Modeling images

Subhransu Maji

CMPSCI 670: Computer Vision

December 6, 2016

Administrivia

- This is the last lecture! Next two will be project presentations by you
  - Upload your presentations on Moodle by 11 AM, Thursday, Dec. 8
  - 6 min presentation + 2 mins of questions
  - The order of presentations will be chosen randomly

- Remaning grading
  - Homework 3 will be posted later today
  - Homework 4 (soon)

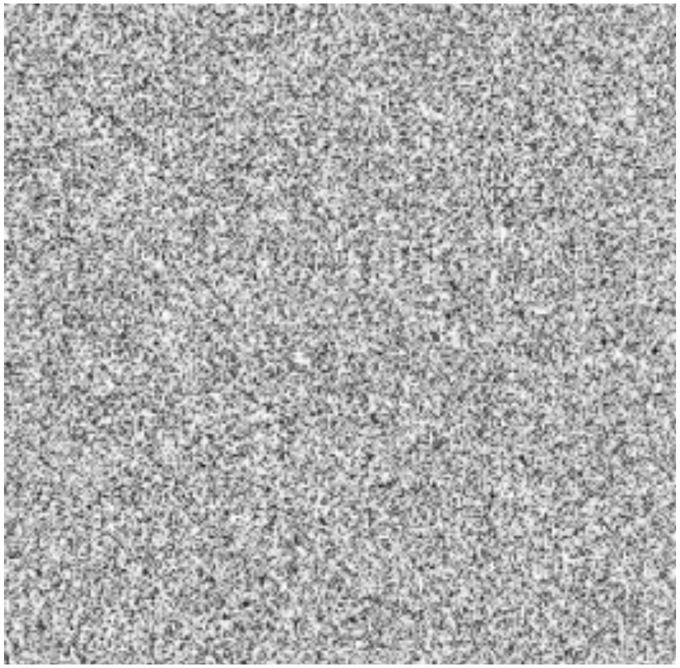
- Questions?

Modeling images

- Learn a probability distribution over natural images

Image credit: Flickr @Kenny (zoompict) Teo

$P(\mathbf{x}) \sim 1$



$P(\mathbf{x}) \sim 0$

- Many applications:
  - image synthesis: sample **x** from P(**x**)
  - image denoising: find most-likely clean image given a noisy image
  - image deblurring: find most-likely crisp image given a blurry image

Modeling images: challenges

- How many 64x64 pixels binary images are there?

10 random 64x64 binary images

$$2^{64 \times 64} \sim 10^{400}$$
$$\text{atoms in the known universe: } 10^{80}$$

- Assumption
  - Each pixel is generated *independently*

$$P(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{64,64}) = P(\mathbf{x}_{1,1})P(\mathbf{x}_{1,2})\ldots P(\mathbf{x}_{64,64})$$

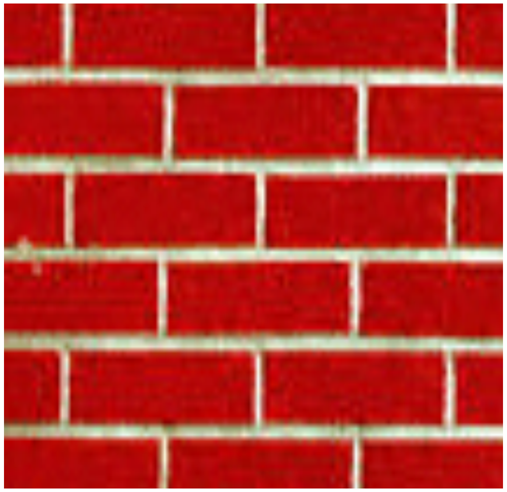
  - Is this a good assumption?

Texture synthesis

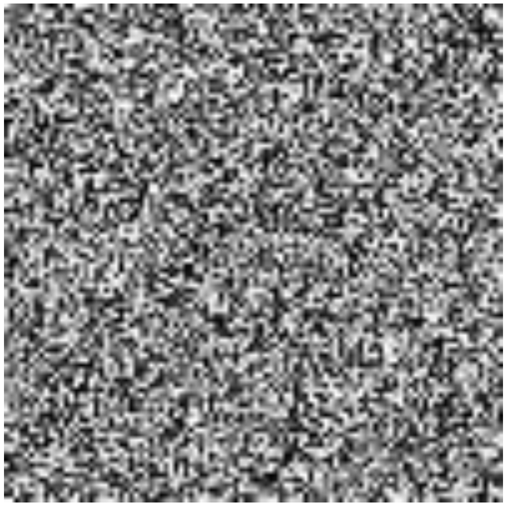
- **Goal:** create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces

The challenge

- Need to model the whole spectrum: from repeated to stochastic texture



repeated



stochastic

Both?

Alexei A. Efros and Thomas K. Leung, “Texture Synthesis by Non-parametric Sampling,” Proc. International Conference on Computer Vision (ICCV), 1999.

Markov chains

Markov chain

- A *sequence* of random variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$
- $\mathbf{x}_t$ is the **state** of the model at time t

$$\boxed{\mathbf{x}_1} \rightarrow \boxed{\mathbf{x}_2} \rightarrow \boxed{\mathbf{x}_3} \rightarrow \boxed{\mathbf{x}_4} \rightarrow \boxed{\mathbf{x}_5}$$

- **Markov assumption**: each state is dependent only on the previous one
  - dependency given by a **conditional probability**:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1})$$

- The above is actually a *first-order* Markov chain
- An *N'th-order* Markov chain:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-N})$$

Source: S. Seitz

Markov chain example: Text

"A dog is a man's best friend. It's a dog eat dog world out there."

Rows: $\mathbf{x}_{t-1}$; columns: $\mathbf{x}_t$

$\mathbf{x}_{t-1}$ \ $\mathbf{x}_t$	a	dog	is	man's	best	friend	it's	eat	world	out	there	.
a		2/3		1/3								
dog			1/3					1/3	1/3			
is	1											
man's					1							
best						1						
friend												1
it's	1											
eat		1										
world										1		
out											1	
there												1
.							1					

$$p(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Source: S. Seitz

Text synthesis

Create plausible looking poetry, love letters, term papers, etc.

Most basic algorithm

1. Build probability histogram
   - find all blocks of N consecutive words/letters in training documents
   - compute probability of occurrence $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-(n-1)})$
2. Given words $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k-1}$
   - compute $\mathbf{x}_k$ by sampling from $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-(n-1)})$

WE NEED TO EAT CAKE

Source: S. Seitz

Text synthesis

"As I've commented before, really relating to someone involves standing next to impossible."

"One morning I shot an elephant in my arms and kissed him."

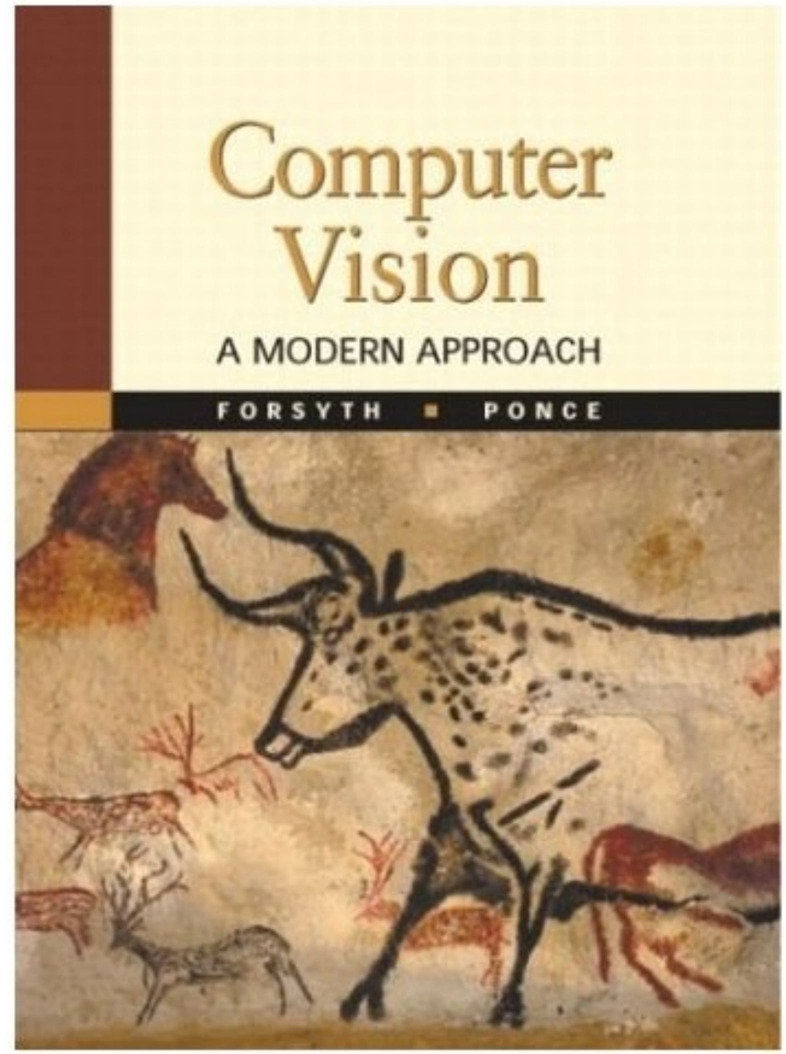
"I spent an interesting evening recently with a grain of salt"

Dewdney, "A potpourri of programmed prose and prosody" *Scientific American,* 1989.

Slide from Alyosha Efros, ICCV 1999

Synthesizing computer vision text

What do we get if we extract the probabilities from a chapter on Linear Filters, and then synthesize new statements?



Check out Yisong Yue's website implementing text generation: build your own text Markov Chain for a given text corpus. http://www.yisongyue.com/shaney/index.php

Kristen Grauman

Synthesized text

- This means we cannot obtain a separate copy of the best studied regions in the sum.
- All this activity will result in the primate visual system.
- The response is also Gaussian, and hence isn't bandlimited.
- Instead, we need to know only its response to any data vector, we need to apply a low pass filter that strongly reduces the content of the Fourier transform of a very large standard deviation.
- It is clear how this integral exist (it is sufficient for all pixels within a $2k + 1 \times 2k + 1 \times 2k + 1 \times 2k + 1$ — required for the images separately.

Kristen Grauman

Markov random field

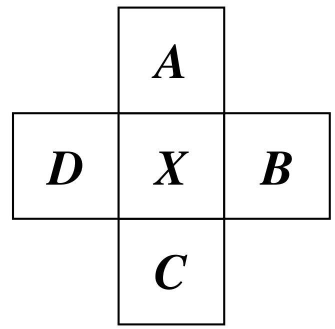
A Markov random field (MRF)

- generalization of Markov chains to two or more dimensions.

First-order MRF:

- probability that pixel *X* takes a certain value given the values of neighbors *A*, *B*, *C*, and *D*:

$$P(\mathbf{X}|\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$$



Source: S. Seitz

Texture synthesis

Can apply 2D version of text synthesis

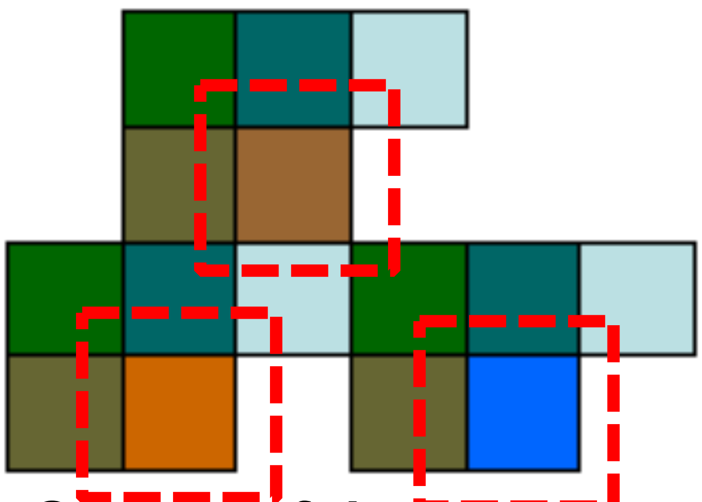
Texture corpus
(sample)

Output

[**Efros & Leung, ICCV 99**]()

Texture synthesis: intuition

- Before, we inserted the next word based on existing nearby words…
- Now we want to insert pixel intensities based on existing nearby pixel values.

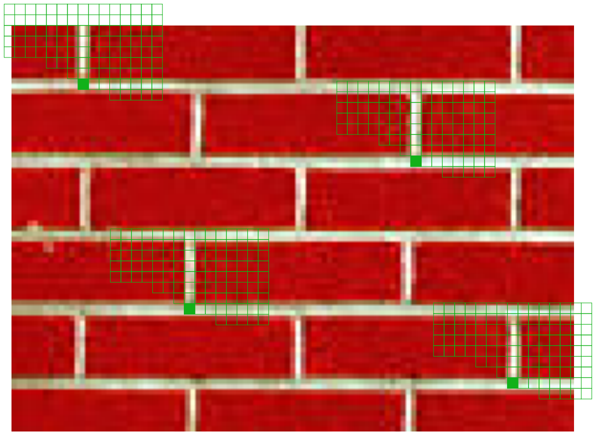


Sample of the texture (“corpus”)

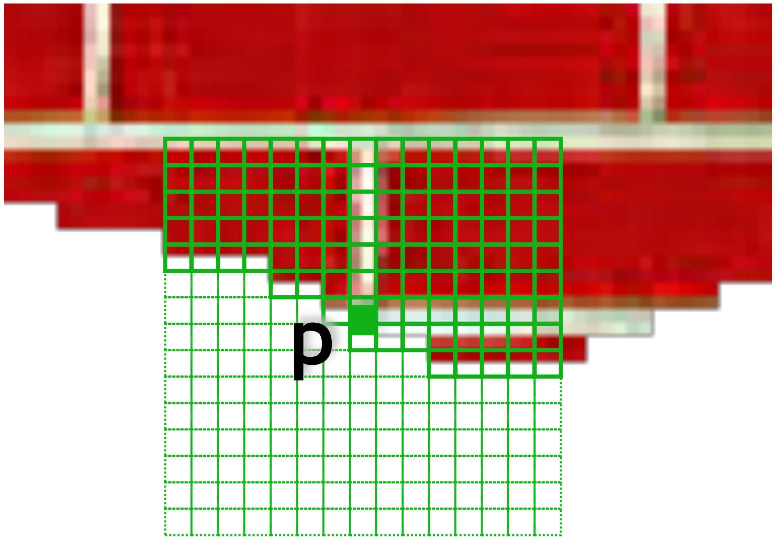
Place we want to insert next

Distribution of a value of a pixel is conditioned on its neighbors alone.

Synthesizing one pixel



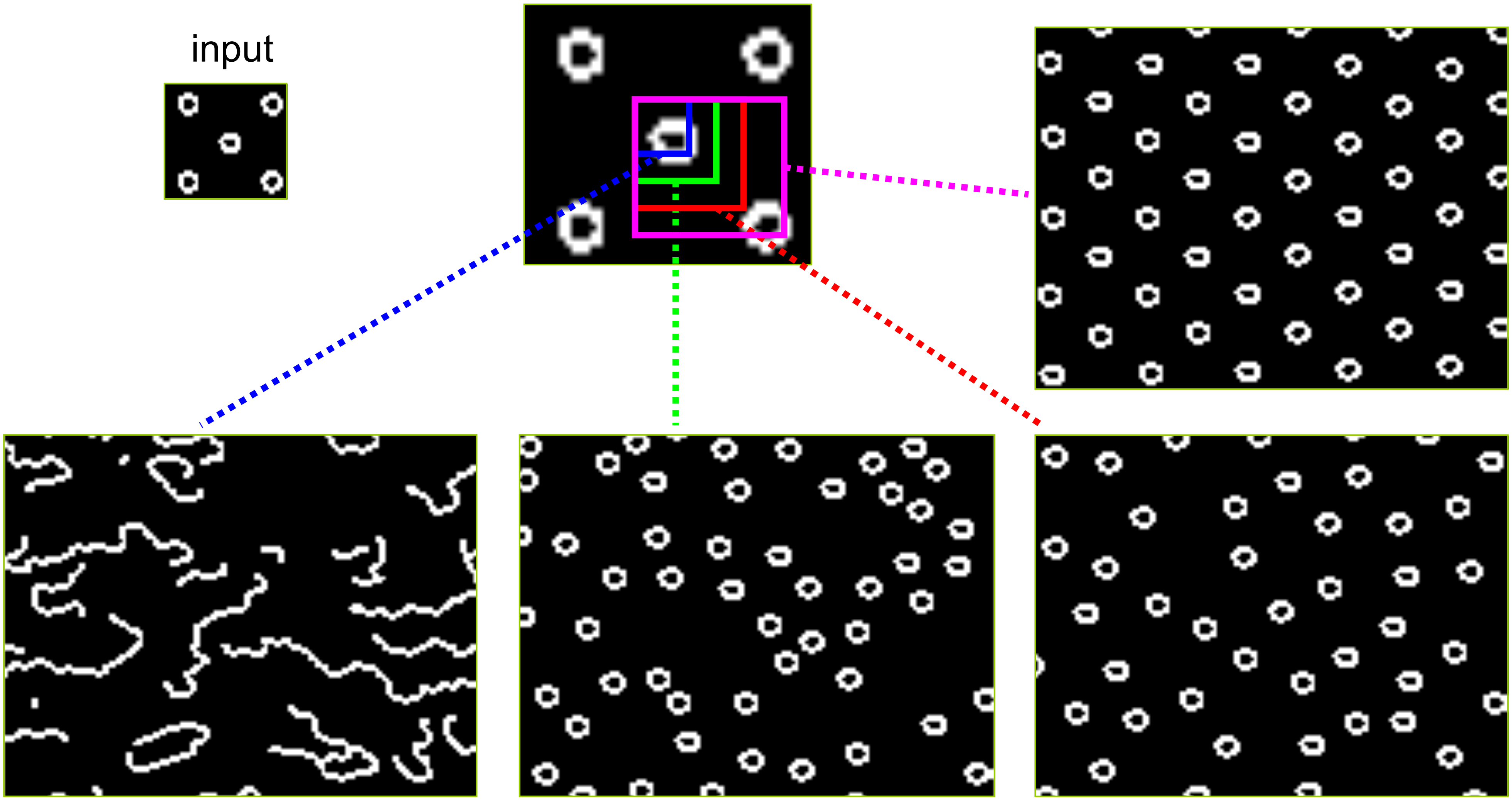
input image



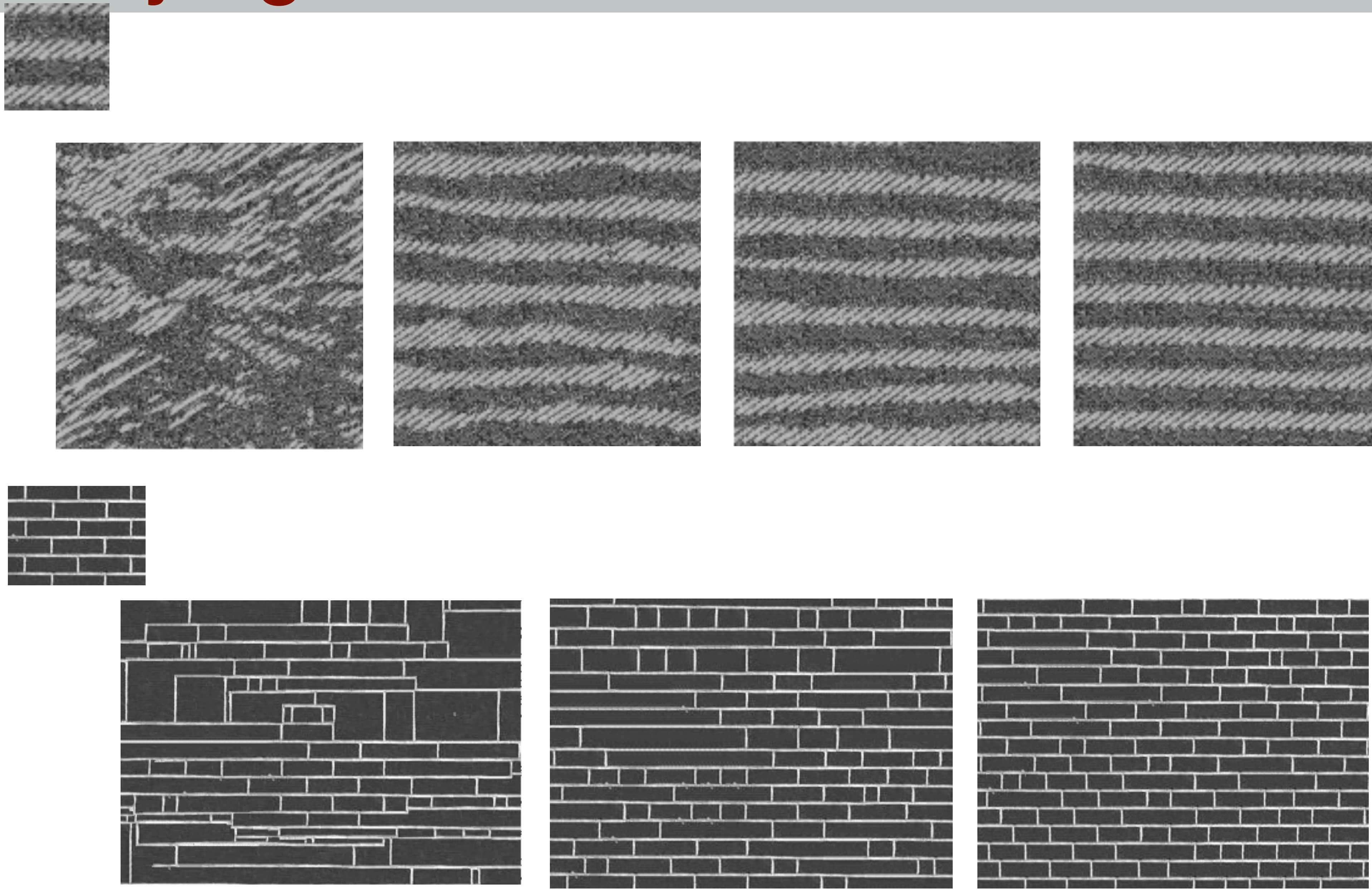
synthesized image

- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- To synthesize **x**
  - ➡ pick one matching window at random
  - ➡ assign **x** to be the center pixel of that window
- An **exact** neighbourhood match might not be present, so find the **best** matches using **SSD error** and randomly choose between them, preferring better matches with higher probability

Neighborhood window



Varying window size



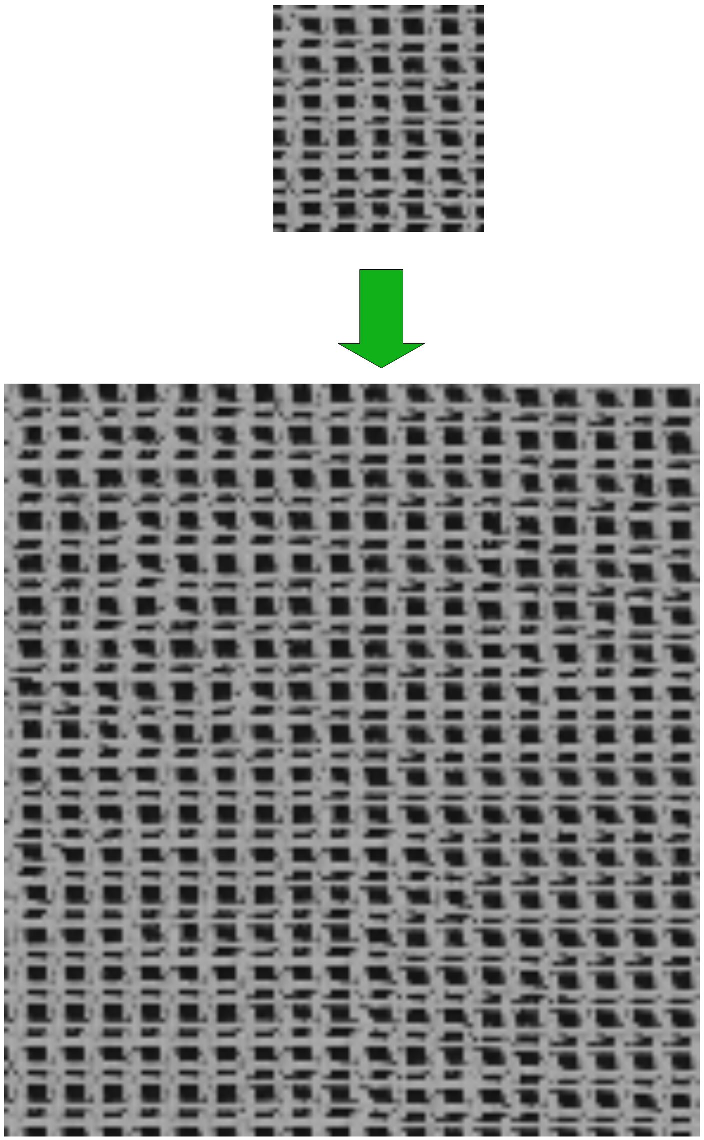
Increasing window size

Growing texture

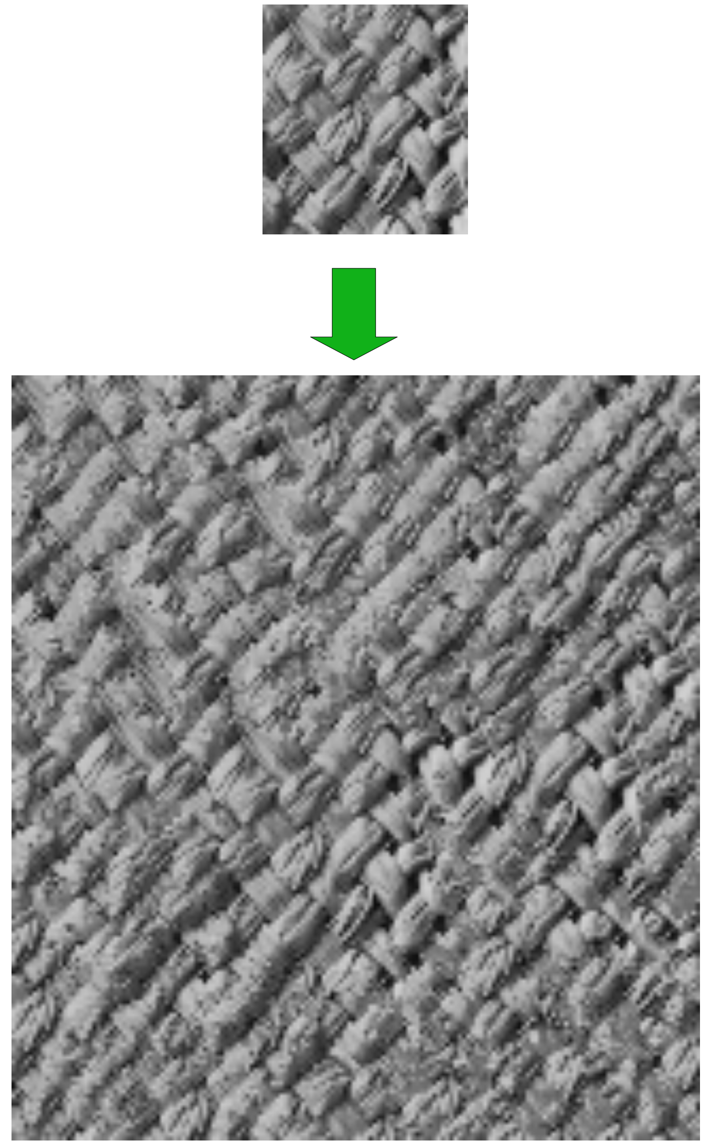
- Starting from the initial image, "grow" the texture one pixel at a time

Synthesis results

french canvas



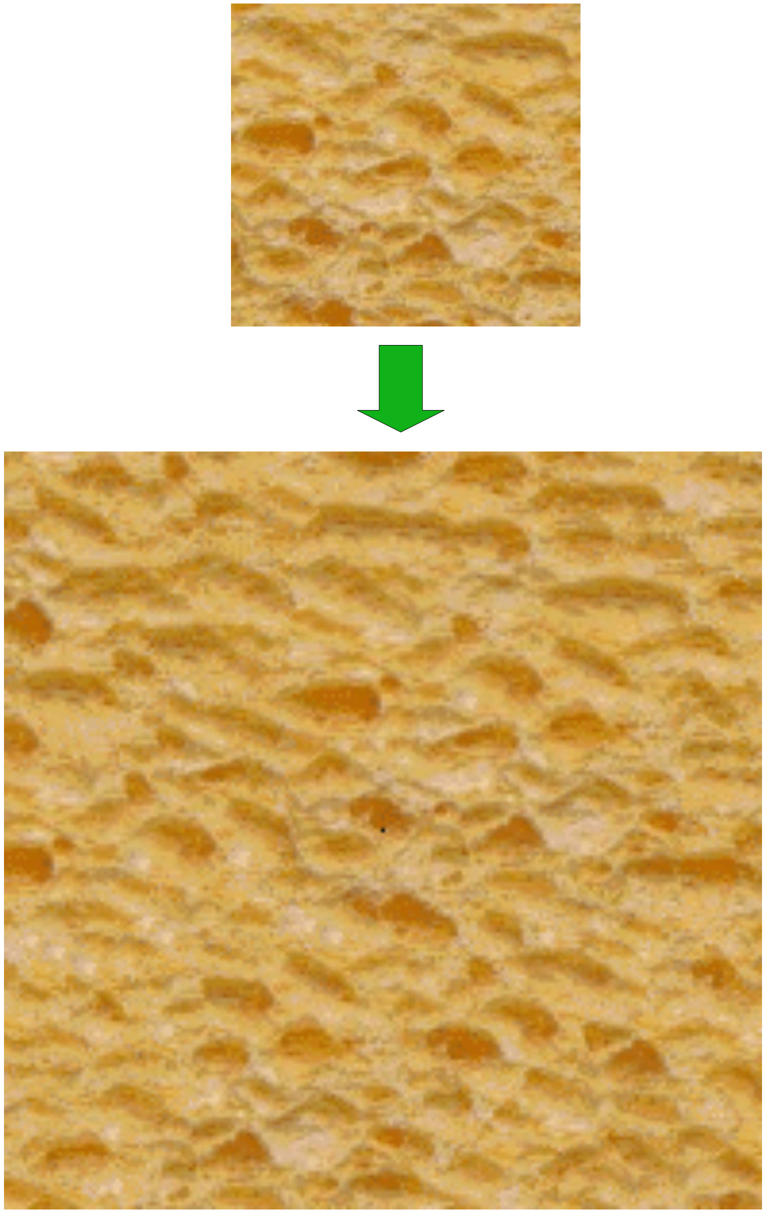
rafia weave



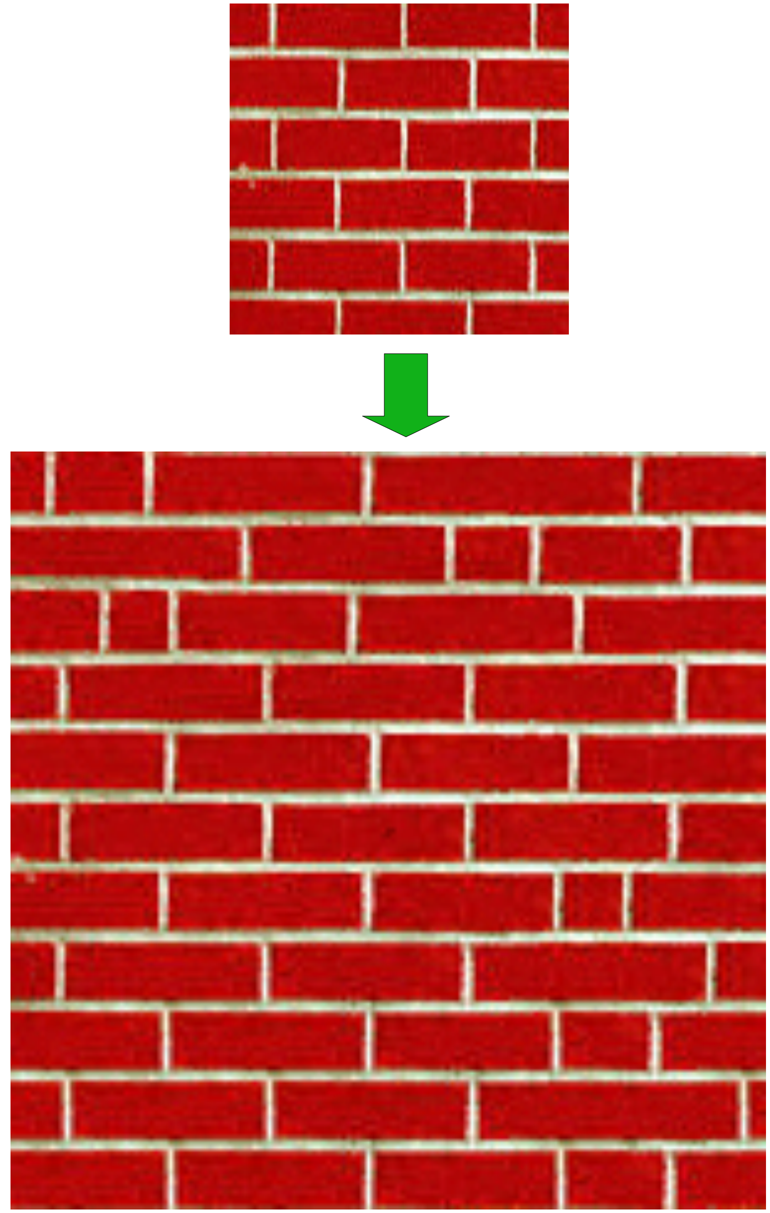
Slide from Alyosha Efros, ICCV 1999

Synthesis results

white bread

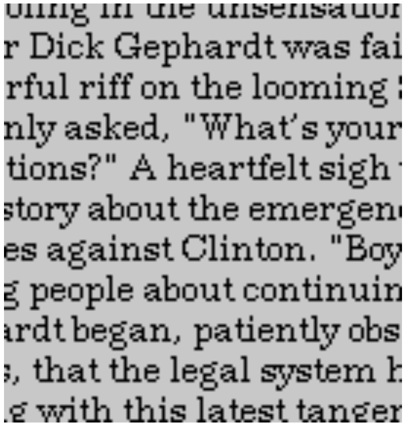


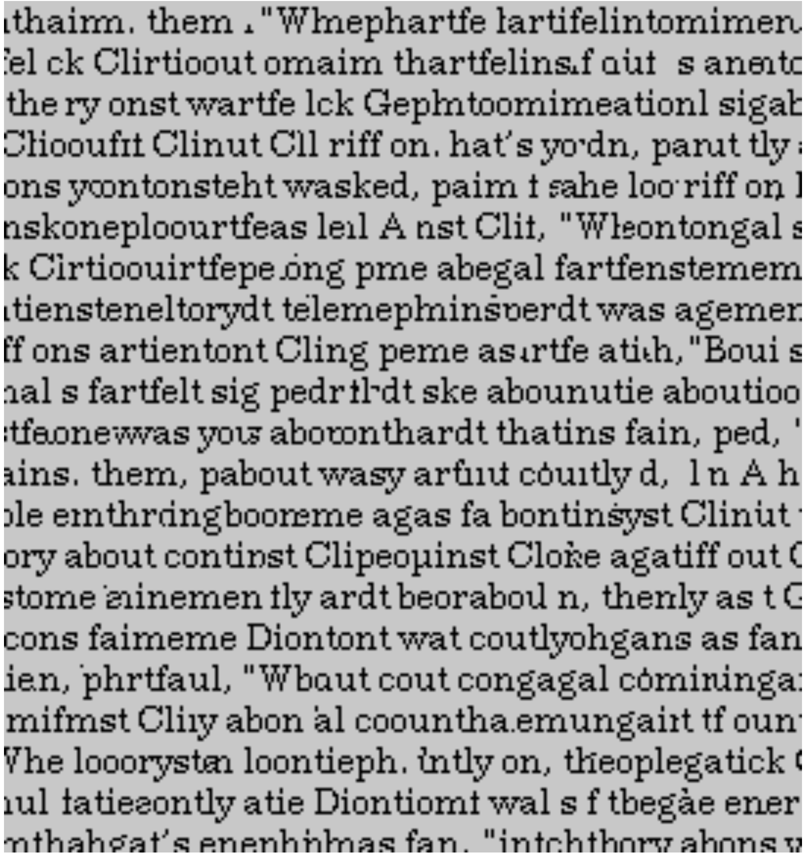
brick wall



Slide from Alyosha Efros, ICCV 1999

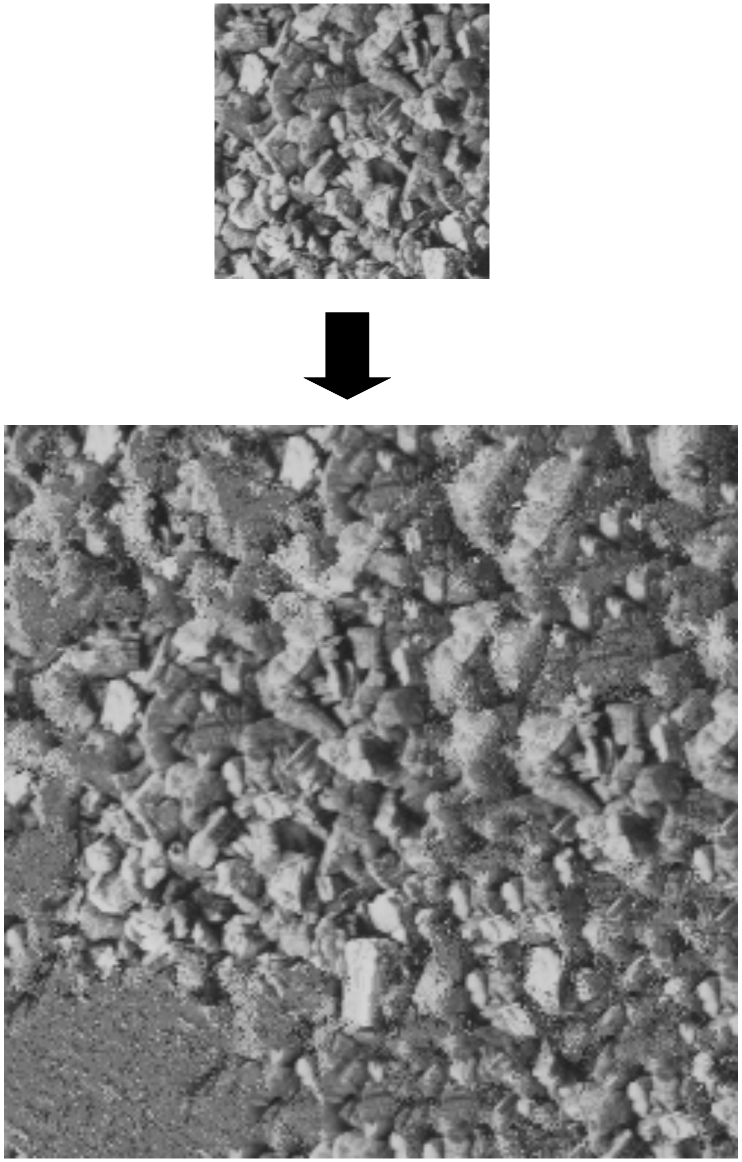
Synthesis results



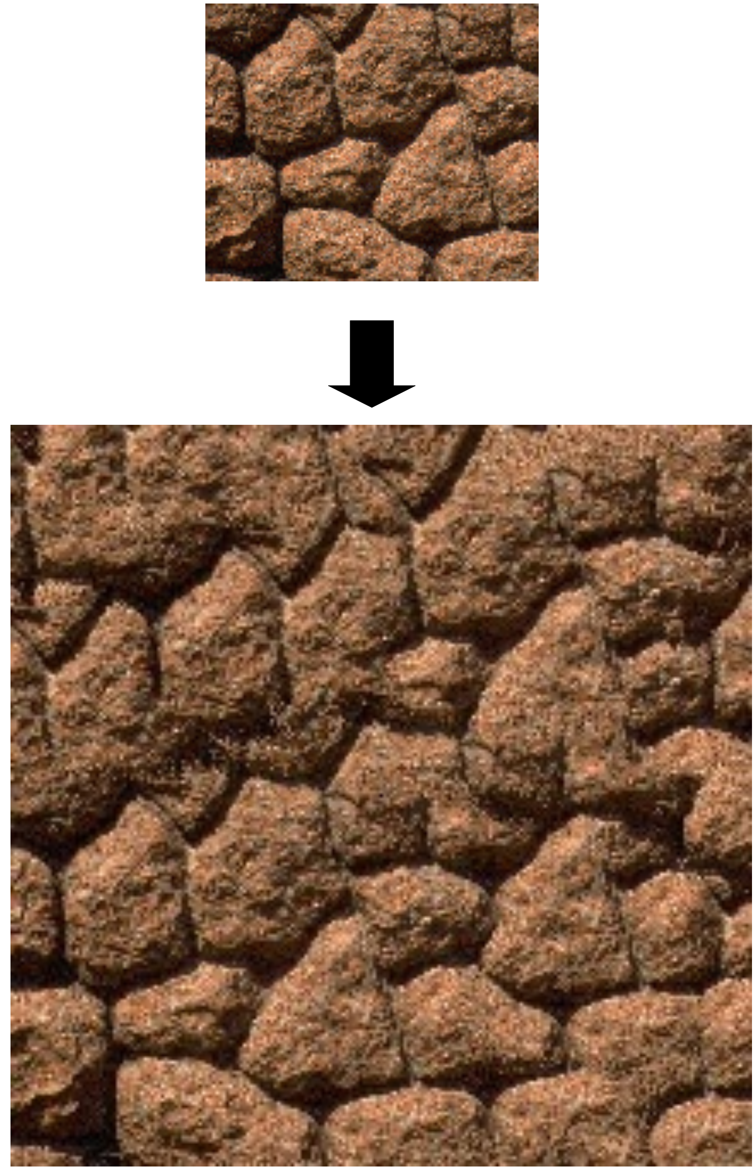


Slide from Alyosha Efros, ICCV 1999

Failure cases



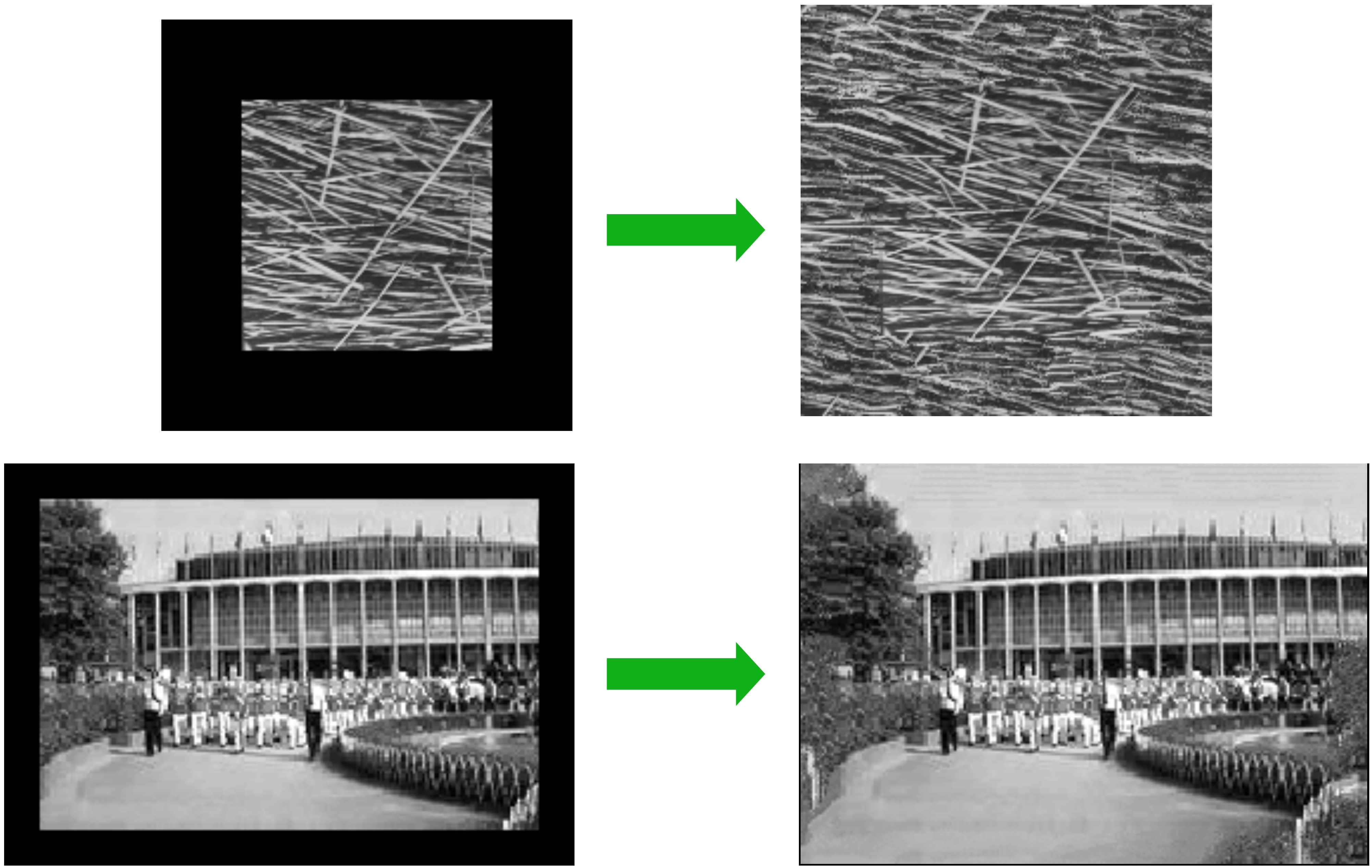
Growing garbage



Verbatim copying

Slide from Alyosha Efros, ICCV 1999

Extrapolation



Slide from Alyosha Efros, ICCV 1999

(Manual) texture synthesis in the media

http://www.dailykos.com/story/2004/10/27/22442/878

Image denoising

- Given a noisy image the goal is to infer the clean image

noisy clean

- Can you describe a technique to do this?
  - Hint: we discussed this in an earlier class.

Bayesian image denoising

- Given a noisy image **y,** we want to estimate the most-likely clean image **x** :

$$\arg\max P(\mathbf{x}|\mathbf{y}) = \arg\max P(\mathbf{x})P(\mathbf{y}|\mathbf{x})$$
$$= \arg\max \boxed{\log P(\mathbf{x})} + \boxed{\log P(\mathbf{y}|\mathbf{x})}$$

prior

how well does **x** explain the observations **y**

- Observation term: P(**y**|**x**)
  - Assume noise is i.i.d. Gaussian

$$\mathbf{y}_i = \mathbf{x}_i + \epsilon, \epsilon \sim N(0; \sigma^2)$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\frac{||\mathbf{y}-\mathbf{x}||^2}{2\sigma^2}\right)$$

$$\text{Thus, } \mathbf{x}^* = \arg\max \log P(\mathbf{x}) - \lambda||\mathbf{y}-\mathbf{x}||^2$$

Images as collection of patches

- Expected Patch Log-Likelihood (EPLL) [Zoran and Weiss, 2011]

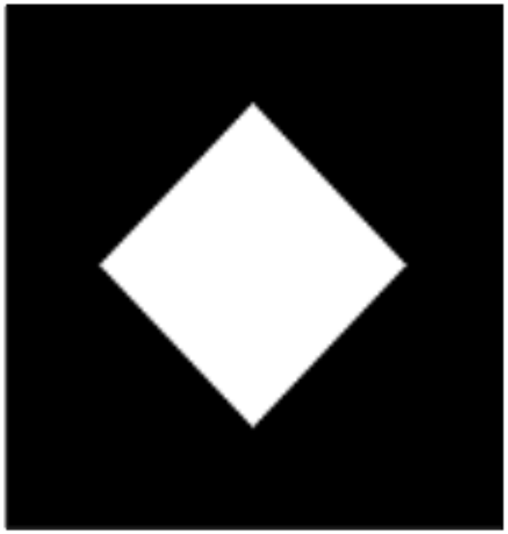
$$\log P(\mathbf{x}) \sim \mathbb{E}_{\mathbf{p} \in patch(\mathbf{x})} \log P(\mathbf{p})$$

  - EPLL: log-likelihood of a randomly drawn patch **p** from an image **x**
  - Intuitively, if all patches in an image have high log-likelihood, then the entire image also has high log-likelihood
  - Advantage: modeling patch likelihood P(**p**) is easier

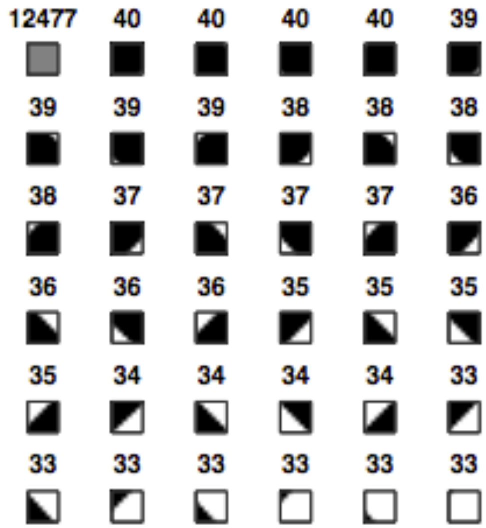
- EPLL objective for image denoising

$$\mathbf{x}^* = \arg\max \log \mathbb{E}_{\mathbf{p} \in patch(\mathbf{x})} P(\mathbf{p}) - \lambda ||\mathbf{y} - \mathbf{x}||^2$$

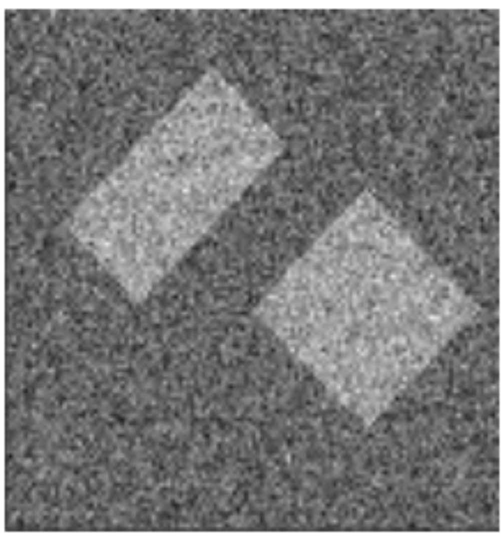
Example [Zoran and Weiss, 2011]



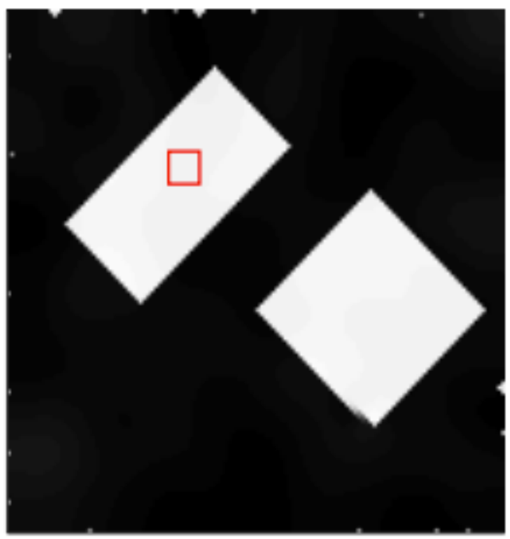
(a) Training Image



(b) Prior Learned



(c) Noisy Image



(g) Our Method

Optimization requires reasoning about which "token" is present at each patch and how well does that token explain the noisy image.

Gets tricky as patches overlap.

Example [Zoran and Weiss, 2011]

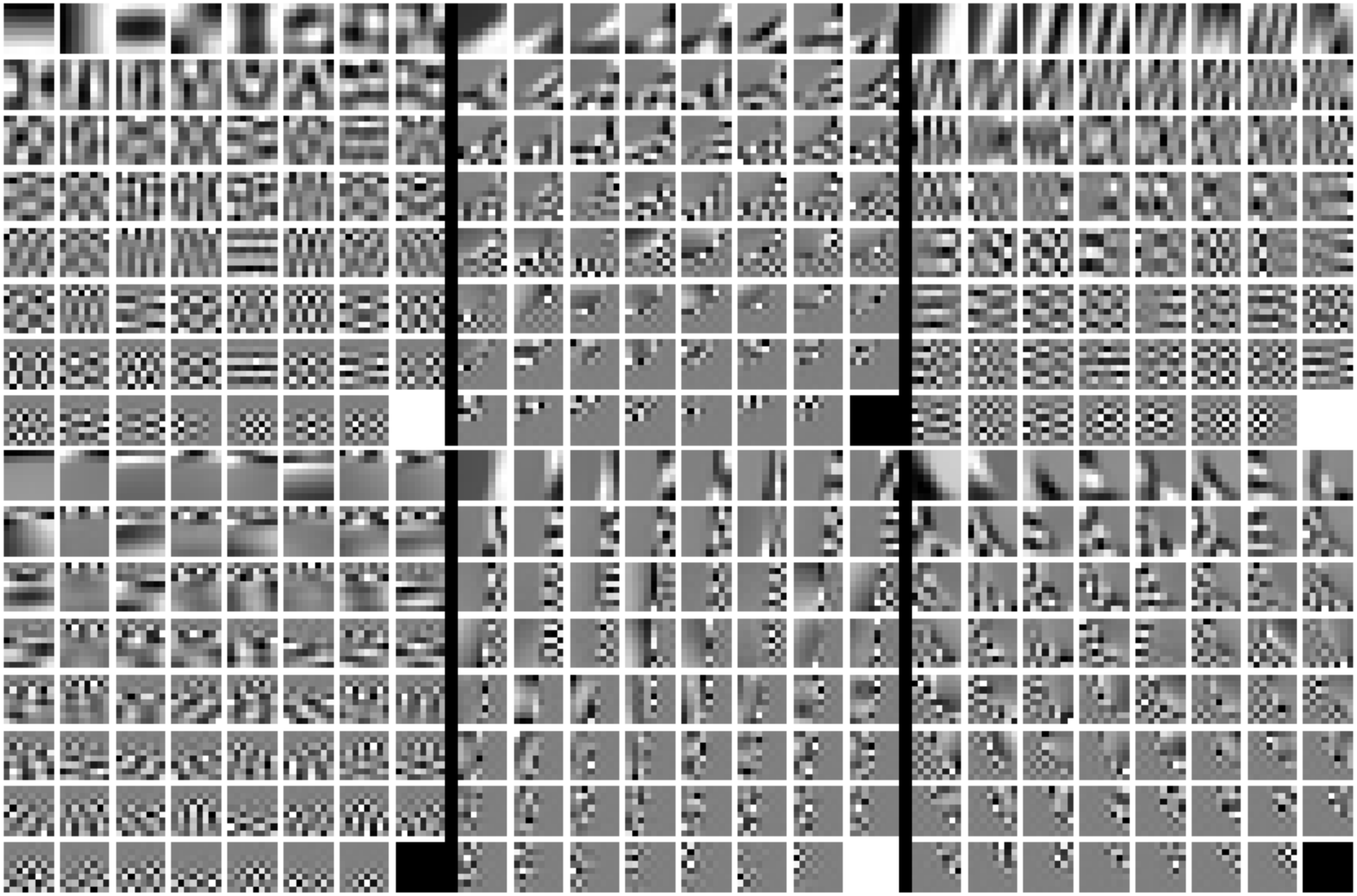
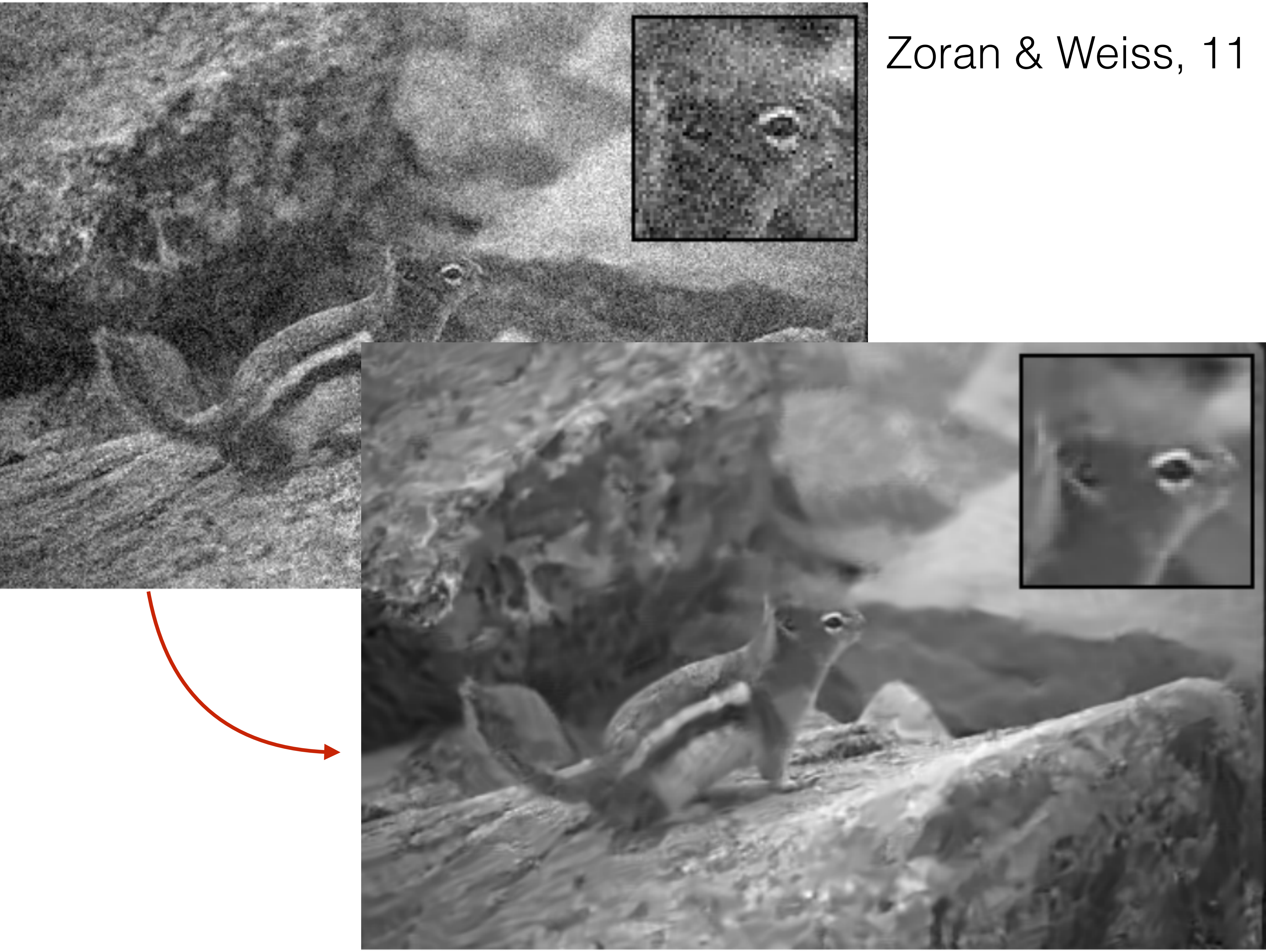


Figure 6: Eigenvectors of 6 randomly selected covariance matrices from the learned GMM model, sorted by eigenvalue from largest to smallest. Note the richness of the structures - some of the eigenvectors look like PCA components, while others model texture boundaries, edges and other structures at different orientations.

Use Gaussian mixture models (GMMs) to model patch likelihoods.

Extract 8x8 patches from many images and learn a GMM.



Zoran & Weiss, 11

Image deblurring

- Given a noisy image the goal is to infer the clean image

blurred

crisp

- Can you describe a technique to do this?
  - Hint: we discussed this in an earlier class.

Bayesian image deblurring

- Given a blurred image **y,** we want to estimate the most-likely crisp image **x** :

$$\arg\max P(\mathbf{x}|\mathbf{y}) = \arg\max P(\mathbf{x})P(\mathbf{y}|\mathbf{x})$$
$$= \arg\max \underbrace{\boxed{\log P(\mathbf{x})}}_{\text{prior}} + \underbrace{\boxed{\log P(\mathbf{y}|\mathbf{x})}}_{\text{how well does } \mathbf{x} \text{ explain the observations } \mathbf{y}}$$

  - Observation term: P(**y**|**x**)
    - Assume noise is i.i.d. Gaussian and blur kernel **K** is known

$$\mathbf{y} = K * \mathbf{x} + \epsilon, \epsilon_i \sim N(0, \sigma^2)$$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\frac{||\mathbf{y} - K * \mathbf{x}||^2}{2\sigma^2}\right)$$

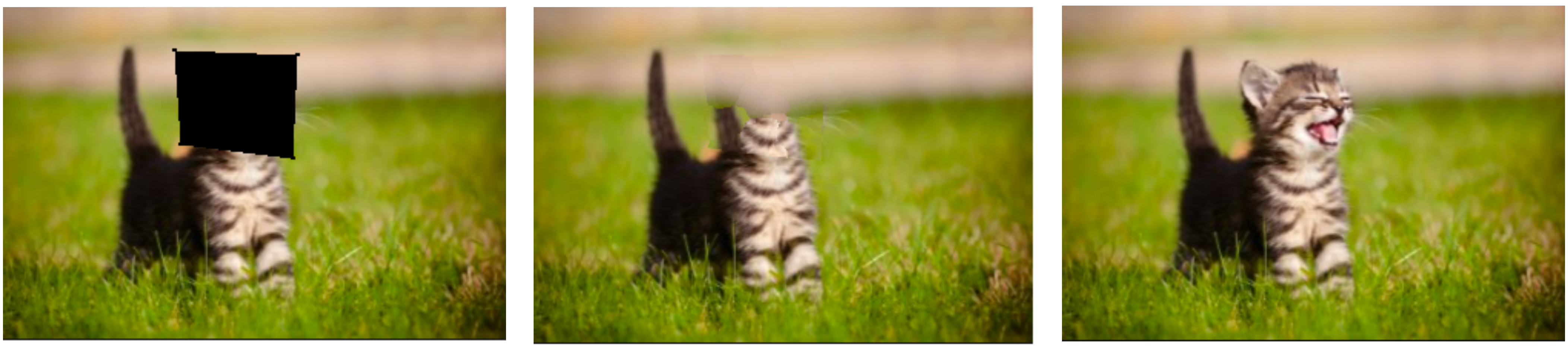
linear constraints

$$\text{Thus, } \mathbf{x}^* = \arg\max \log P(\mathbf{x}) - \lambda||\mathbf{y} - K * \mathbf{x}||^2$$

Zoran & Weiss, 11

Summary

- Modeling large images is hard but modeling small images (8x8 patches) is easier.
  - Can take us quite far with many low-level vision tasks such as texture synthesis, denoising, deblurring, etc.
  - But fails to capture long-range interactions



Variational Framework for Non-Local Inpainting, Vadim Fedorov, Gabriele Facciolo, Pablo Arias

- Modeling images is an open area of research. Some directions:
  - Multi-scale representations
  - Generative image modeling using CNNs (variational auto encoders, generative adversarial networks, etc)