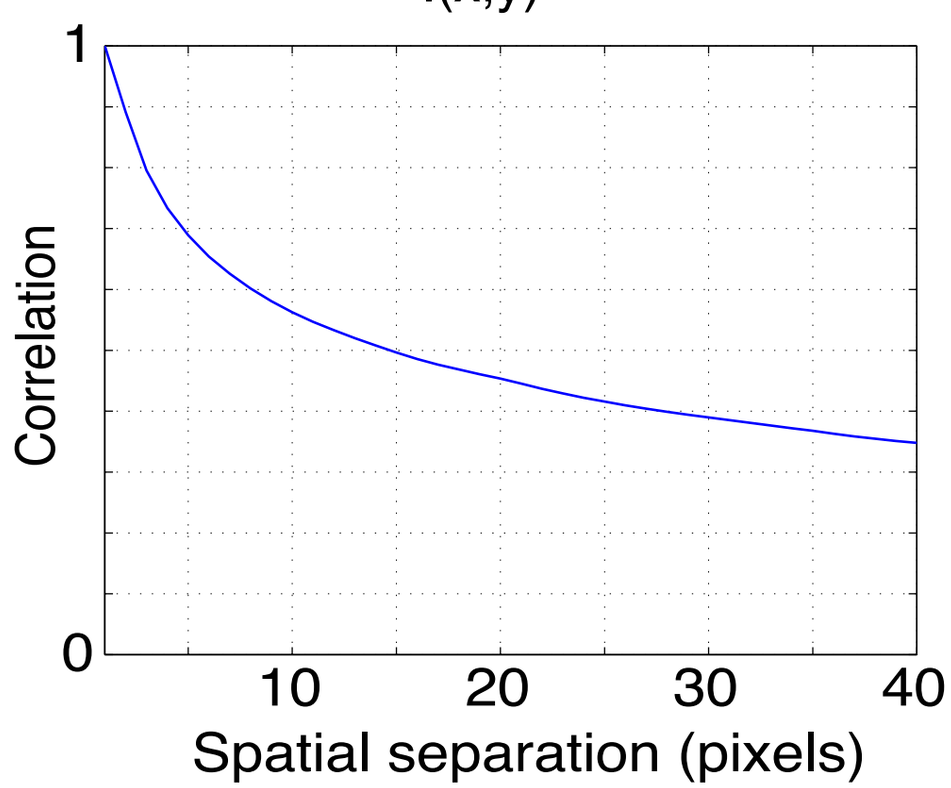
Pixel - joint







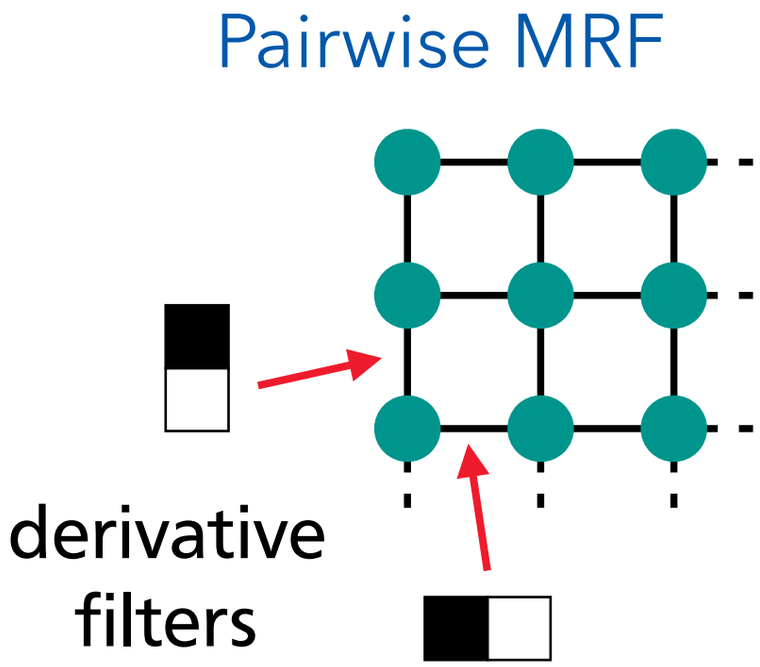
second-order correlation

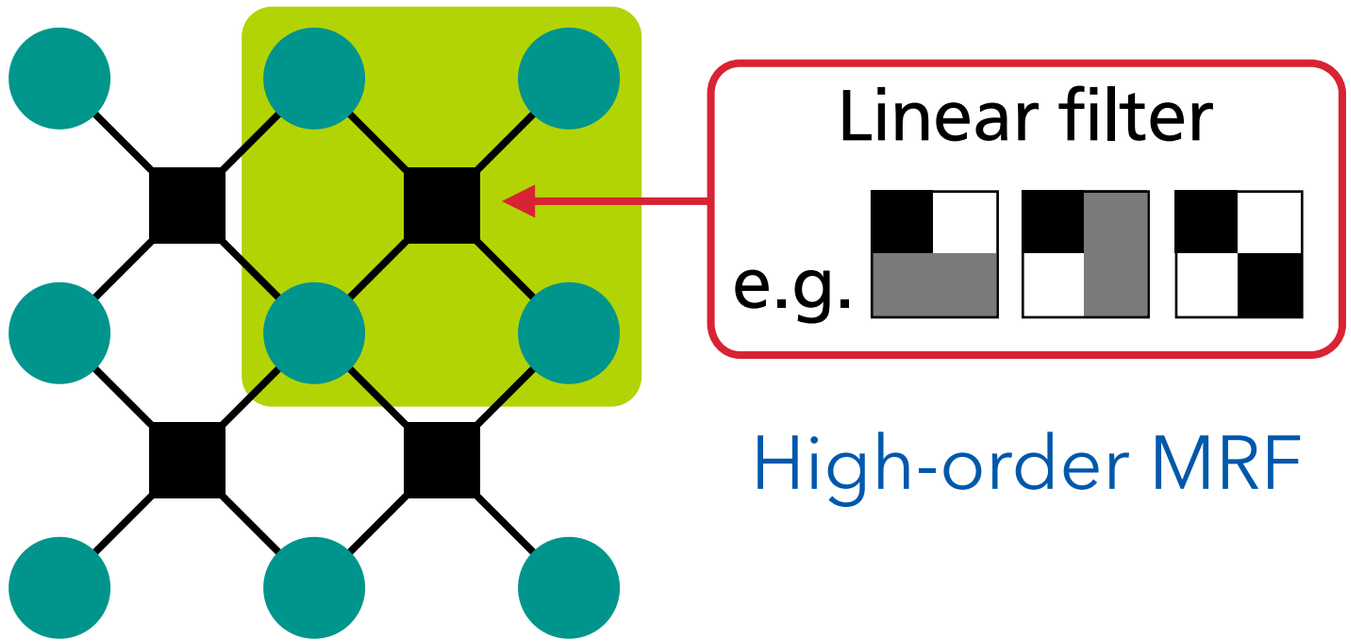


High-order MRF Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- Directly model extended pixel neighborhoods
  - Weaker conditional independence assumption (→ Markov blanket)
- High-dimensional potential functions
  - Model product of linear filter responses





High-order MRF Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

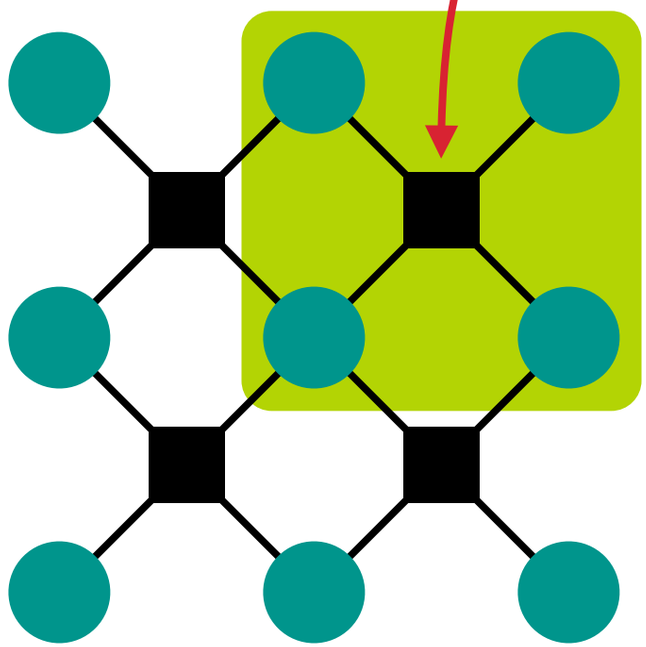
- Fields of Experts (FoE) [Roth & Black '05, '09]
  - Includes various pairwise & high-order MRFs

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon\|\mathbf{x}\|^2/2} \prod_{c\in\mathcal{C}} \prod_{i=1}^{N} \phi\left(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(c)}; \boldsymbol{\alpha}_i\right)$$

Image

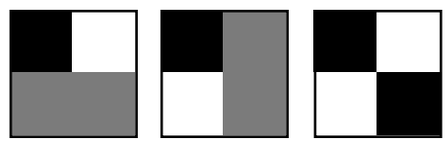
Parameters

$$\boldsymbol{\Theta} = \{\mathbf{J}_i, \boldsymbol{\alpha}_i\}_{i=1,\ldots,N}$$



Expert function

Linear filter

e.g. 

Vector of nodes in clique c

High-order MRF Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fields of Experts (FoE) [Roth & Black '05, '09]
  - Includes various pairwise & high-order MRFs

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon\|\mathbf{x}\|^2/2} \prod_{c\in\mathcal{C}} \prod_{i=1}^{N} \phi\left(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(c)};\boldsymbol{\alpha}_i\right)$$

Intractable partition function

Guarantees model to be normalizable [Weiss & Freeman '07]

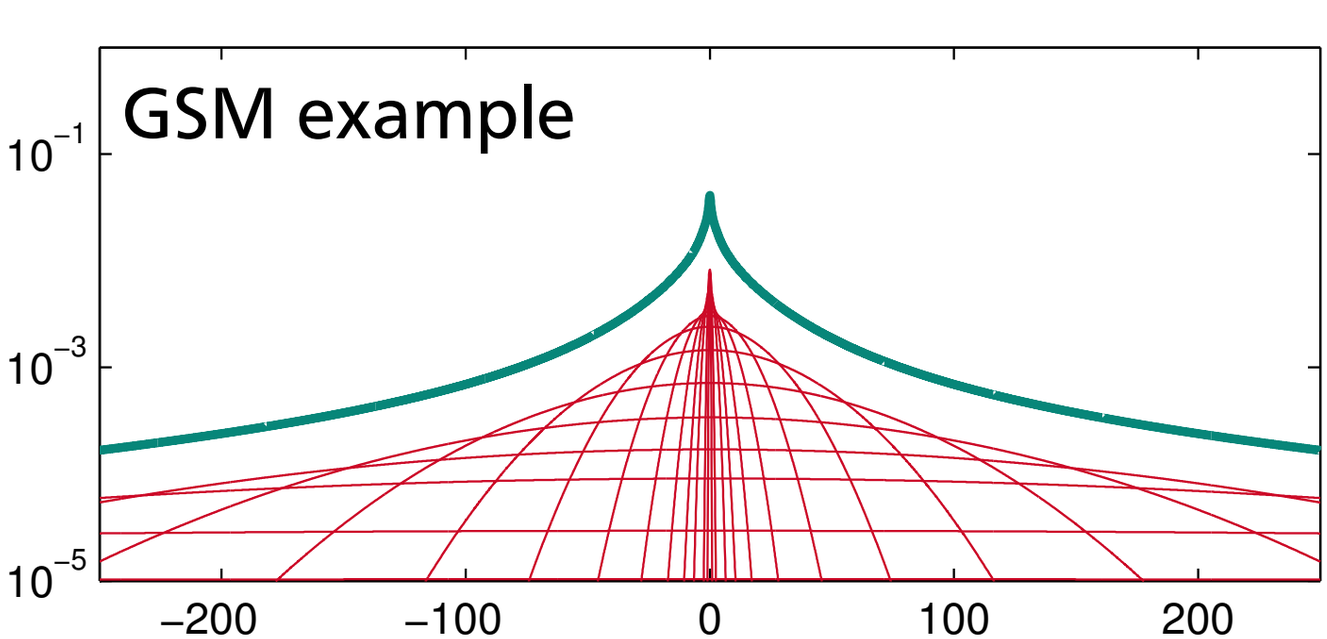
High-order MRF Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fields of Experts (FoE) [Roth & Black '05, '09]
  - Includes various pairwise & high-order MRFs

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon\|\mathbf{x}\|^2/2} \prod_{c\in\mathcal{C}} \prod_{i=1}^{N} \phi\left(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(c)}; \boldsymbol{\alpha}_i\right)$$

Mixture Weights



Gaussian Scale Mixture (GSM) Distribution

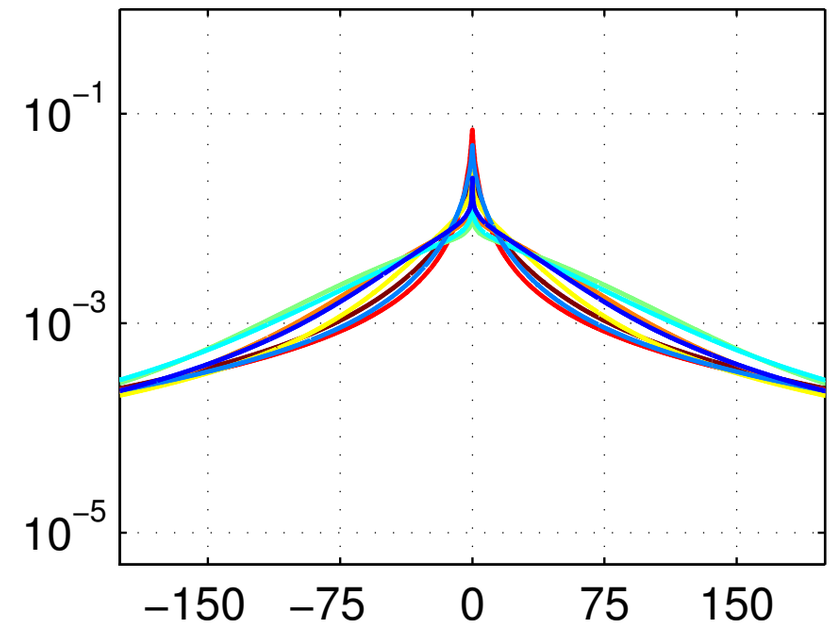
[Wainwright & Simoncelli '99, Weiss & Freeman '07]

$$\phi\left(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(c)}; \boldsymbol{\alpha}_i\right) = \sum_{j=1}^{J} \alpha_{ij} \cdot \mathcal{N}\left(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(c)}; 0, \sigma_i^2/s_j\right)$$

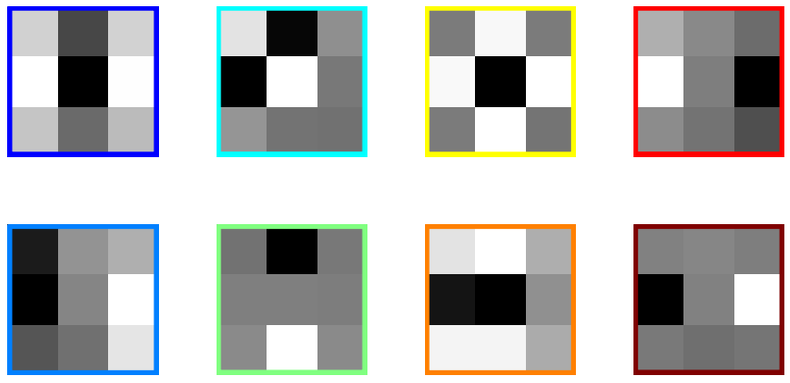
High-order MRF Prior

TECHNISCHE UNIVERSITÄT DARMSTADT

- More flexibility → more parameters
  - difficult to choose manually
- Parameter learning necessary to obtain good results



Potential (expert) functions and associated 3x3 filters of a learned fields of experts prior



Maximum Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Training data $\mathcal{D} = \{x_1, \ldots, x_n\}$
- Goal: estimate $p(x|\theta)$

- Aim to estimate one single θ
- Likelihood of θ

$$\mathcal{L}(\theta) = p(\mathcal{D}|\theta)$$

- Assume that the data is independent and identically distributed (iid)

$$\mathcal{L}(\theta) = p(\mathcal{D}|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

- Now choose θ such that it maximizes this likelihood

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta; \mathcal{D})$$

Maximum Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Maximum Likelihood

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta; \mathcal{D})$$

  is equivalent to

- minimizing the negative log-Likelihood

$$\begin{aligned} \theta_{ML} &= \underset{\theta}{\operatorname{argmax}} \ln \mathcal{L}(\theta) \\ &= \underset{\theta}{\operatorname{argmin}} - \ln \mathcal{L}(\theta) \\ &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^{n} \ln p(x_i|\theta) \end{aligned}$$

- Numerically more stable

MRF Learning – Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Markov Network defined on (not necessarily maximal) cliques

$$p(x \mid \theta) = \frac{1}{Z(\theta)} \prod_c \Phi_c(x_{(c)} \mid \theta_c)$$

- Clique variables $x_{(c)}$
- Partition function

$$Z(\theta) = \sum_x \prod_c \Phi_c(x_{(c)} \mid \theta_c)$$

- Training data $\mathcal{D} = \{x^1, \ldots, x^N\}$
- Log-Likelihood

$$\mathcal{L}(\theta; \mathcal{D}) = \sum_n \sum_c \log \Phi_c(x^n_{(c)} \mid \theta_c) - N \log Z(\theta)$$

MRF Learning – Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- The Likelihood (repeated from before)

$$L(\theta;\mathcal{D}) = \sum_n \sum_c \log \Phi_c(x^n_{(c)} \mid \theta_c) - N \log Z(\theta)$$

$$= N \left\langle \sum_c \log \Phi_c(x_{(c)} \mid \theta_c) \right\rangle_{q(x;\mathcal{D})} - N \log Z(\theta)$$

- with the empirical distribution

$$q(x;\mathcal{D}) = \frac{1}{N} \sum_n \delta(x - x^n)$$

Likelihood Gradient

TECHNISCHE UNIVERSITÄT DARMSTADT

- First step:

$$\begin{aligned}
\frac{\partial}{\partial \theta_d} \mathcal{L}(\theta; \mathcal{D}) &= N \frac{\partial}{\partial \theta_d} \left\langle \sum_c \log \Phi_c(x_{(c)} \mid \theta_c) \right\rangle_{q(x;\mathcal{D})} - N \frac{\partial}{\partial \theta_d} \log Z(\theta) \\
&= N \left\langle \sum_c \frac{\partial}{\partial \theta_d} \log \Phi_c(x_{(c)} \mid \theta_c) \right\rangle_{q(x;\mathcal{D})} - N \frac{\partial}{\partial \theta_d} \log Z(\theta) \\
&= N \left\langle \frac{\partial}{\partial \theta_d} \log \Phi_d(x_{(d)} \mid \theta_d) \right\rangle_{q(x;\mathcal{D})} - N \frac{\partial}{\partial \theta_d} \log Z(\theta)
\end{aligned}$$

- Second step:

$$\frac{\partial}{\partial \theta_d} \log Z(\theta) = \frac{\partial}{\partial \theta_d} \log \left(\int \prod_c \Phi_c(x_{(c)} \mid \theta_c) \, dx \right)$$

Likelihood Gradient

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\frac{\partial}{\partial x}\log f(x) = \frac{1}{f(x)}\cdot\frac{\partial}{\partial x}f(x)$$

$$\begin{aligned}
\frac{\partial}{\partial\theta_d}\log Z(\theta) &= \frac{\partial}{\partial\theta_d}\log\left(\int\prod_c \Phi_c(x_{(c)}\mid\theta_c)\;dx\right)\\
&= \frac{\frac{\partial}{\partial\theta_d}\int\prod_c \Phi_c(x_{(c)}\mid\theta_c)\;dx}{Z(\theta)}\\
&= \frac{1}{Z(\theta)}\int\left[\frac{\partial}{\partial\theta_d}\Phi_d(x_{(d)}\mid\theta_d)\right]\prod_{c\neq d}\Phi_c(x_{(c)}\mid\theta_c)\;dx\\
&= \frac{1}{Z(\theta)}\int\left[\frac{\partial}{\partial\theta_d}\log\Phi_d(x_{(d)}\mid\theta_d)\right]\prod_{c}\Phi_c(x_{(c)}\mid\theta_c)\;dx\\
&= \int\left[\frac{\partial}{\partial\theta_d}\log\Phi_d(x_{(d)}\mid\theta_d)\right]p(x\mid\theta)dx
\end{aligned}$$

Likelihood Gradient

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{aligned}
\frac{\partial}{\partial \theta_d} \log Z(\theta) &= \int \left[\frac{\partial}{\partial \theta_d} \log \Phi_d(x_{(d)} \mid \theta_d) \right] p(x \mid \theta) dx \\
&= \int \left[\frac{\partial}{\partial \theta_d} \log \Phi_d(x_{(d)} \mid \theta_d) \right] p(x_{(d)} \mid \theta) dx_{(d)} \\
&= \left\langle \frac{\partial}{\partial \theta_d} \log \Phi_d(x_{(d)} \mid \theta_d) \right\rangle_{p(x_{(d)} \mid \theta)}
\end{aligned}
$$

Likelihood Gradient

TECHNISCHE UNIVERSITÄT DARMSTADT

- Finally – Likelihood gradient:

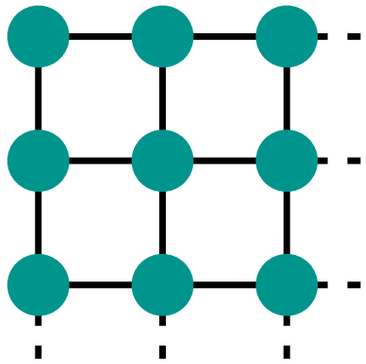
$$\frac{\partial}{\partial \theta_d}\mathcal{L}(\theta;\mathcal{D}) = N\left\langle \frac{\partial}{\partial \theta_d}\log \Phi_d(x_{(d)} \mid \theta_d)\right\rangle_{q(x;\mathcal{D})} - N\frac{\partial}{\partial \theta_d}\log Z(\theta)$$

$$= N\left\langle \frac{\partial}{\partial \theta_d}\log \Phi_d(x_{(d)} \mid \theta_d)\right\rangle_{q(x;\mathcal{D})} - N\left\langle \frac{\partial}{\partial \theta_d}\log \Phi_d(x_{(d)} \mid \theta_d)\right\rangle_{p(x_{(d)}|\theta)}$$

- Problem: last term depends on $p(x_{(d)} \mid \theta)$
- Need to take an average over the model distribution $p(x \mid \theta)$
- Learning requires inference!

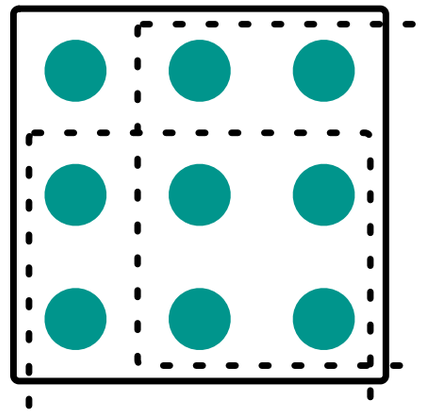
MRF Sampling – Example

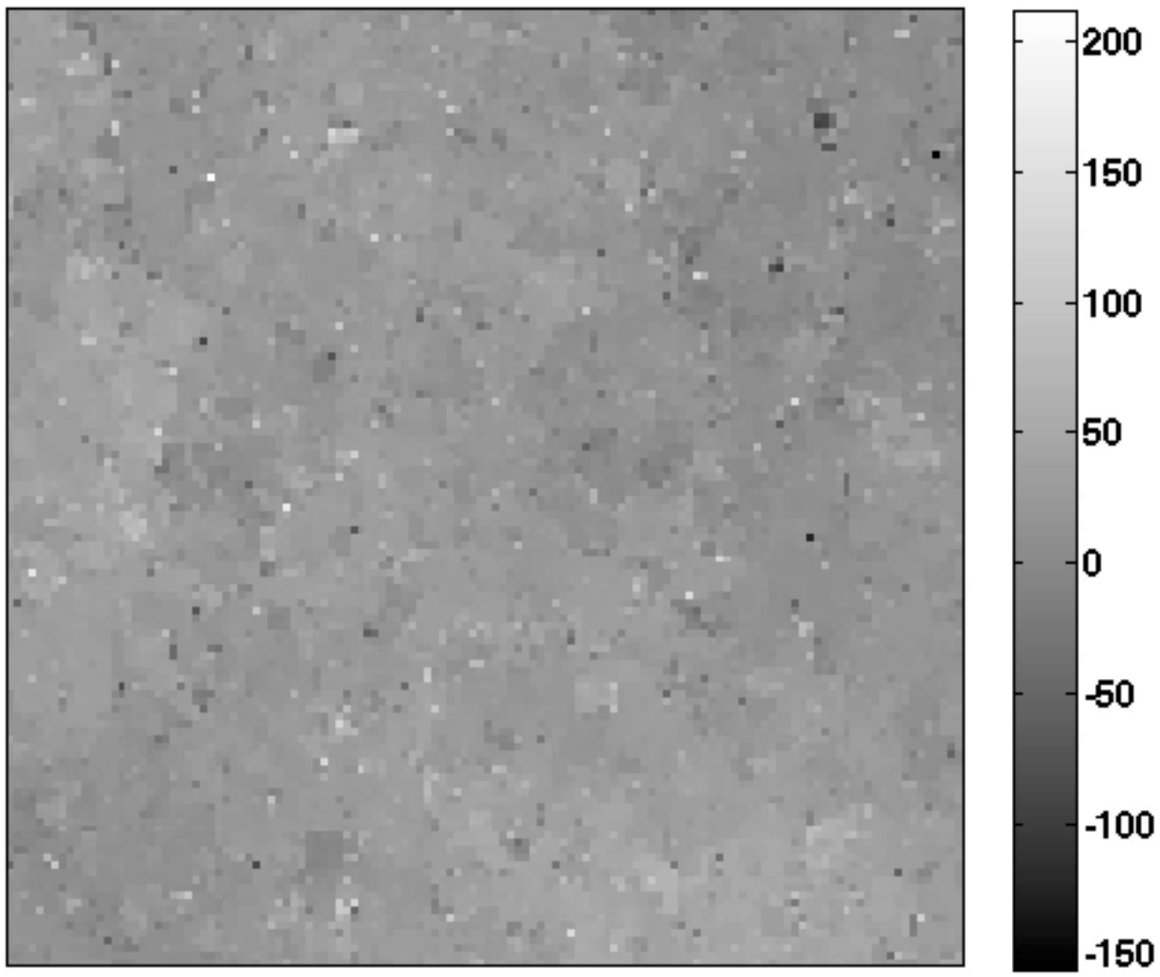
TECHNISCHE UNIVERSITÄT DARMSTADT

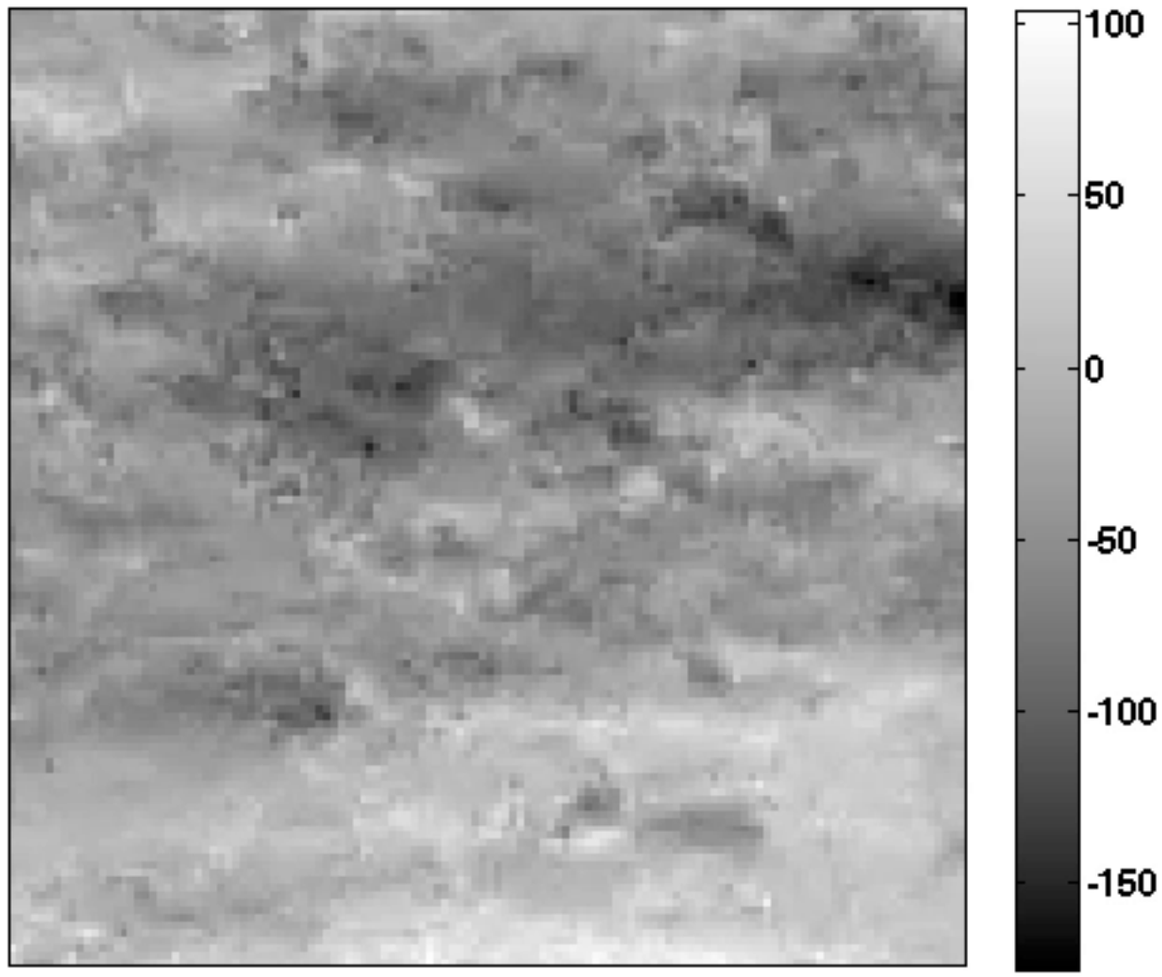


Pairwise MRF

High-order MRF
with 3×3 cliques







Generative Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

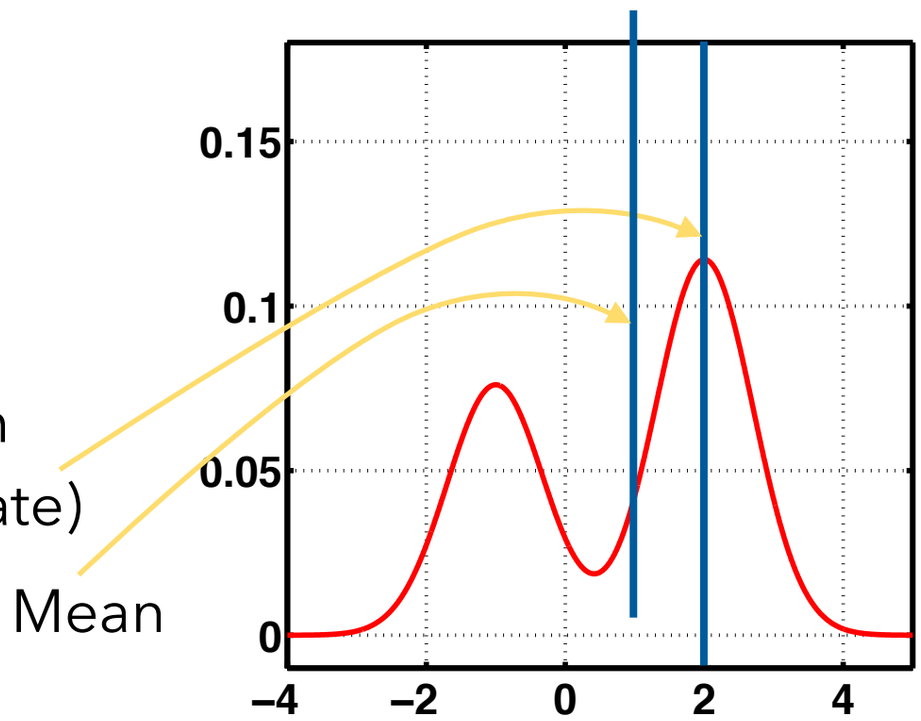
- We have discussed so far “generative” learning/training
- Image prior independent of application
  - Same prior can be used for denoising, super-resolution, etc.

- Training only requires “good” natural images
- Learning unfortunately difficult due to intractable normalization constant

Probabilistic Inference

TECHNISCHE UNIVERSITÄT DARMSTADT

- We learned a prior and combined it with the likelihood
- How do we infer a single solution from the posterior?
- Common choices:
  - Maximum
  - Mean
- How do these differ?
  - Assume we have the following posterior distribution:
  - The posterior may be multi-modal!



Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Your physician tells you there's a 1% chance of you having a disease. Should you get treated now?
  - A drug to treat the disease *now* costs 10 €.
  - *Later* treatment requires an operation that costs 10000 €.

- Decision to get treated *now*. Expected cost/loss:

$$\Delta(\text{now}|\text{healthy}) \cdot p(\text{healthy}) + \Delta(\text{now}|\text{sick}) \cdot p(\text{sick}) = 10 \cdot 0.99 + 10 \cdot 0.01 = 10$$

- Decision to get treated *later*. Expected cost/loss:

$$\Delta(\text{later}|\text{healthy}) \cdot p(\text{healthy}) + \Delta(\text{later}|\text{sick}) \cdot p(\text{sick}) = 0 \cdot 0.99 + 10000 \cdot 0.01 = 100$$

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Minimize expected loss to choose restored image

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}'} \int_{\mathbf{x}} \Delta(\mathbf{x}', \mathbf{x}) \cdot p(\mathbf{x}|\mathbf{y}) \, \mathrm{d}\mathbf{x}$$

- Loss function: $\Delta(\mathbf{x}', \mathbf{x})$
- Posterior distribution: $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$

- Choice of loss function determines decision of restored image (called estimator)
  - in turn, each estimator implies a loss function

Loss Functions & Estimators

TECHNISCHE UNIVERSITÄT DARMSTADT

- Maximum a posteriori (MAP) estimation implies 0-1 loss

$$\mathbf{x}_{\mathrm{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}}\, p(\mathbf{x}|\mathbf{y}) \qquad \Delta(\mathbf{x}', \mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x}' = \mathbf{x} \\ 1 & \text{if } \mathbf{x}' \neq \mathbf{x} \end{cases}$$

- Minimum mean squared error (MMSE) estimation

$$\mathbf{x}_{\mathrm{MMSE}} = \underset{\mathbf{x}'}{\operatorname{argmin}} \int_{\mathbf{x}} \underbrace{\|\mathbf{x}' - \mathbf{x}\|^2}_{\Delta(\mathbf{x}', \mathbf{x})} p(\mathbf{x}|\mathbf{y}) d\mathbf{x}$$

$$= \underset{\mathbf{x}'}{\operatorname{argmin}} \left[\|\mathbf{x}'\|^2 - 2 \left(\int_{\mathbf{x}} \mathbf{x} \cdot p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \right)^{\mathsf{T}} \mathbf{x}' \right] = E(\mathbf{x}|\mathbf{y})$$

Can we do better? Yes.

TECHNISCHE UNIVERSITÄT DARMSTADT

4-connected MRF
MAP estimation

3x3 Fields of Experts
MMSE estimation

Generative vs. Discriminative

TECHNISCHE UNIVERSITÄT DARMSTADT

Generative

- model $p(\mathbf{x})$
- train on natural images

✔ application neutral

✘ learning & inference is more difficult

Discriminative

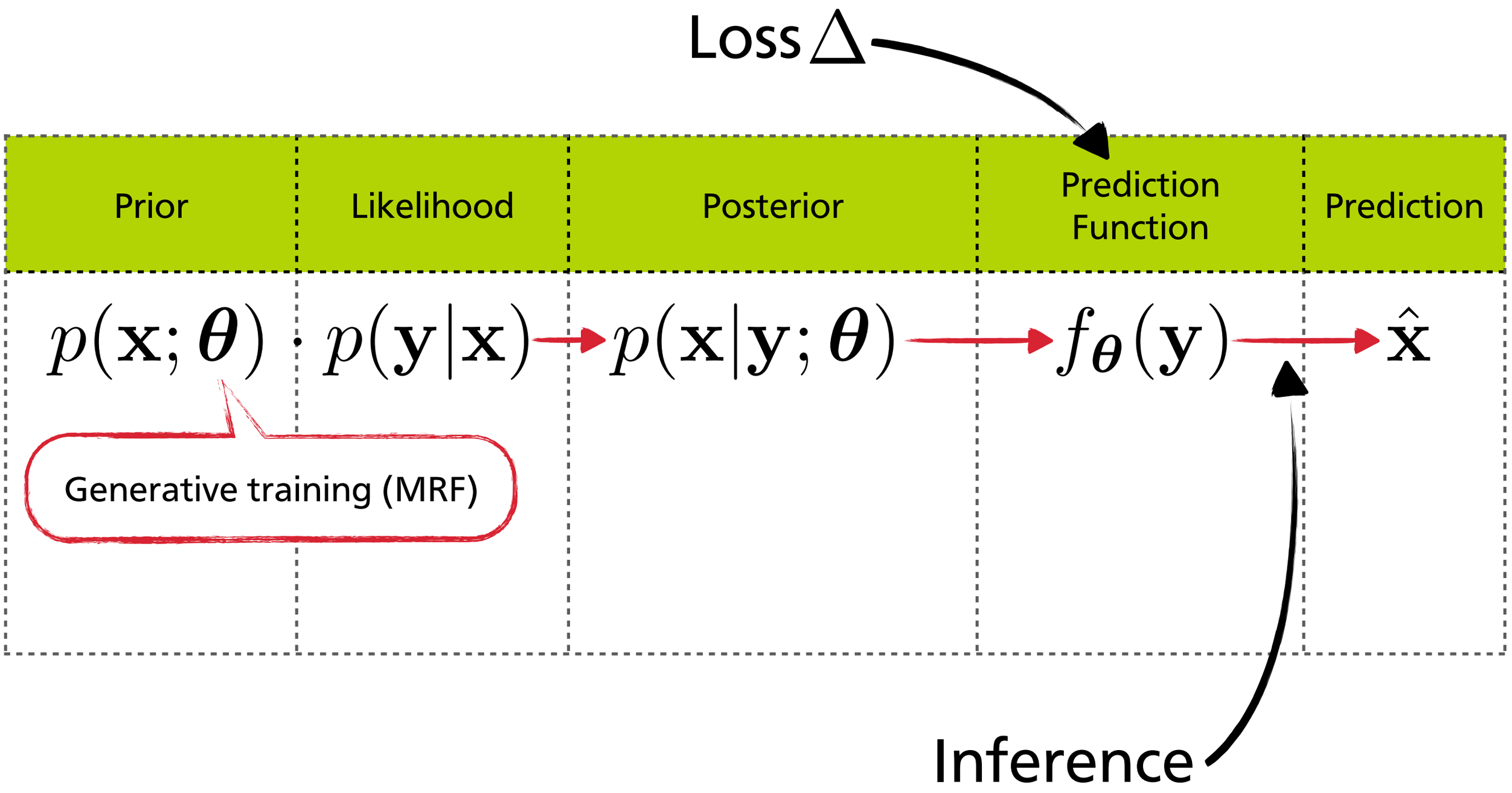
- model $p(\mathbf{x}|\mathbf{y})$
- train on input/output pairs

✘ application specific

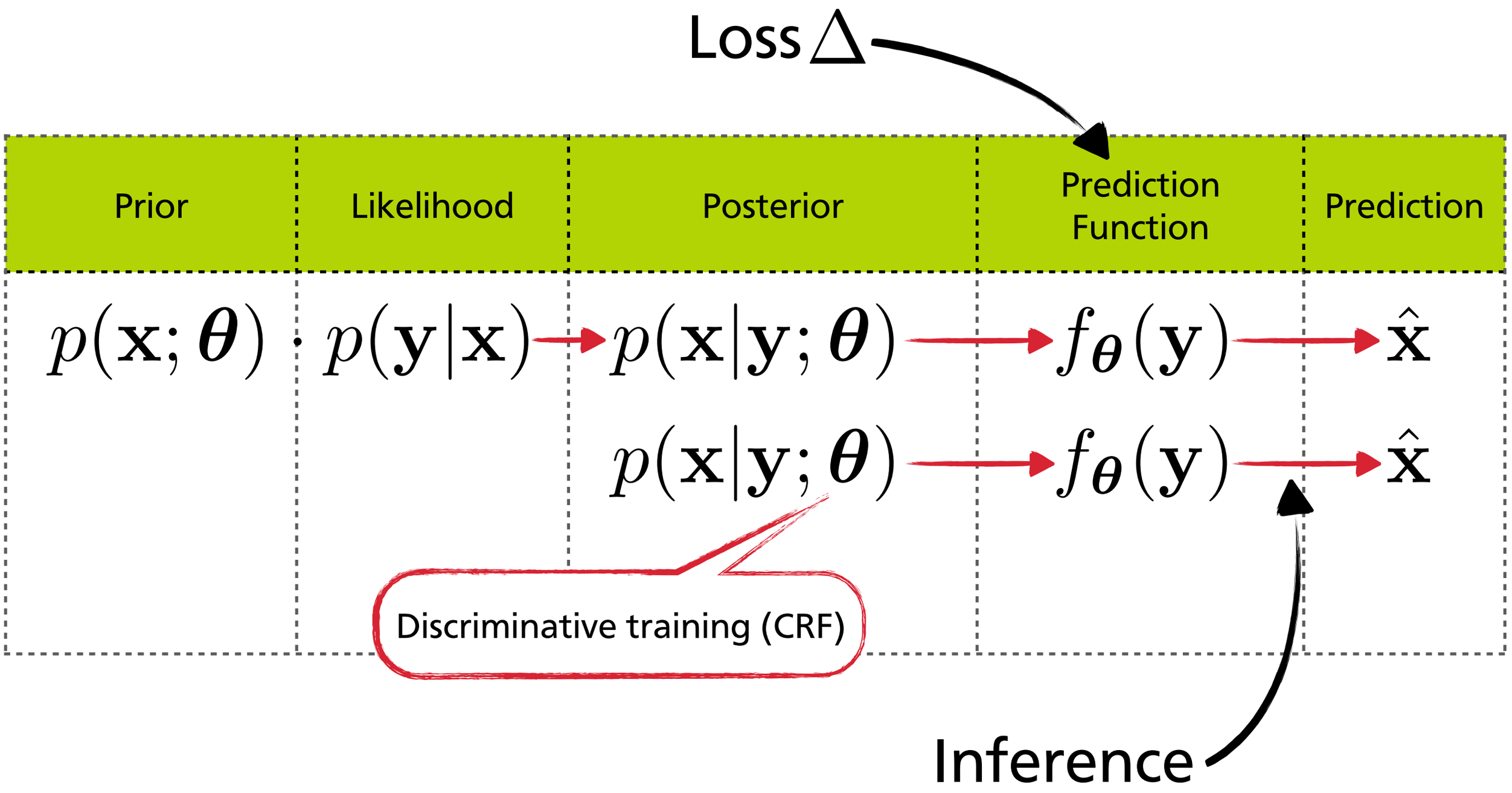
✔ learning & inference are generally easier

✔ can lead to better performance

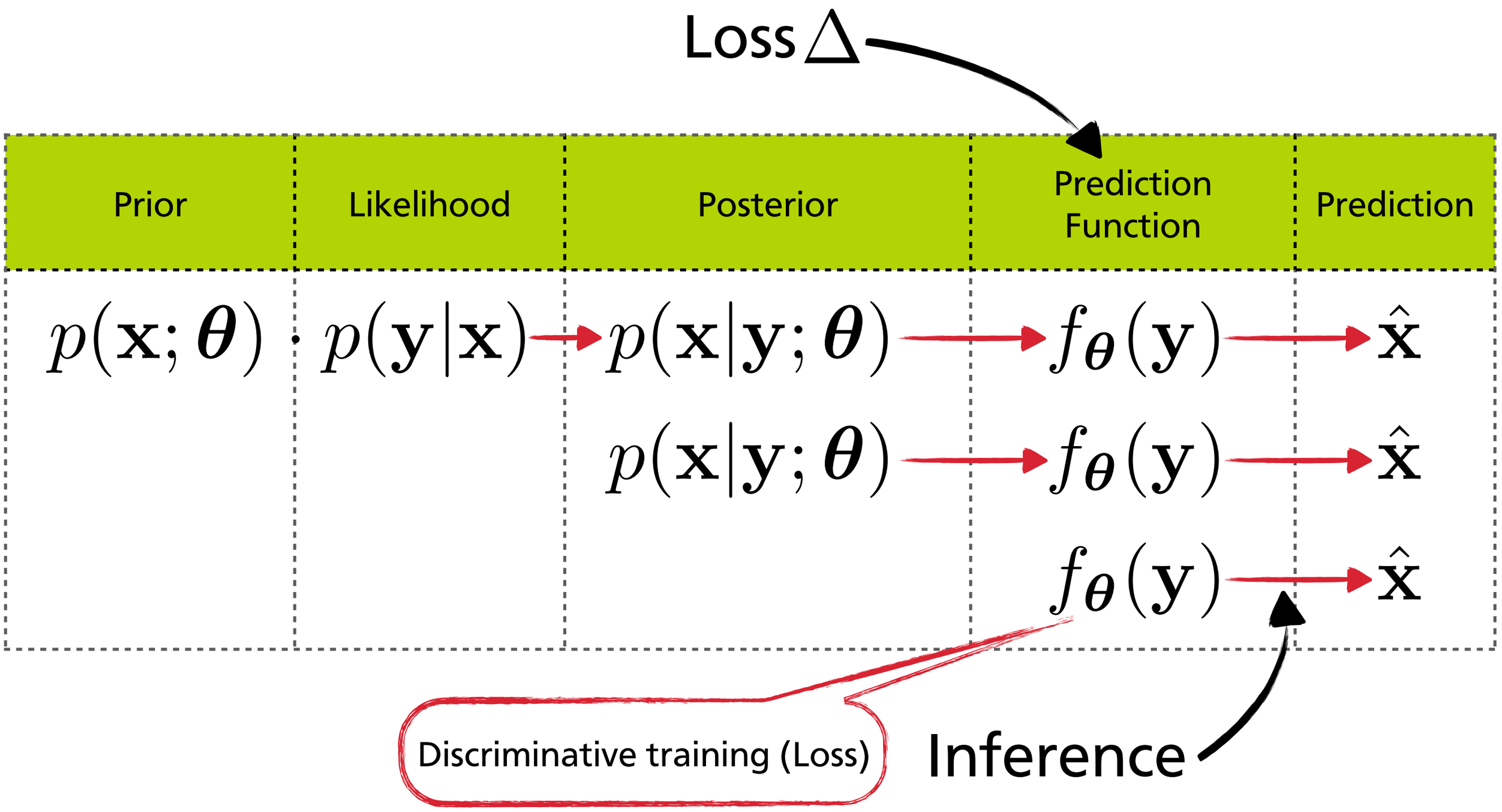
Generative vs. Discriminative Training



Generative vs. Discriminative Training



Generative vs. Discriminative Training



Discriminative Training

TECHNISCHE UNIVERSITÄT DARMSTADT

- Two types of discriminative training
  - probabilistic CRF training of conditional likelihood
  - loss-based training (not probabilistic)

- Maximum Conditional likelihood learning also has problem of intractable normalization constant

Loss-based Training

TECHNISCHE UNIVERSITÄT DARMSTADT

- Define "posterior" in terms of an energy function $E(\mathbf{x}|\mathbf{y})$

$$p(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) \propto \exp\left(-E(\mathbf{x}|\mathbf{y};\boldsymbol{\theta})\right)$$

- Prediction via energy minimization:

$$f_{\boldsymbol{\theta}}(\mathbf{y}) = \arg\min_{\mathbf{x}} E(\mathbf{x}|\mathbf{y};\boldsymbol{\theta})$$

- Parameter learning with $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \Delta(\mathbf{x}_i, f_{\boldsymbol{\theta}}(\mathbf{y}_i))$$

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Loss-based Training

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nested optimization problems. How to solve?

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \Delta(\mathbf{x}_i, f_{\boldsymbol{\theta}}(\mathbf{y}_i))$$

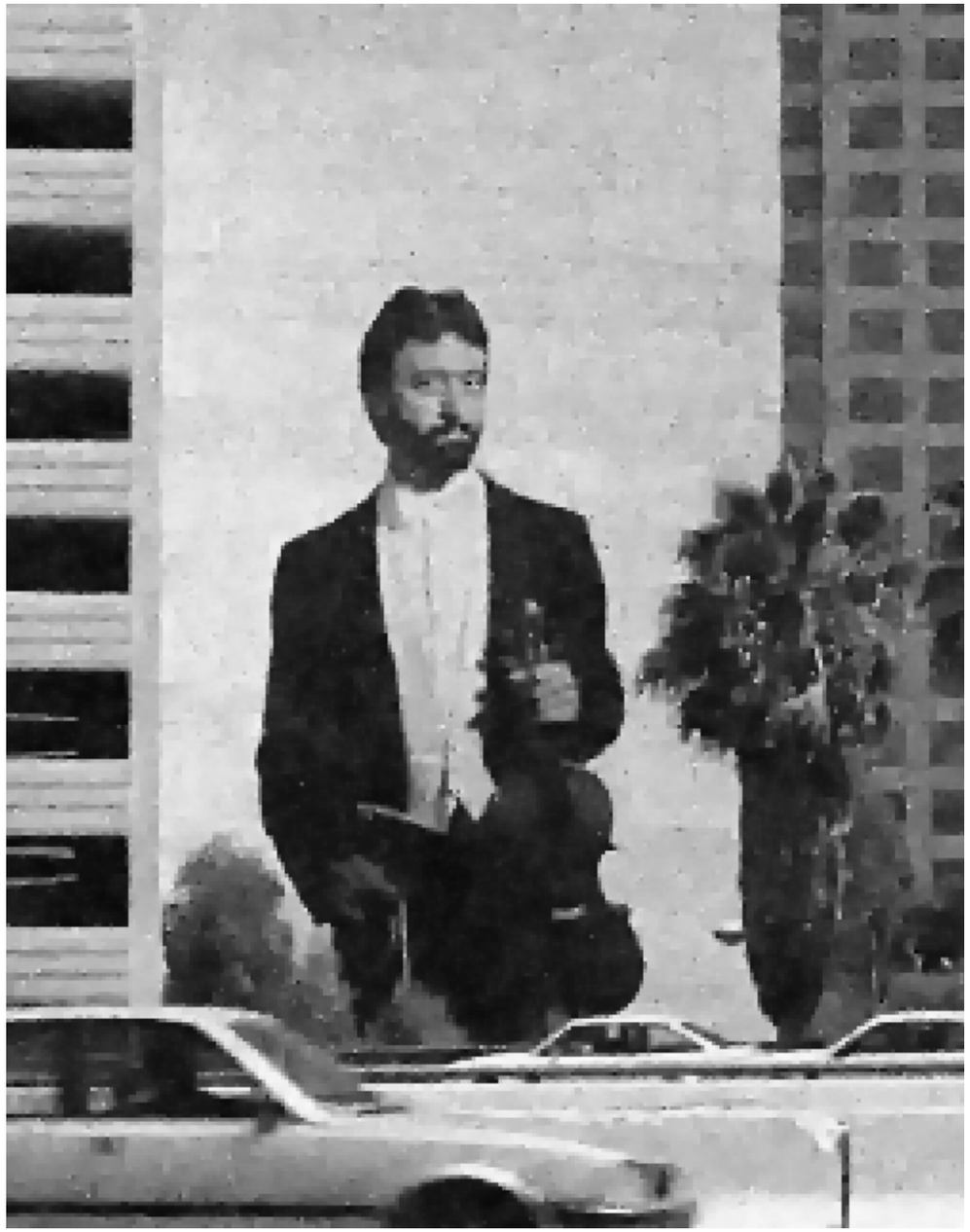
$$f_{\boldsymbol{\theta}}(\mathbf{y}) = \arg\min_{\mathbf{x}} E(\mathbf{x}|\mathbf{y}; \boldsymbol{\theta})$$

- Options:
  - Prediction $f_{\boldsymbol{\theta}}(\mathbf{y})$ and gradient $\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \boldsymbol{\theta}}$ computable in closed form
  - Bilevel optimization with implicit differentiation
    - requires Hessian-vector products, often slow
  - "Truncated" optimization

Can we do better? Yes.

TECHNISCHE UNIVERSITÄT DARMSTADT

4-connected MRF, MAP
Hand-defined



4-connected MRF, MAP
Loss-based training

Can we do better? Yes.

TECHNISCHE UNIVERSITÄT DARMSTADT

3x3 Fields of Experts, MMSE
Generative training

3x3 Fields of Experts, MAP
Loss-based training

Generative vs. Discriminative

TECHNISCHE UNIVERSITÄT DARMSTADT

Generative

- model $p(\mathbf{x})$
- train on natural images

✔ application neutral

✘ learning & inference is more difficult

Discriminative

- model $p(\mathbf{x}|\mathbf{y})$
- train on input/output pairs

✘ application specific

✔ learning & inference are generally easier

✔ can lead to better performance

Probabilistic vs. Loss-based

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Probabilistic**
  - ✘ model misspecification
  - ✔ handle unobserved variables
  - ✔ not restricted to specific loss
  - ✘ often expensive computation
  - ✔ able to draw samples
- **Loss-based** (discriminative)
  - ✔ model misspecification
  - ✘ handle unobserved variables
  - ✘ re-training for other losses
  - ✔ often faster runtime
  - ✘ labeled data for training