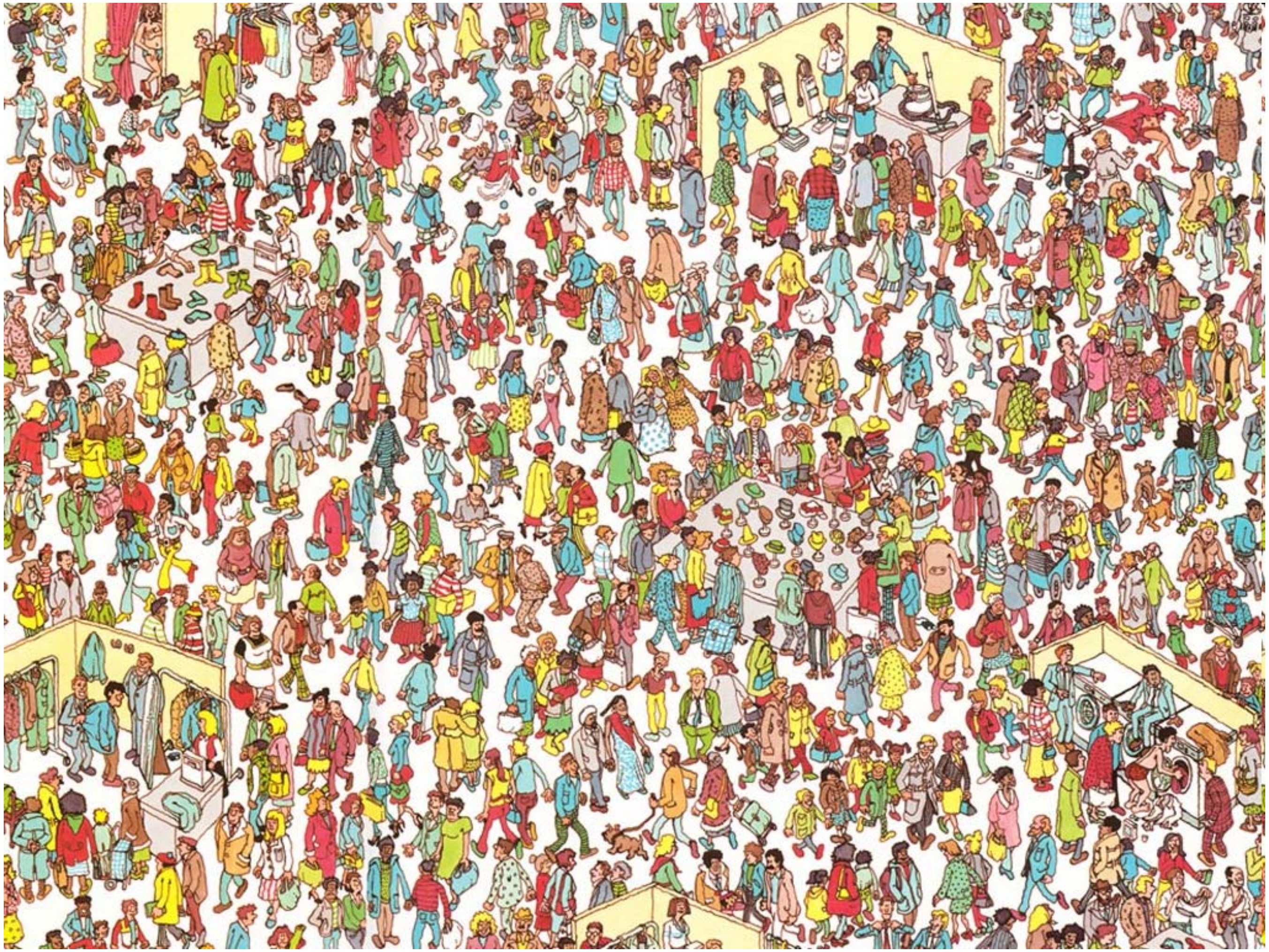
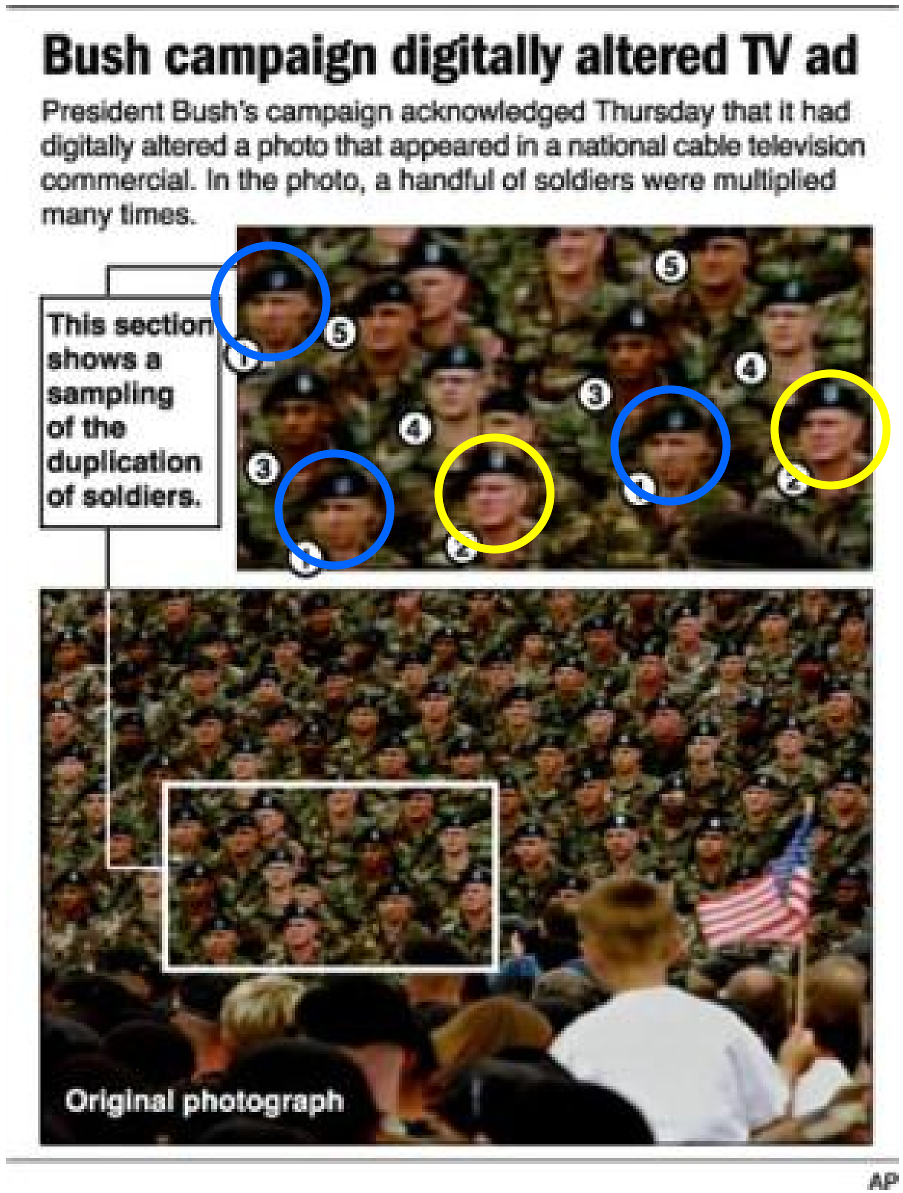
Lecture 12

Texture



http://whereswaldo.com/

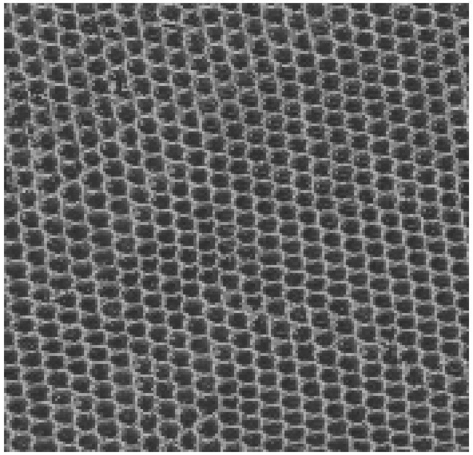
Texture in the News!

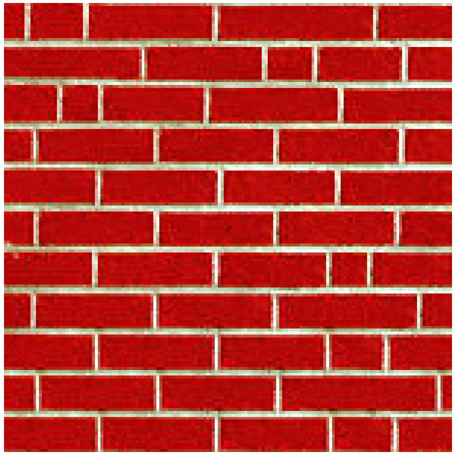


Bush campaign digitally altered TV ad

President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

What is Texture?





- An image obeying some statistical properties
- Similar structures repeated over and over again
- Often has some degree of randomness

Some slides adapted from Alyosha Efros, Steve Seitz, Linda Shapiro

Understanding Texture

Texture Analysis

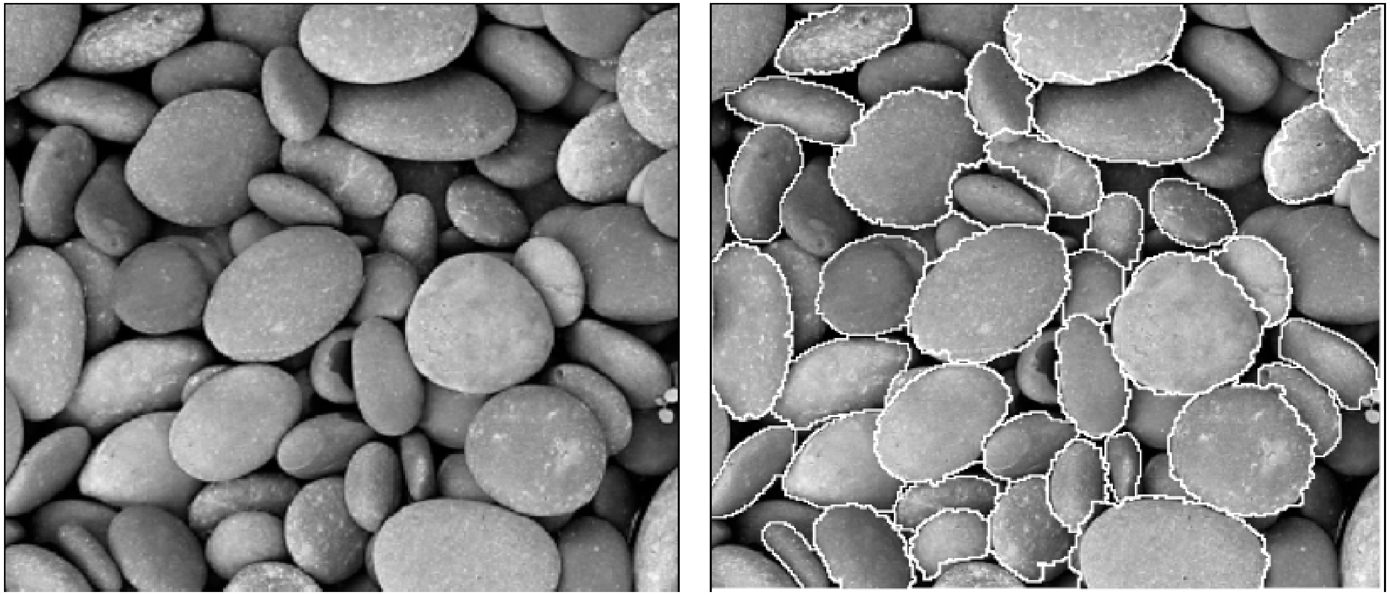
- Structural approach
- Statistical approach
- Fourier approach

Texture Synthesis

- Sampling using Markov random fields
- Graph cut textures
- Image analogies

Structural approach to describing texture

A texture is a set of texture elements or ***texels*** occurring in some regular or repeated pattern



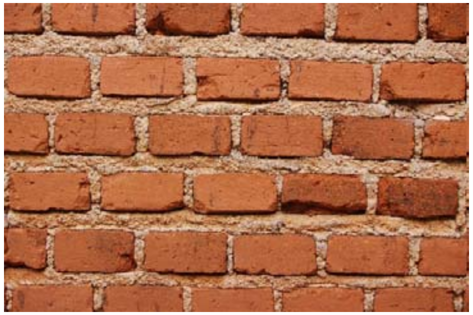
http://vision.ai.uiuc.edu/~sintod/

Aspects of texture

- Size/Granularity (sand versus pebbles versus boulders)
- Directionality/Orientation
- Random or regular (stucco versus bricks)







Problem with Structural Approach

What/Where are the texels?

Extracting texels in real images may be difficult or impossible

Statistical Approach to Texture

- Characterize texture using statistical measures computed from grayscale intensities (or colors) alone

- Less intuitive, but applicable to all images and computationally efficient

- Can be used for both classification of a given input texture and segmentation of an image into different textured regions

Some Simple Statistical Texture Measures

Edge Density and Direction

- Use an edge detector as the first step in texture analysis.
- The number of edge pixels in a fixed-size region tells us how busy that region is
- The directions of the edges also help characterize the texture

Two Edge-based Texture Measures

1. Edgeness per unit area: Given pixels p in a region:

$$\boxed{\mathbf{F_{edgeness} = |\{ p \mid gradient\_magnitude(p) \geq threshold\}| / N}}$$

where N is the size of the image region being analyzed

2. Histograms of edge magnitude and direction for a region R:

$$\boxed{\mathbf{F_{magdir} = (H_{magnitude}(R), H_{direction}(R))}}$$

where these are the normalized histograms of gradient magnitudes and gradient directions, respectively.

Original Image | Edge Image | Thresholded Edge Image

Different $F_{edgeness}$ for different regions

Texture Energy Features

- Use texture filters applied to the image to create filtered images from which texture features are computed.

- Laws' Technique (Laws, 1980):
  - Filter the input image using texture filters.
  - Compute texture energy by summing the absolute value of filtering results in local neighborhoods around each pixel.
  - Combine features to achieve rotational invariance.

Law's texture masks

L5	(Level)	=	[1 4 6 4 1]
E5	(Edge)	=	[-1 -2 0 2 1]
S5	(Spot)	=	[-1 0 2 0 -1]
R5	(Ripple)	=	[1 -4 6 -4 1]

- (L5) (Gaussian) gives a center-weighted local average
- (E5) (gradient) responds to row or col step edges
- (S5) (LOG) detects spots
- (R5) (Gabor) detects ripples

Law's texture masks (2D)

Creation of 2D Masks

- **1D Masks are "multiplied" to construct 2D masks:**
  mask E5L5 is the "product" of E5 and L5 –

$$\text{E5}\quad \begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times \underset{\text{L5}}{\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}} = \underset{\text{E5L5}}{\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}}$$

9D feature vector for each pixel

- Subtract mean neighborhood intensity from pixel (to reduce illumination effects)
- Filter the neighborhood with 16 5x5 masks
- Compute energy at each pixel by summing absolute value of filter output across neighborhood around pixel
- Define 9 features as follows (replace each pair with average):

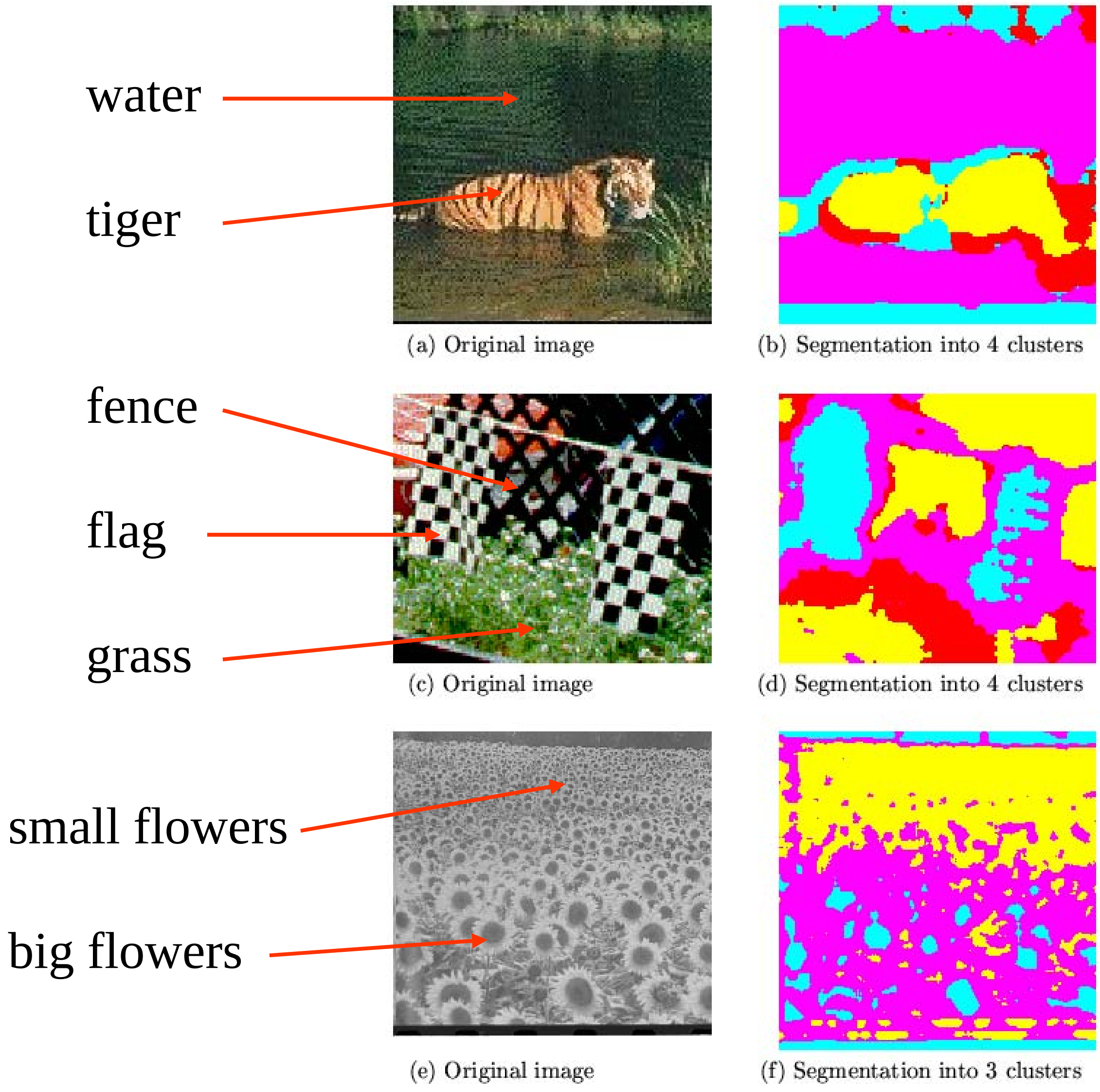
L5E5/E5L5	L5S5/S5L5
L5R5/R5L5	E5E5
E5S5/S5E5	E5R5/R5E5
S5S5	S5R5/R5S5
R5R5	

Texture energy features from sample images

Table 7.2: Laws texture energy measures for major regions of the images of Figure 7.8.

Region	E5E5	S5S5	R5R5	E5L5	S5L5	R5L5	S5E5	R5E5	R5S5
Tiger	168.1	84.0	807.7	553.7	354.4	910.6	116.3	339.2	257.4
Water	68.5	36.9	366.8	218.7	149.3	459.4	49.6	159.1	117.3
Flags	258.1	113.0	787.7	1057.6	702.2	2056.3	182.4	611.5	350.8
Fence	189.5	80.7	624.3	701.7	377.5	803.1	120.6	297.5	215.0
Grass	206.5	103.6	1031.7	625.2	428.3	1153.6	146.0	427.5	323.6
Small flowers	114.9	48.6	289.1	402.6	241.3	484.3	73.6	158.2	109.3
Big flowers	76.7	28.8	177.1	301.5	158.4	270.0	45.6	89.7	62.9
Borders	15.3	6.4	64.4	92.3	36.3	74.5	9.3	26.1	19.5

Using texture energy for segmentation



Using autocorrelation to detect texture

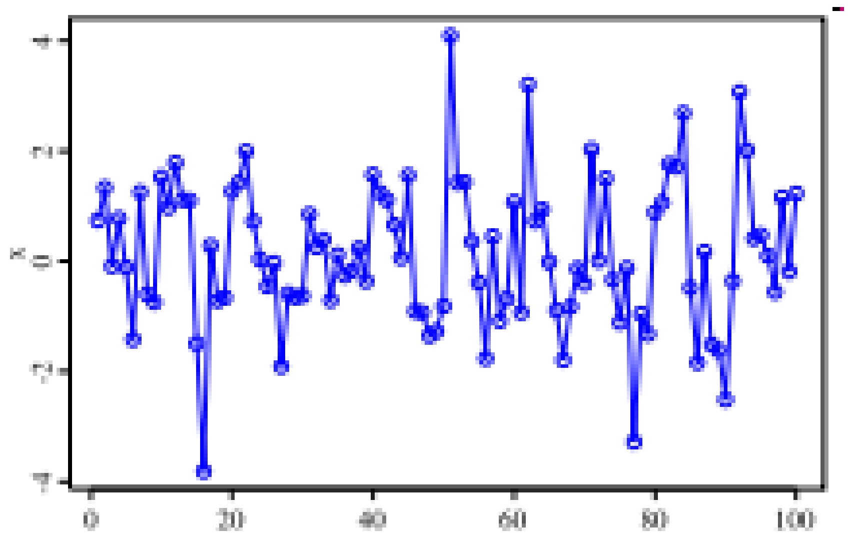
$$\rho(dr, dc) = \frac{\sum_{r=0}^{N} \sum_{c=0}^{N} I[r,c] I(r+dr, c+dc]}{\sum_{r=0}^{N} \sum_{c=0}^{N} I^2[r,c]} = \frac{I[r,c] \circ I_d[r,c]}{I[r,c] \circ I[r,c]}$$

Autocorrelation function computes the dot product (energy) of original image with shifted image for different shifts

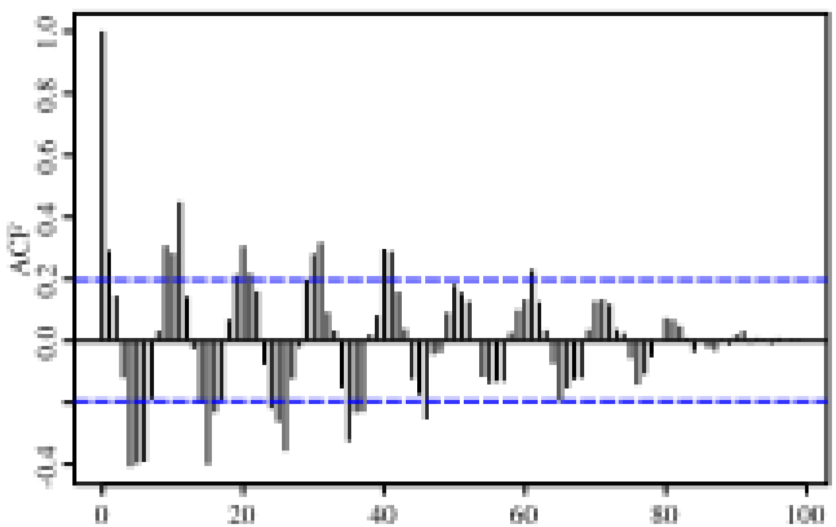
Can detect repetitive patterns of texels

Captures fineness/coarseness of the texture

1D example



Signal = noisy sine wave (wavelength 10 units)



Autocorrelation function showing ***peaks*** (every 10 units starting at 0) and ***valleys*** (every 10 units starting at 5)

(http://en.wikipedia.org/wiki/Autocorrelation)

Interpreting autocorrelation

Regular textures → function will have peaks and valleys

Random textures → only peak at [0, 0]; breadth of peak gives the size of the texture

Coarse texture → function drops off slowly

Fine texture → function drops off rapidly

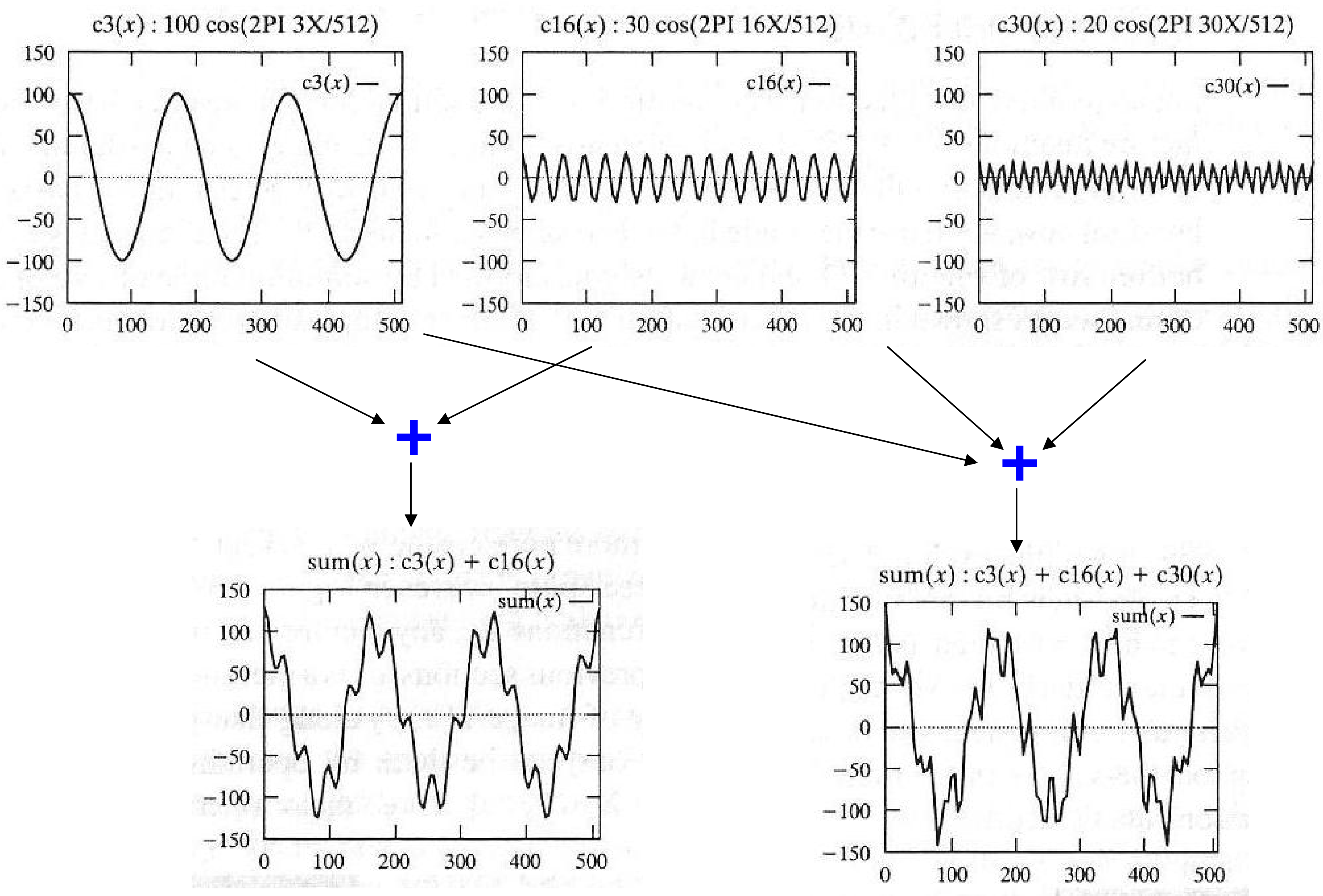
Can drop differently for r and c

Relationship to Fourier Analysis

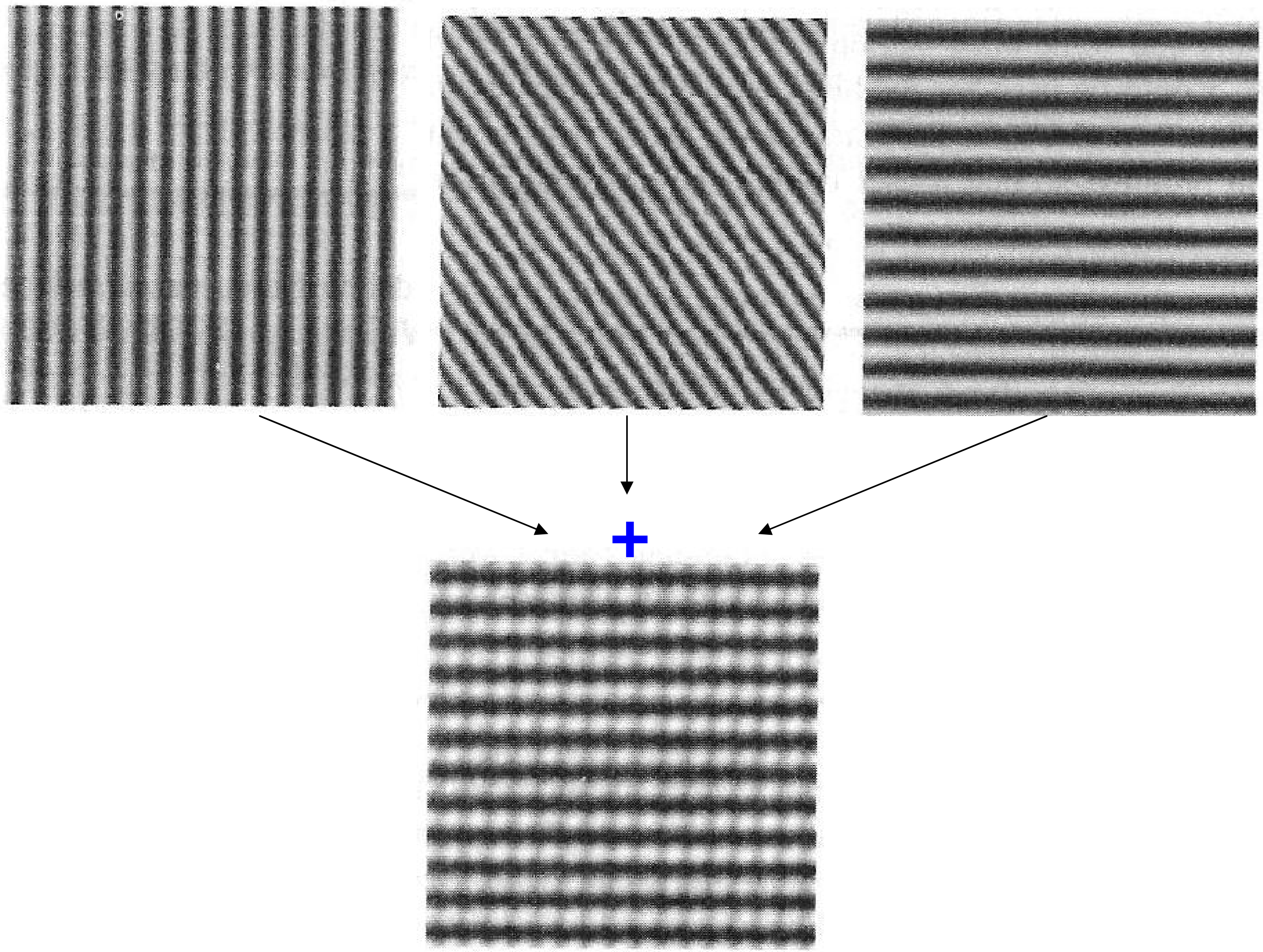
The power spectrum of a signal is the Fourier transform of the autocorrelation function

What is the Fourier transform?

Representing Signals with Sine/Cosine Waves



2D Example



Fourier transform of an image

The 2D Fourier Transform transforms an image f(x,y) into the u,v frequency domain function F:

$$F(u,v) \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)E_{u,v}(x,y)\,dx\,dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j\,2\pi(ux+vy)}\,dx\,dy$$

where $\mathbf{E}_{u,\,v}(\mathbf{x},\mathbf{y}) \equiv e^{-j\,2\pi(ux+vy)}$

$$= cos(2\pi(ux+vy)) - jsin(2\pi(ux+vy))$$

and $j = \sqrt{-1}.$

Discrete Fourier Transform and Inverse

DFT:

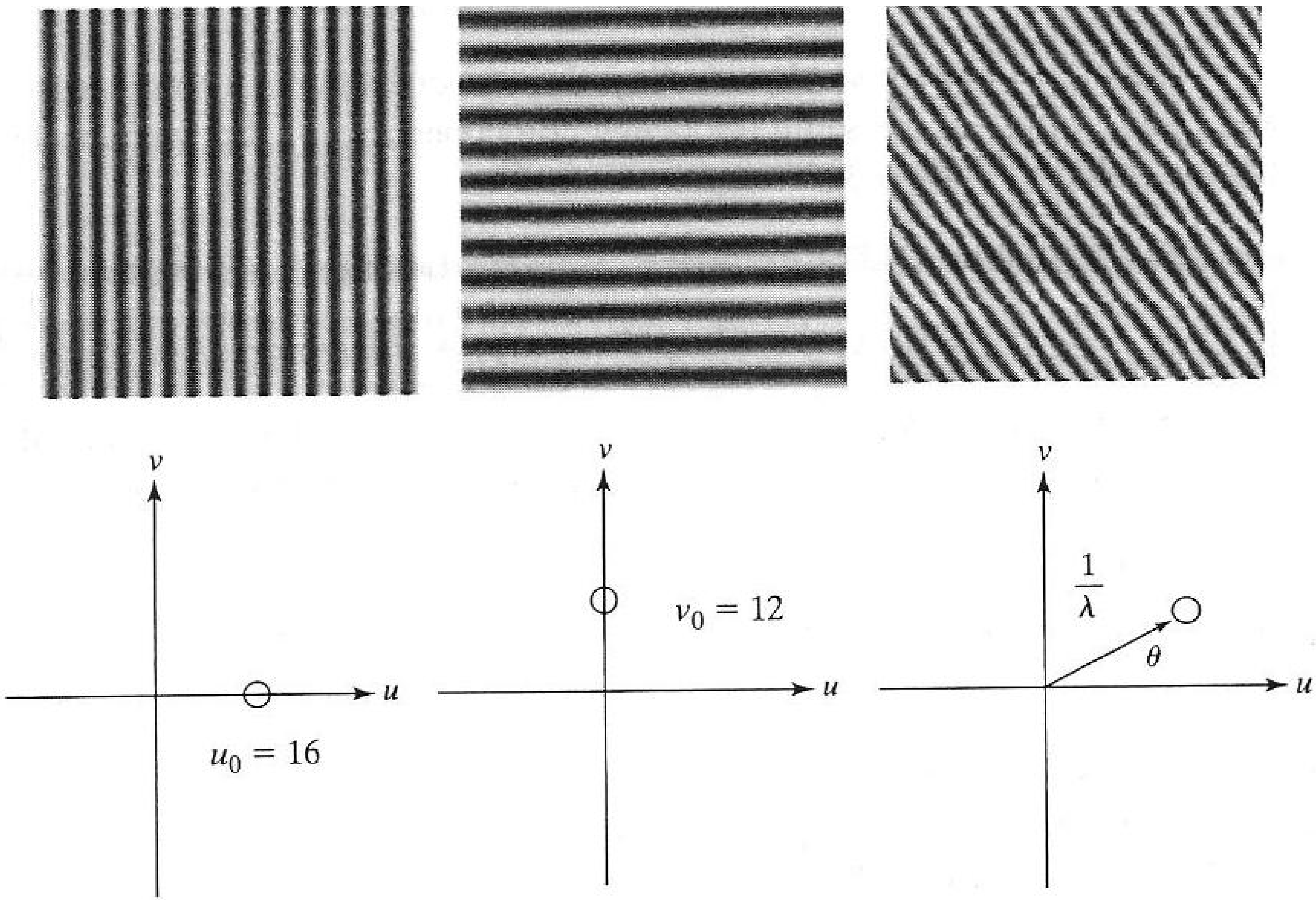
$$F[u,v] \equiv \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I[x,y]e^{\frac{-2\pi j}{N}(xu+yv)}$$

Inverse FT:

$$I[x,y] \equiv \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F[u,v]e^{\frac{+2\pi j}{N}(ux+vy)}$$

Power Spectrum

$$P(u,v) \equiv (Real(F(u,v))^2 + Imaginary(F(u,v))^2\ )$$

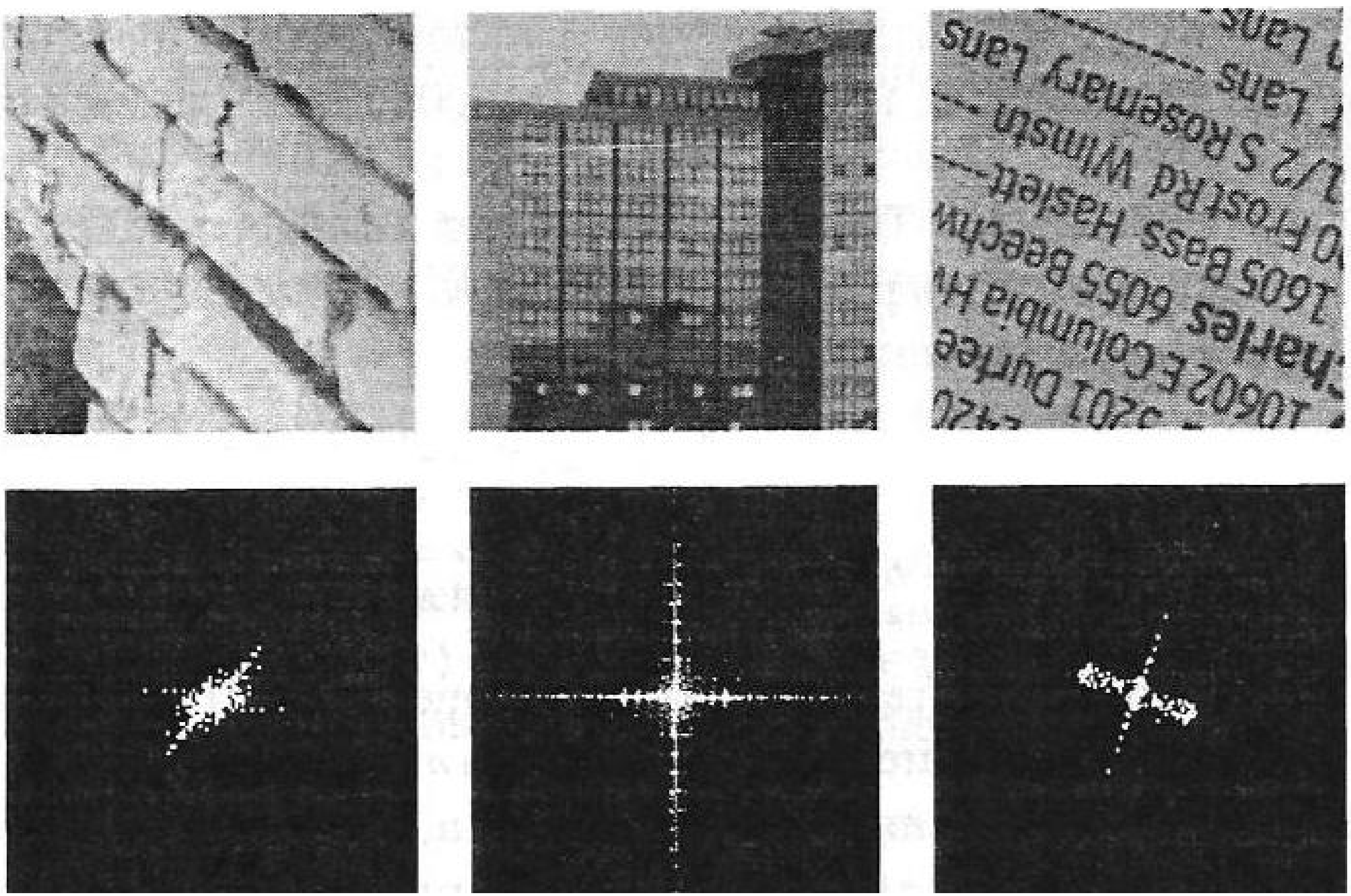


Power spectrum and textures

Concentrated power → regularity

High frequency power → fine texture

Directionality → directional texture



Texture Synthesis

Given a small sample, generate larger realistic versions of the texture

Reading

- Alexei A. Efros and Thomas K. Leung, “Texture Synthesis by Non-parametric Sampling,” Proc. International Conference on Computer Vision (ICCV), 1999.
  - http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.pdf

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

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

- **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:

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

Markov Chain Example: Text

“A dog is a man’s best friend. It’s a dog eat dog world out there.

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

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**					

Markov Chain Example: More Text

A man’s dog is out there. There is a best dog. It’s a man’s world.”

Rows: $\mathbf{x}_{t-1}$; columns: $\mathbf{x}_t$; entries: $p(\mathbf{x}_t|\mathbf{x}_{t-1})$

	a	dog	is	man’s	best	friend	it’s	eat	world	out	there	.
a		**2/6**		**3/6**	**1/6**							
dog			**2/5**					**1/5**	**1/5**			**1/5**
is	**2/3**									**1/3**		
man’s		**1/3**			**1/3**				**1/3**			
best		**1/2**				**1/2**						
friend												**1**
it’s	**1**											
eat		**1**										
world										**1/2**		**1/2**
out											**1**	
there			**1/3**									**2/3**
.	**1/4**						**2/4**				**1/4**	

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

Class-generated example on board...

[Scientific American, June 1989, Dewdney]

"I Spent an Interesting Evening Recently with a Grain of Salt"

- Mark V. Shaney

(computer-generated contributor to UseNet News group called net.singles)

You can try it online here: http://www.yisongyue.com/shaney/

Output of 2nd order word-level Markov Chain after training on 90,000 word philosophical essay:

"Perhaps only the allegory of simulation is unendurable--more cruel than Artaud's Theatre of Cruelty, which was the first to practice deterrence, abstraction, disconnection, deterritorialisation, etc.; and if it were our own past. We are witnessing the end of the negative form. But nothing separates one pole from the very swing of voting "rights" to electoral..."

Markov Random Field

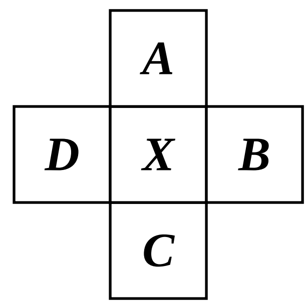
A Markov random field (MRF)

- generalization of Markov chains to two or more dimensions.

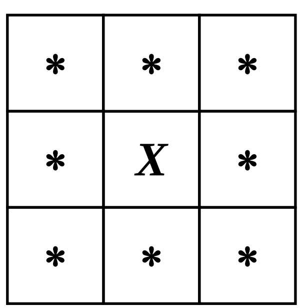
First-order MRF:

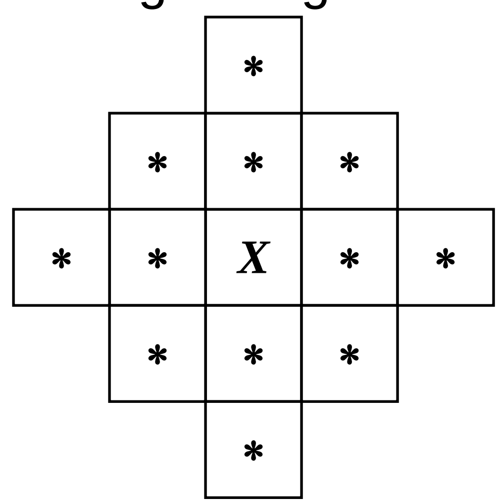
- probability that pixel *X* takes a certain value given the values of neighbors *A*, *B*, *C*, and *D*:

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



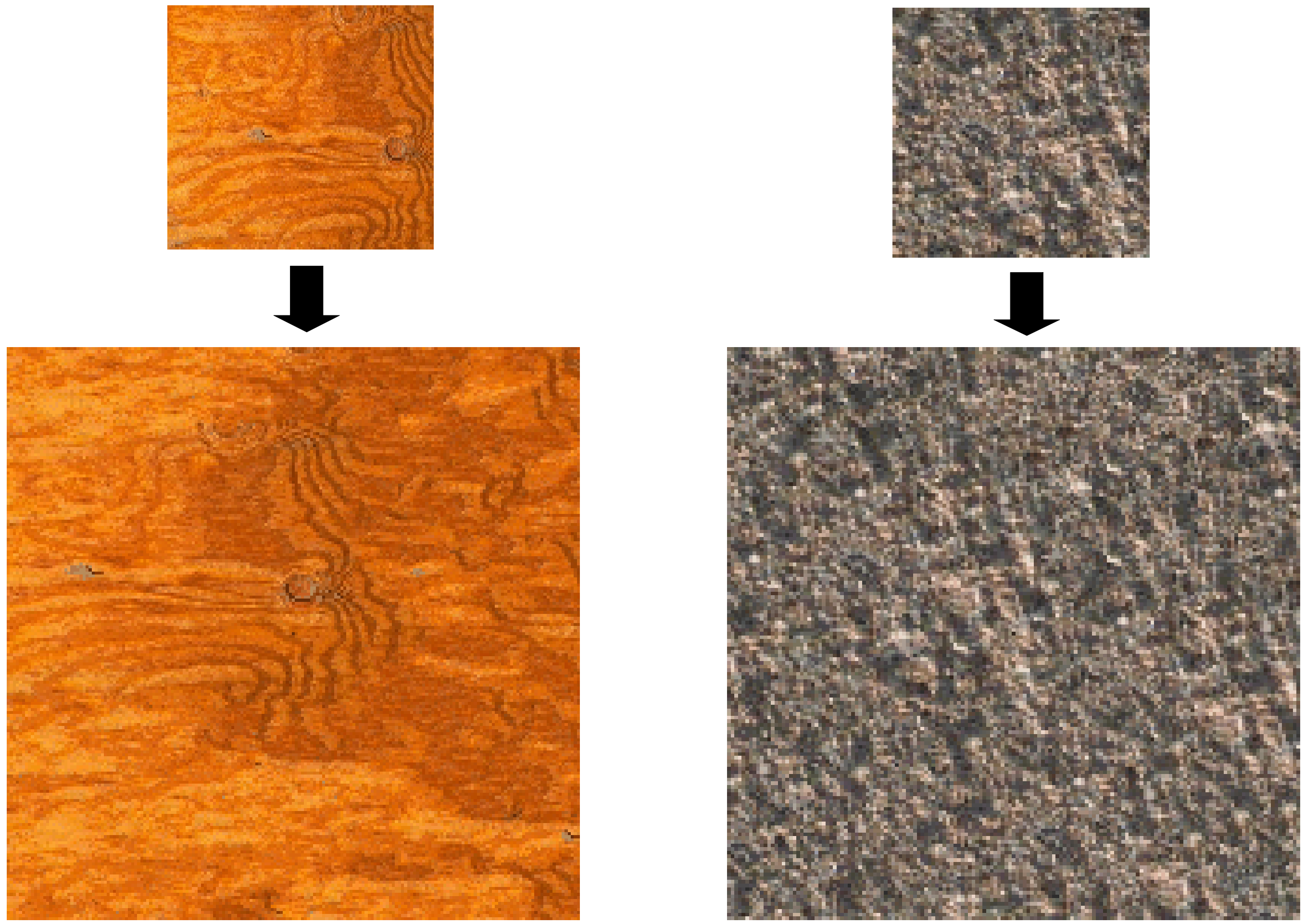
- Higher order MRF's have larger neighborhoods



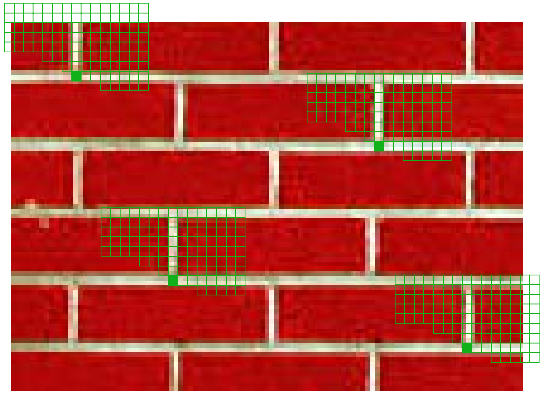


Texture Synthesis [Efros & Leung, ICCV 99]

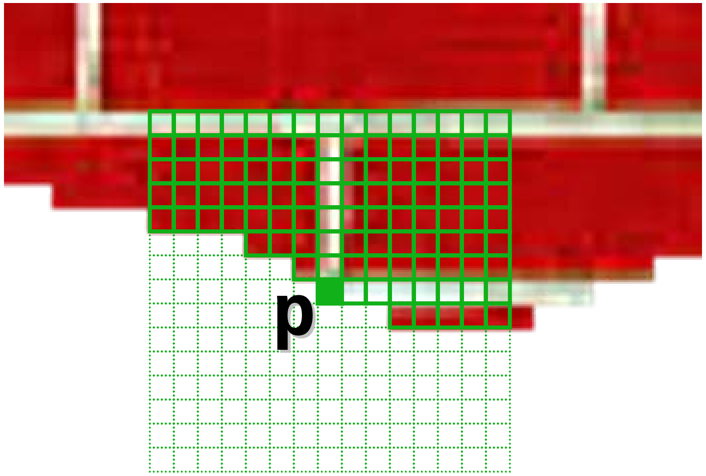
Can apply 2D version of text synthesis



Synthesizing One Pixel



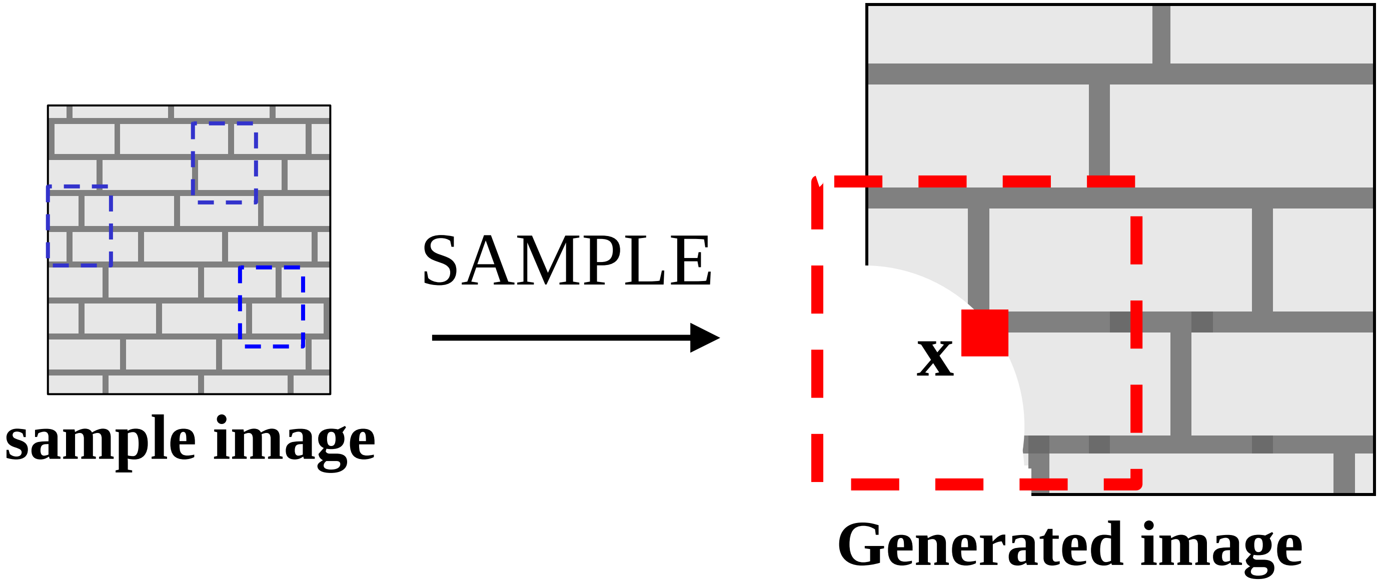
input image



synthesized image

- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
  - consider only pixels in the neighbourhood that are already filled in
- To synthesize **x**
  - pick one matching window at random
  - assign **x** to be the center pixel of that window

Really Synthesizing One Pixel

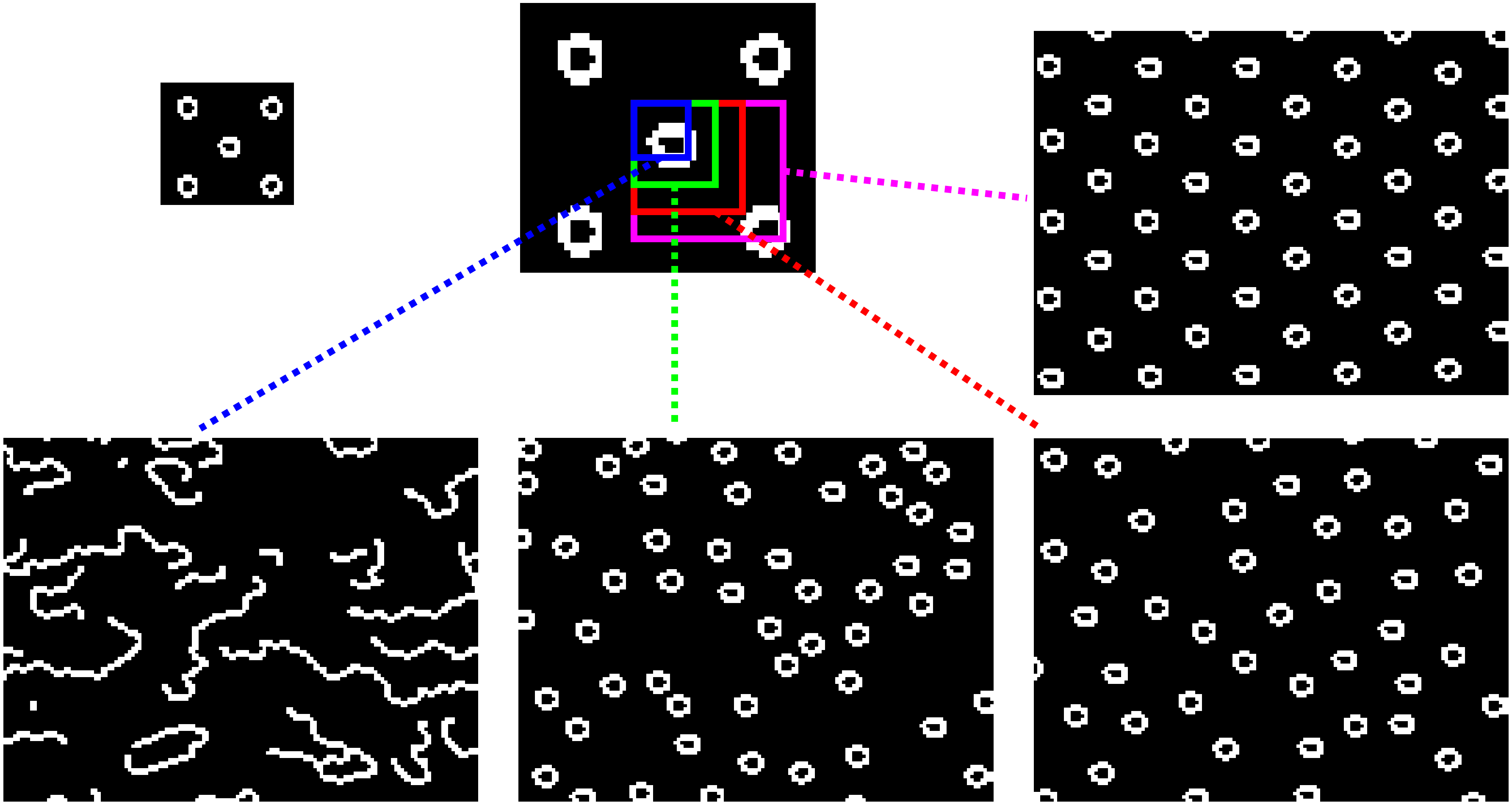


- An exact neighbourhood match might not be present
- So we find the **best** matches using SSD error and randomly choose between them, preferring better matches with higher probability

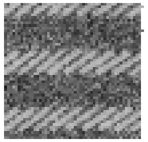
Growing Texture

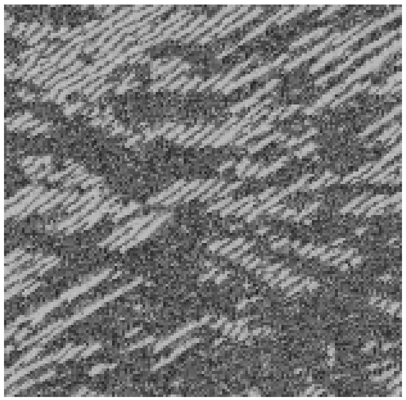
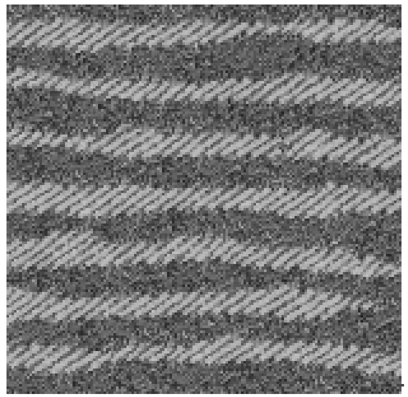
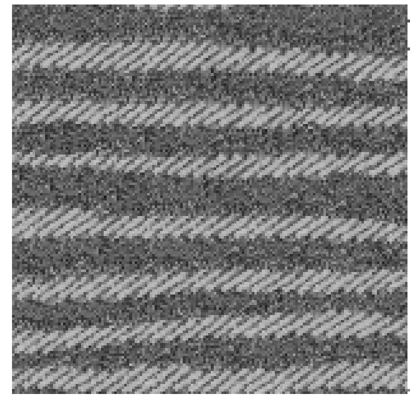
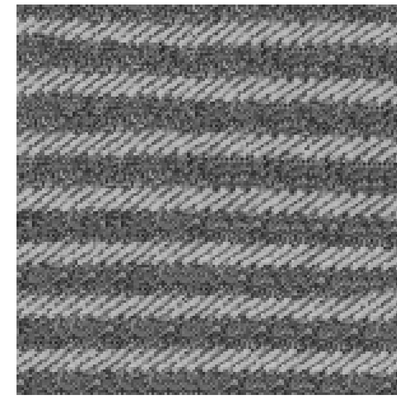
Starting from the initial image, “grow” the texture one pixel at a time

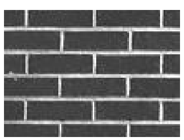
Window Size Controls Regularity



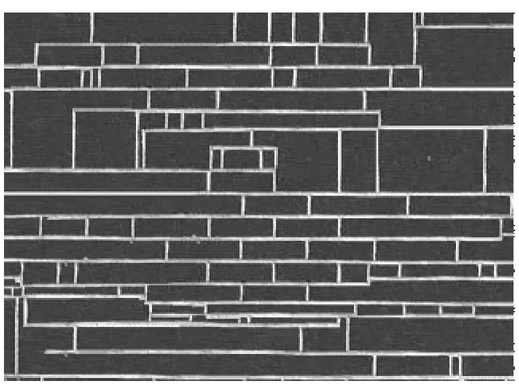
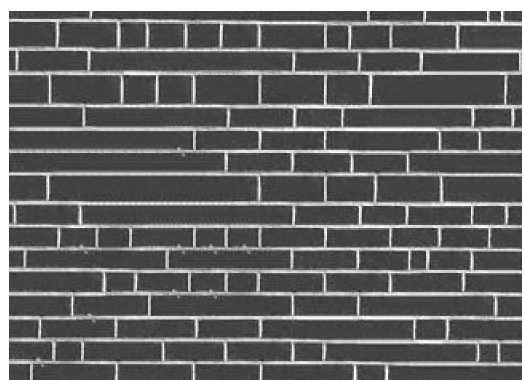
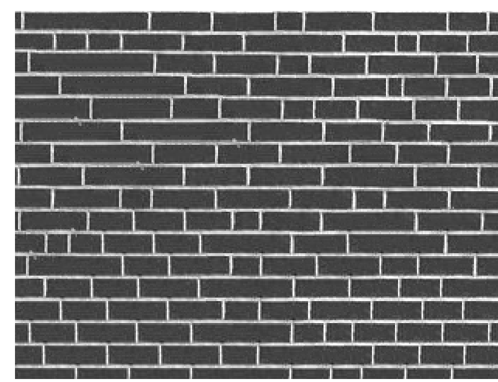
More Synthesis Results



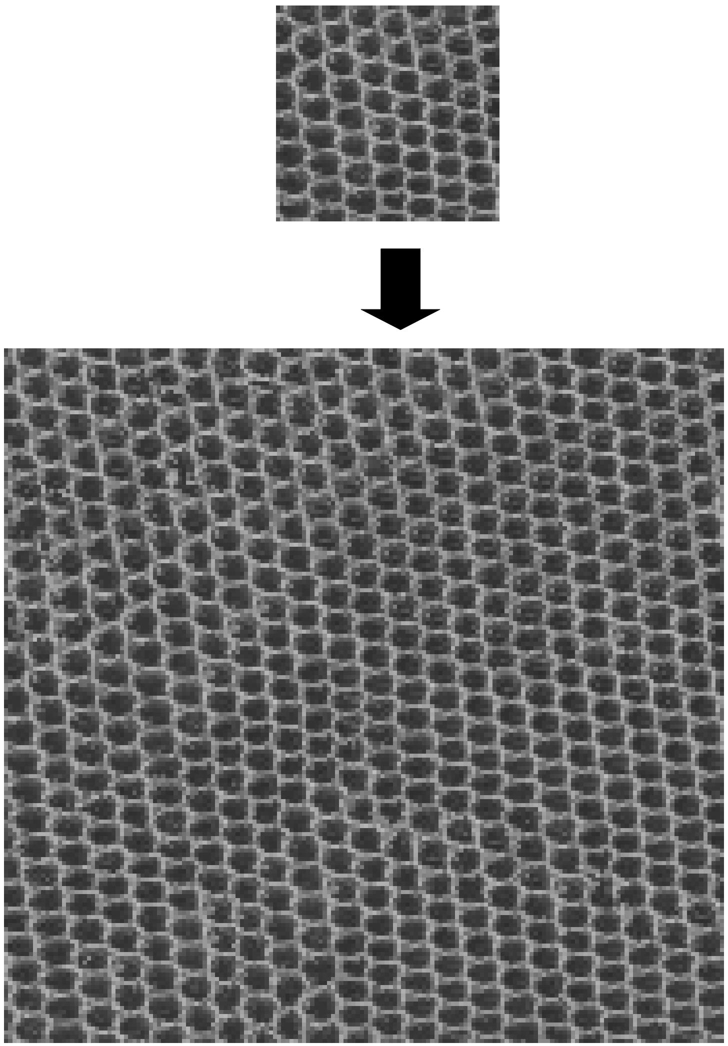


Increasing window size →

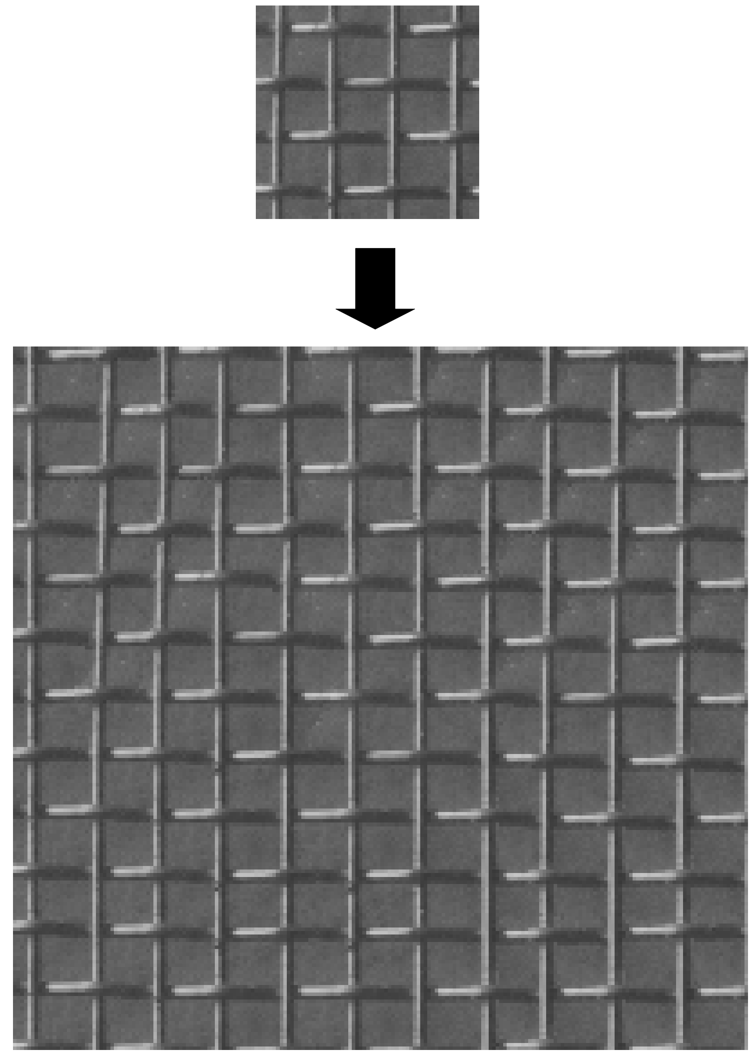
  

More Results

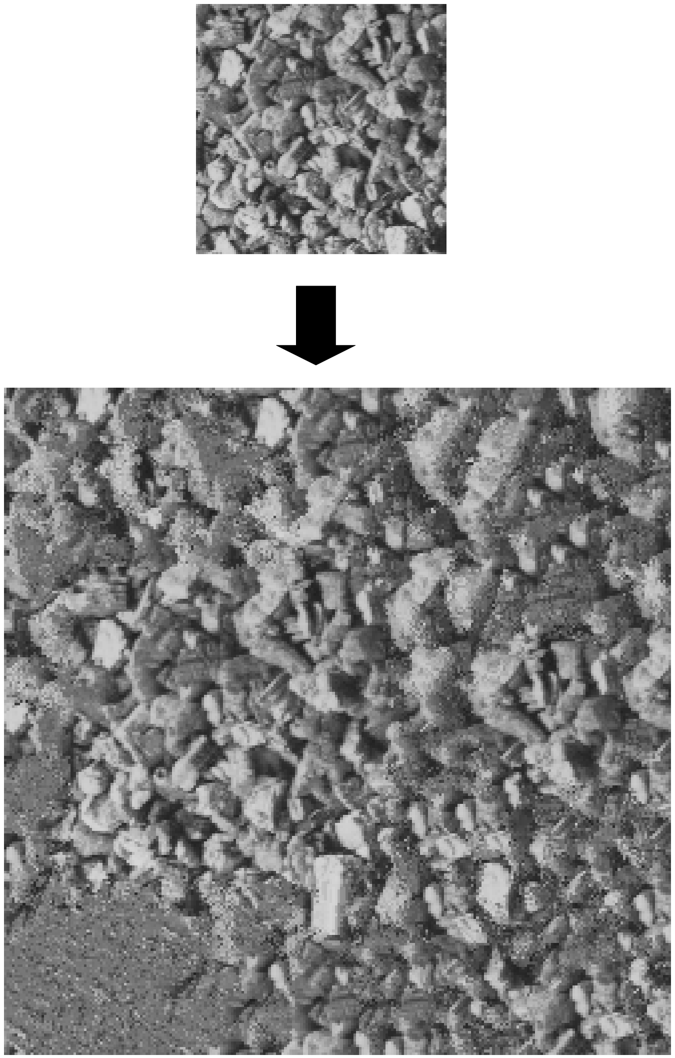
reptile skin



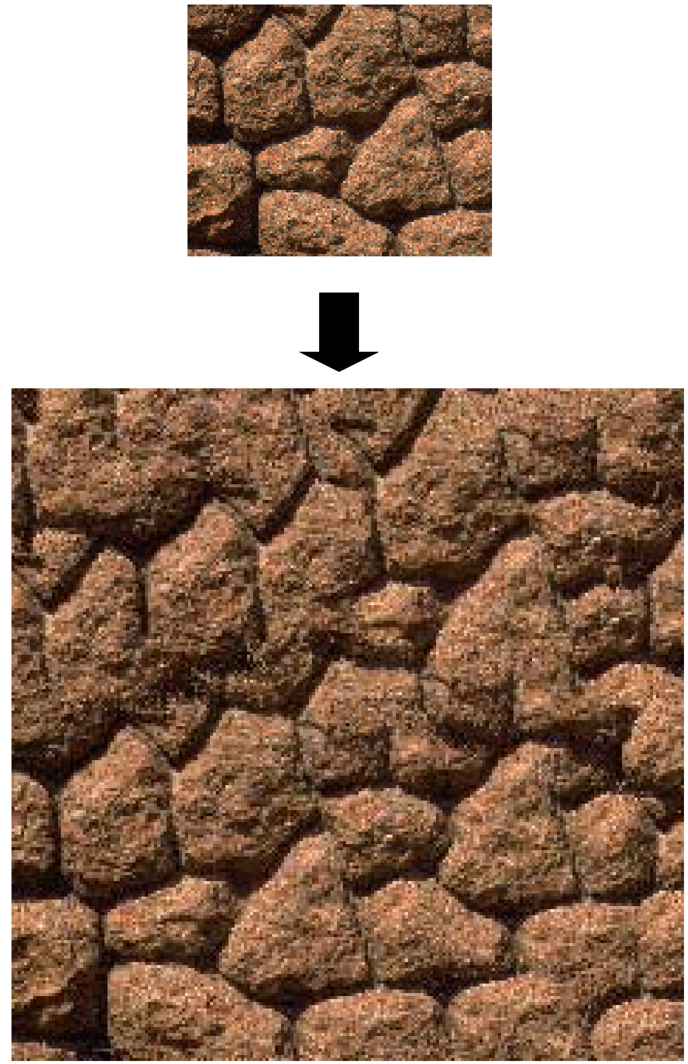
aluminum wire



Failure Cases

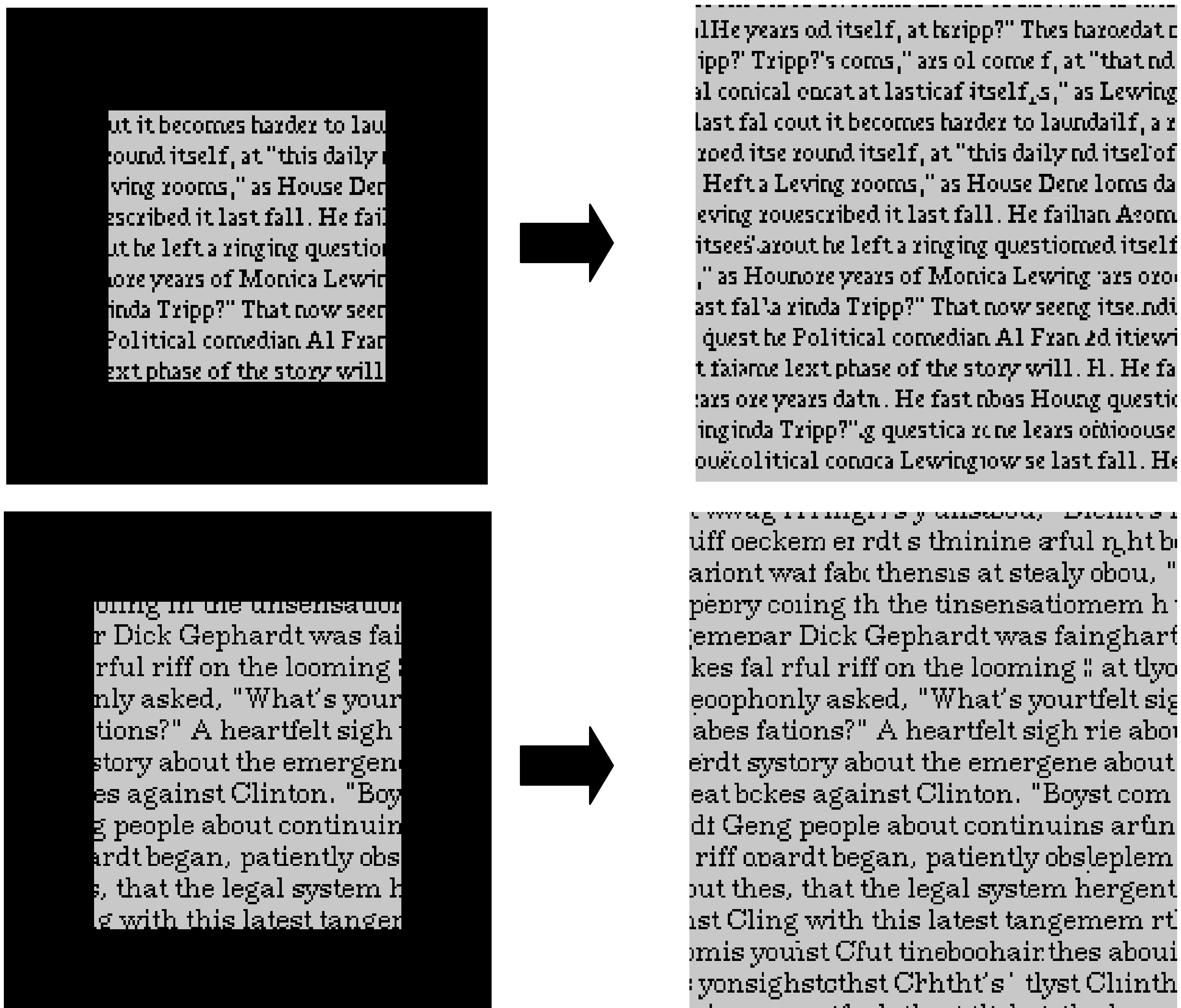


Growing garbage

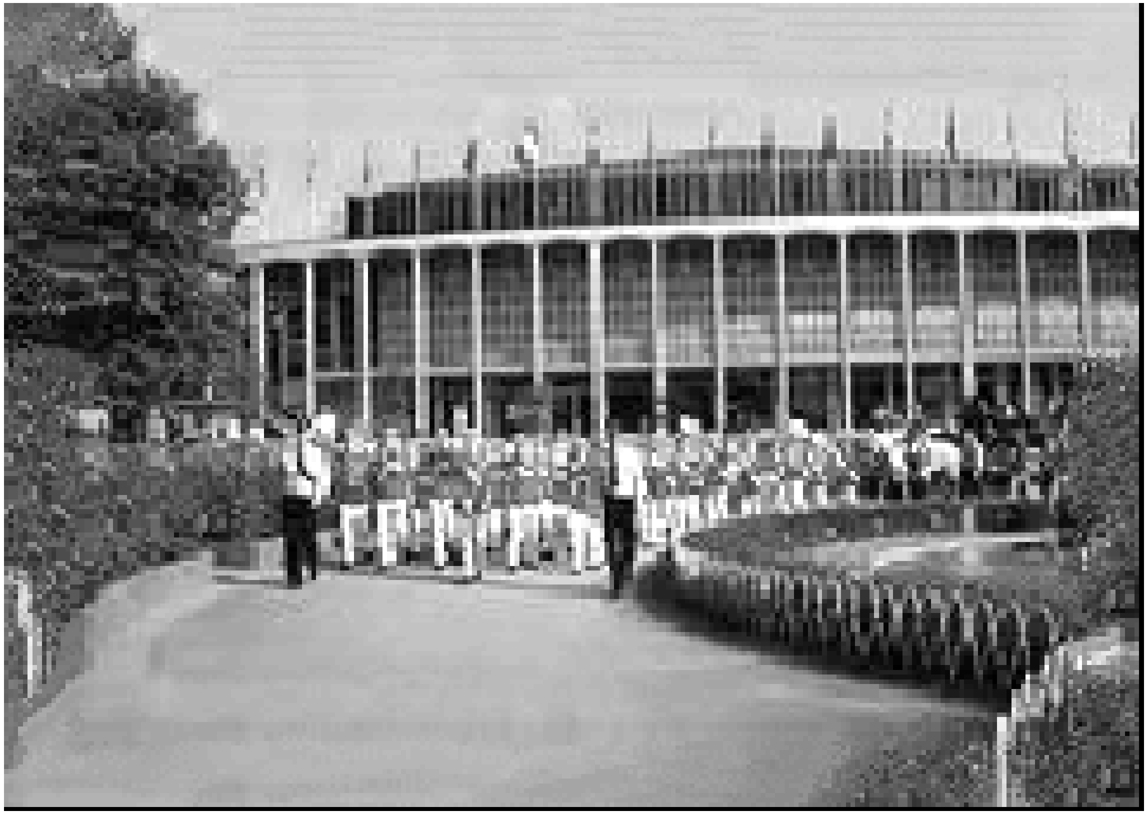


Verbatim copying

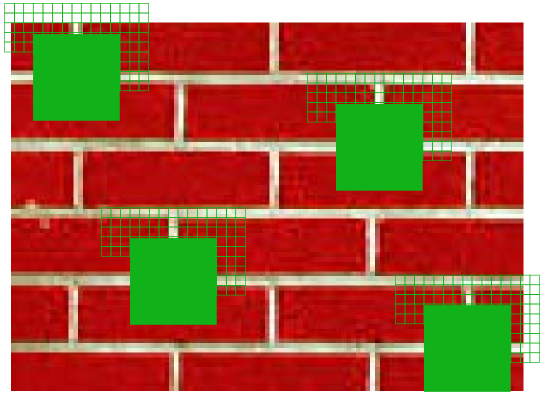
Image-Based Text Synthesis



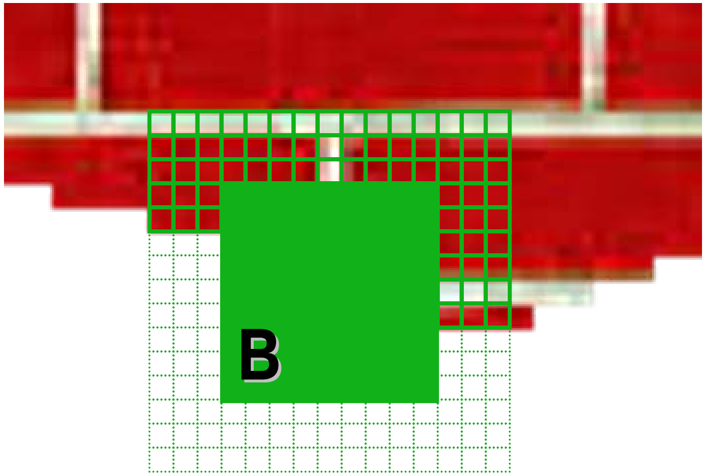
Extrapolation



Block-based texture synthesis



Input image



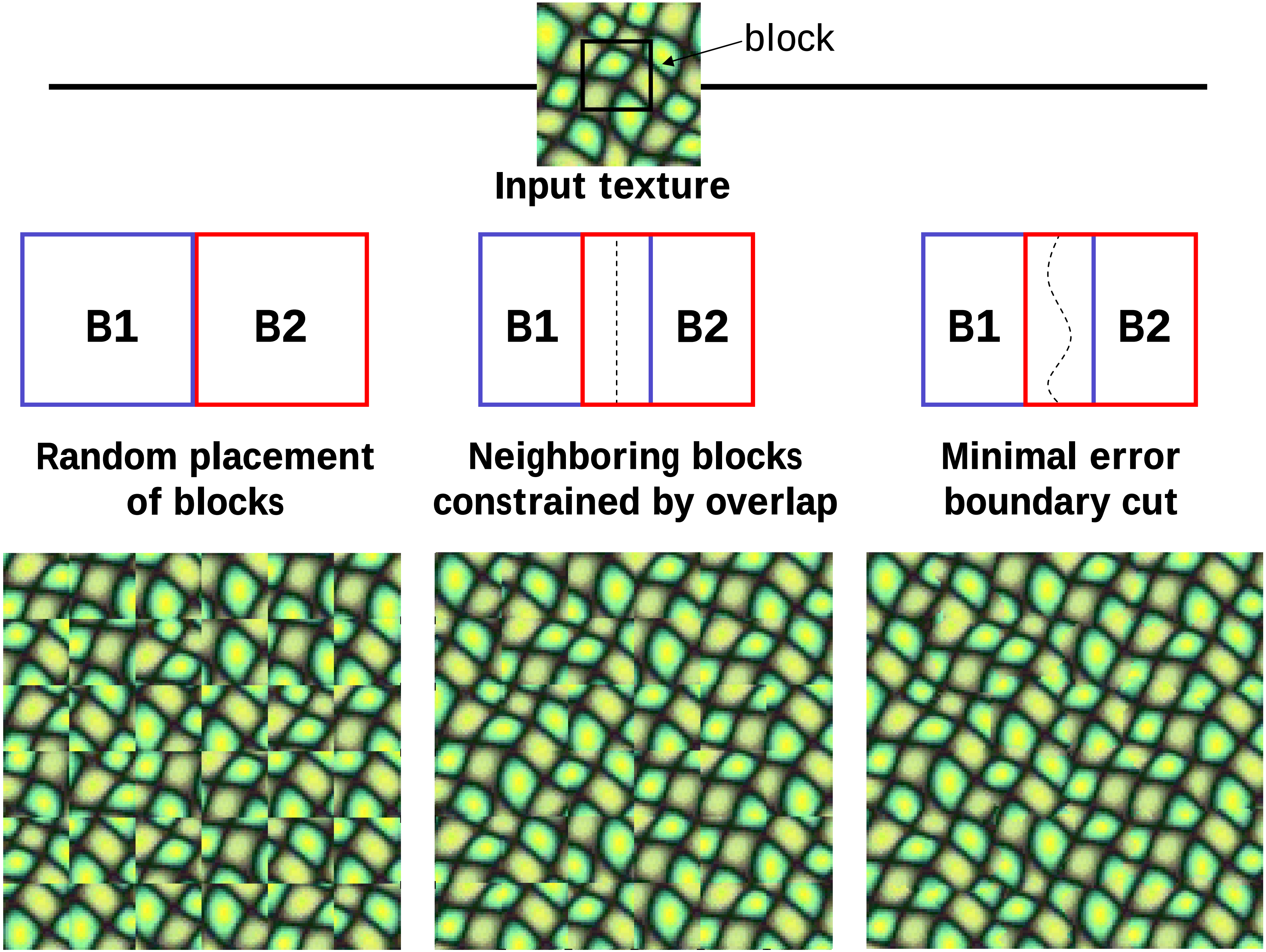
Synthesizing a block

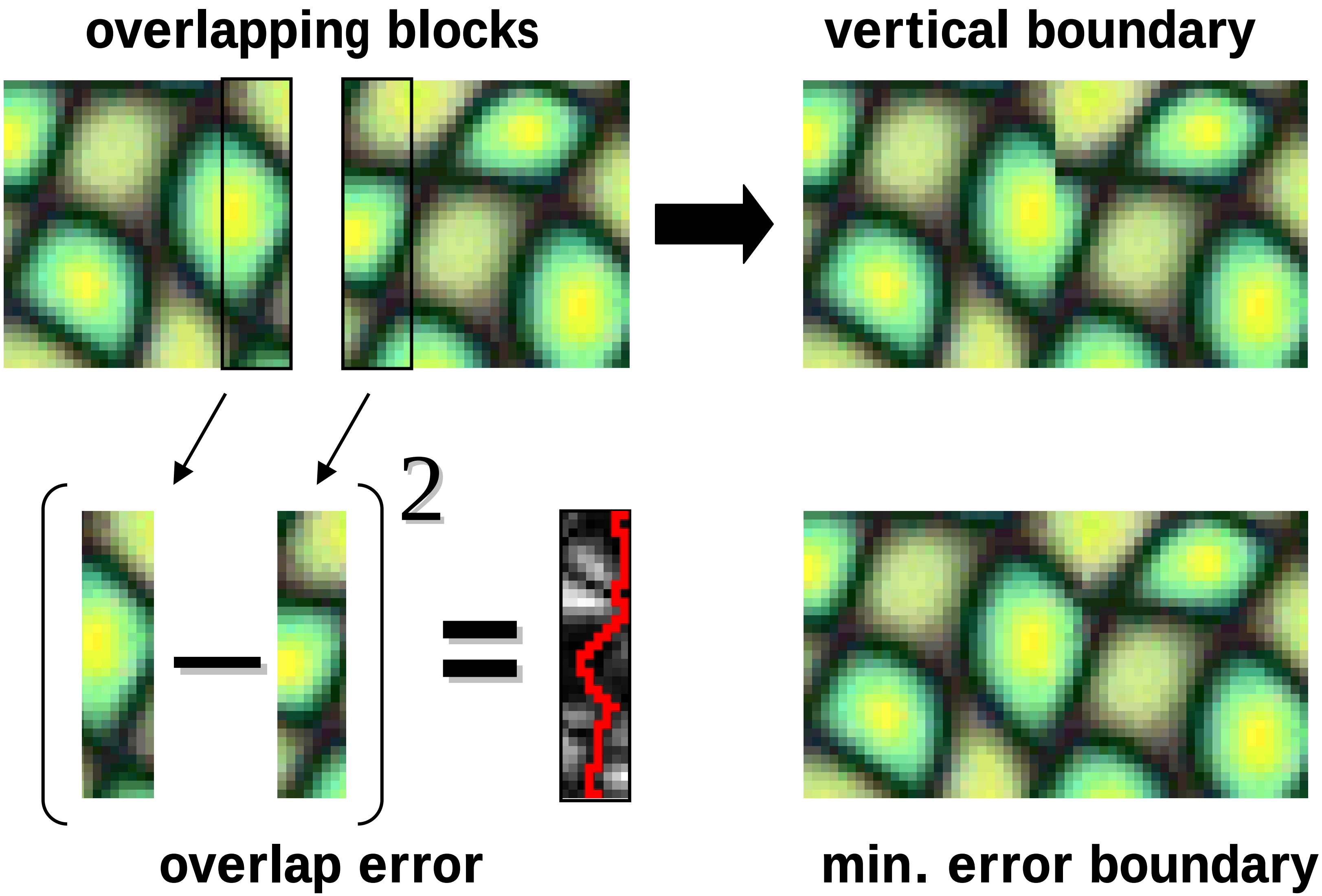
<u>Observation:</u> neighbor pixels are highly correlated

<u>Idea:</u> unit of synthesis = block

- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

[*Image Quilting for Texture Synthesis and Transfer*', Efros & Freeman, SIGGRAPH, 2001.]()


block
Input texture
B1
B2
B1
B2
B1
B2
Random placement of blocks
Neighboring blocks constrained by overlap
Minimal error boundary cut


overlapping blocks
vertical boundary
2
overlap error
min. error boundary

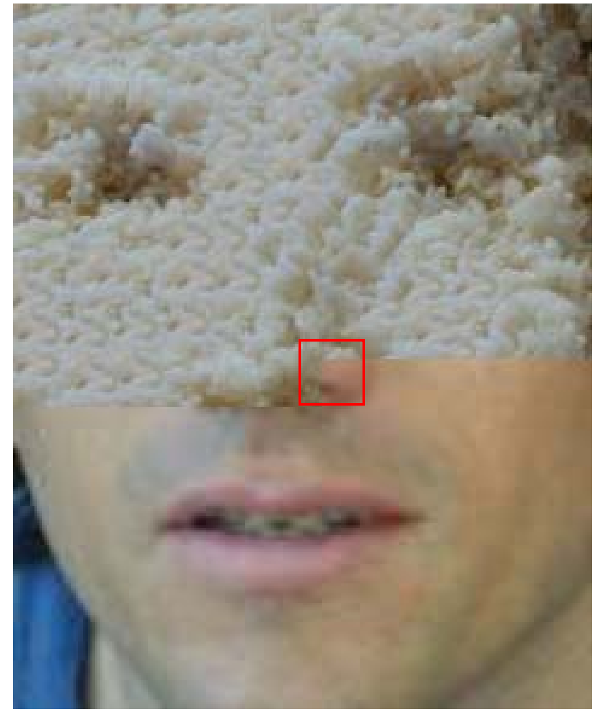
Texture Transfer

Constraint

Texture sample

Texture Transfer

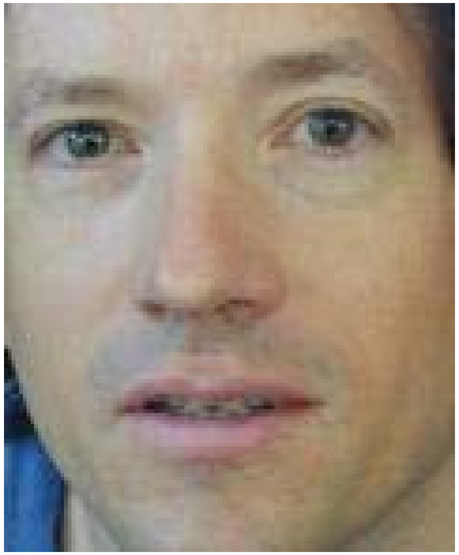
Take the texture from one image
and “paint” it onto another object



Same algorithm as before with additional term

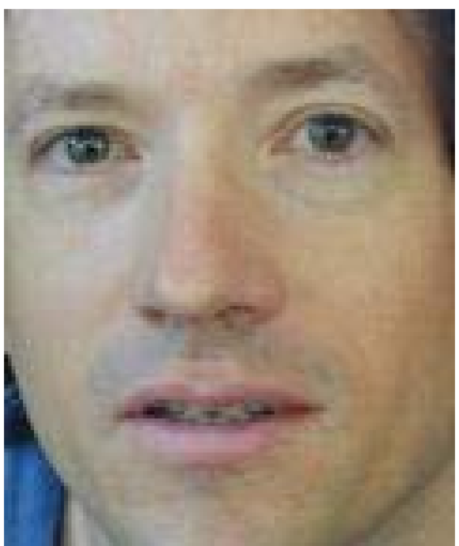
- do texture synthesis on image1, create new image (size of image2)
- add term to match intensity of image2

parmesan



+

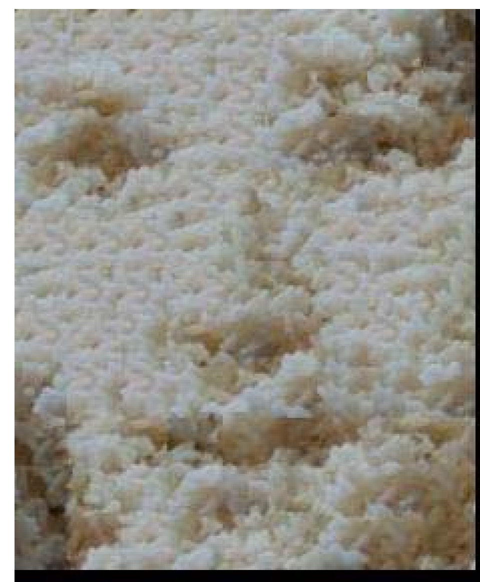
=

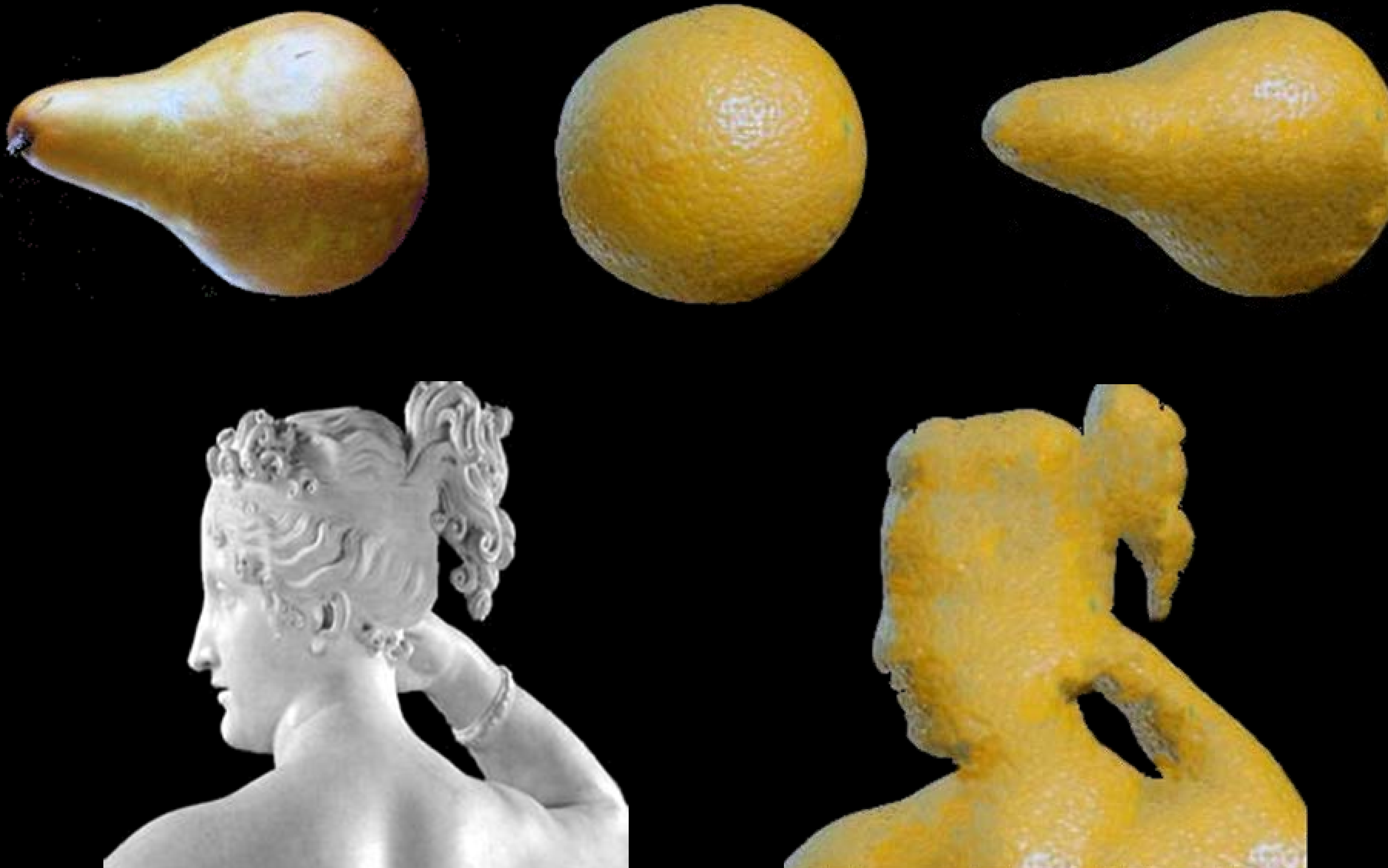


+

rice

=

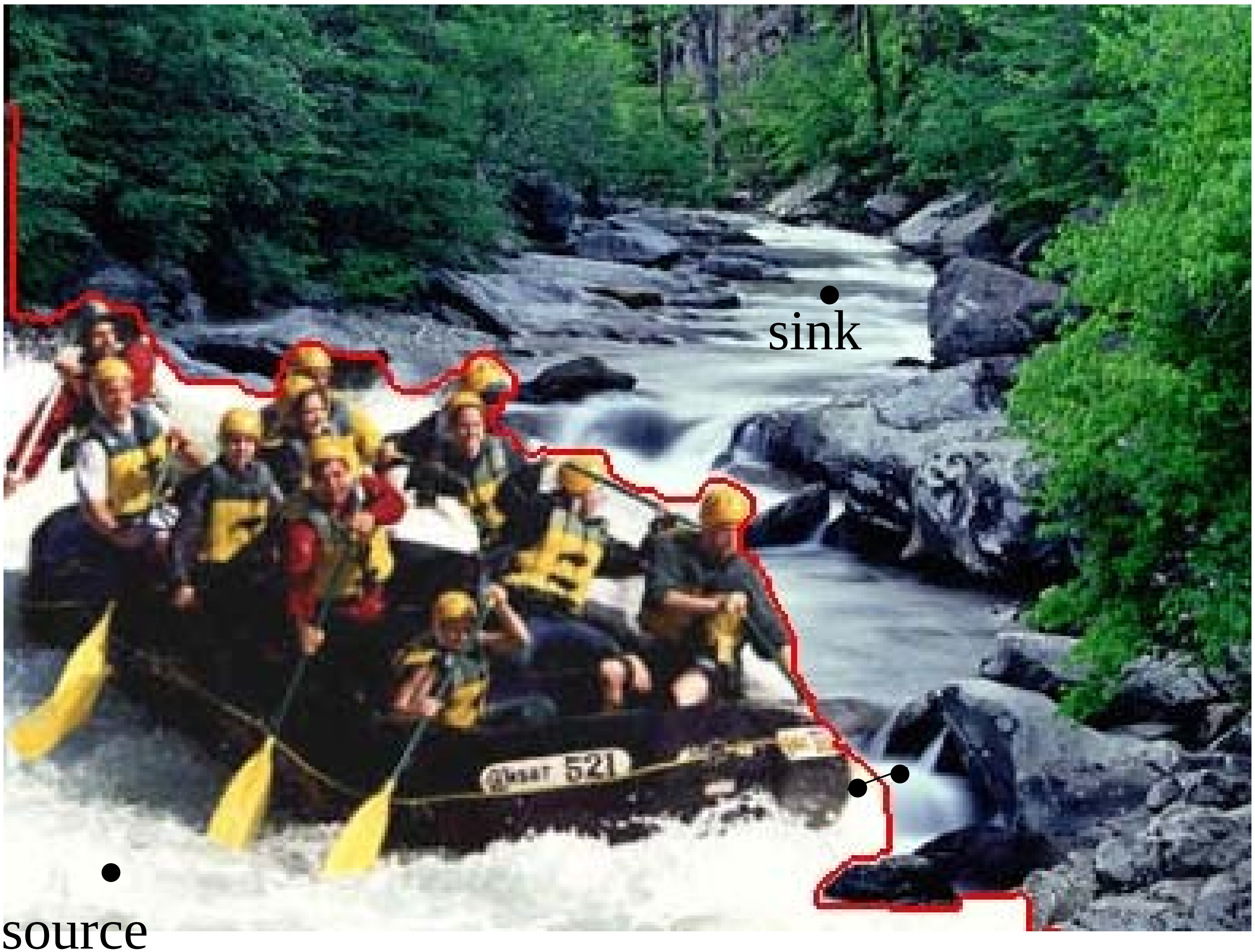




Combining two images

[*Graphcut Textures*, Kwatra et al., SIGGRAPH 2003.]

Graph cut setup



Graph cut texture synthesis: Video

Image Analogies (Hertzmann et al., '01)

A

A'

B

B'

Artistic Filters

A

A’

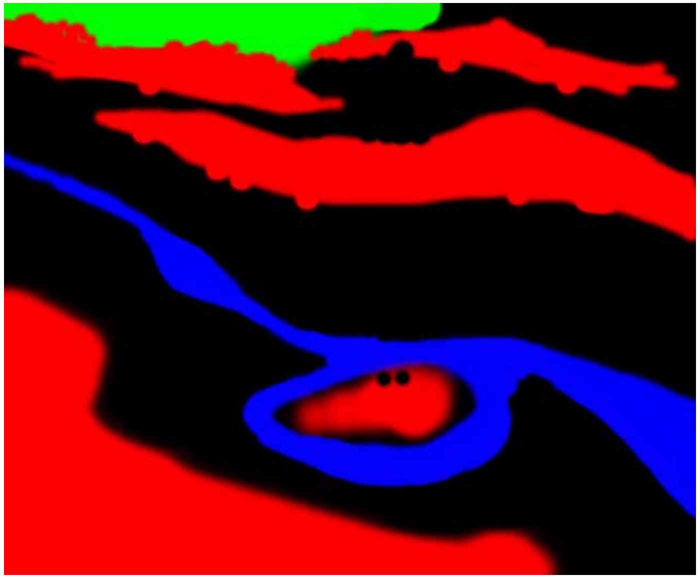
B

B’

Texture-by-numbers

A

A’



B

B’

Colorization

A

:

A’

::

B

:

B’

References

- Chap. 7, Shapiro and Stockman, Computer Vision, Prentice-Hall, 2001.
- Efros and Leung, "Texture Synthesis by Non-parametric Sampling," Proc. ICCV, 1999.
- Efros and Freeman, "Image Quilting for Texture Synthesis and Transfer" Proc. SIGGRAPH 2001.
- Kwatra, Schödl, Essa, Turk, and Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," Proc. SIGGRAPH 2003.
- Hertzmann, Jacobs, Oliver, Curless, and Salesin, "Image Analogies," Proc. SIGGRAPH 2001.

Next Time: Segmentation

Things to do:

- Begin working on Project 3
- Read Chap. 10