6.891

Computer Vision and Applications

Prof. Trevor. Darrell

Lecture 4: Texture

- Filter-based models
- Example-based / Non-parametric approaches
- Quilting and Epitomes

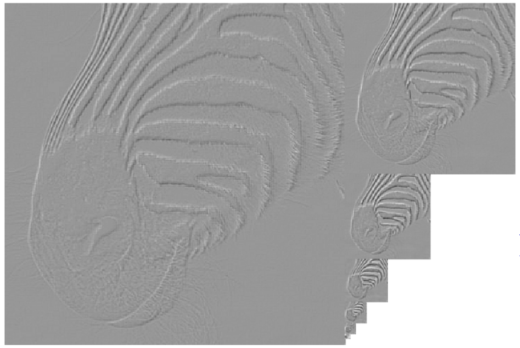
Readings: F & P 9.1, 9.3, 9.4

Last time: image pyramids

- Gaussian

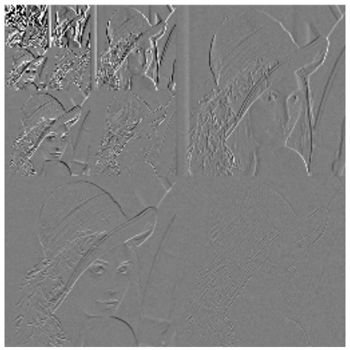
Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

- Laplacian



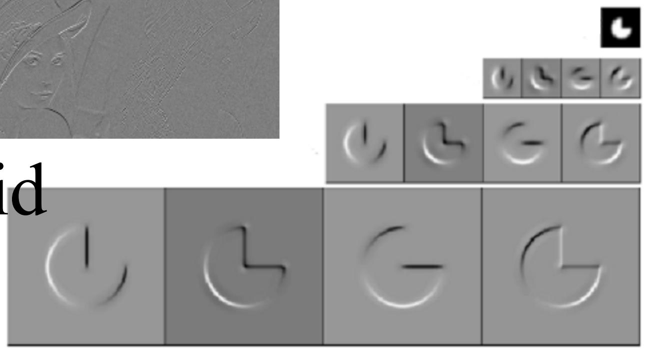
Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- Wavelet/QMF



Bandpassed representation, complete, but with aliasing and some non-oriented subbands.

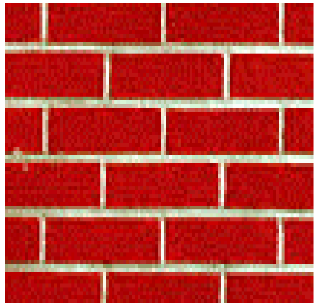
- Steerable pyramid



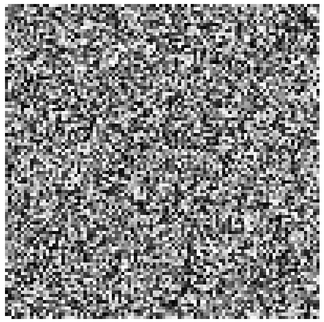
Shows components at each scale and orientation separately. Non-aliased subbands. Good for texture and feature analysis.

The Challenge

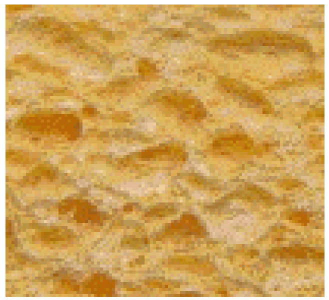
- How to capture the essence of texture?
- Need to model the whole spectrum: from repeated to stochastic texture
- This problem is at intersection of vision, graphics, statistics, and image compression



repeated



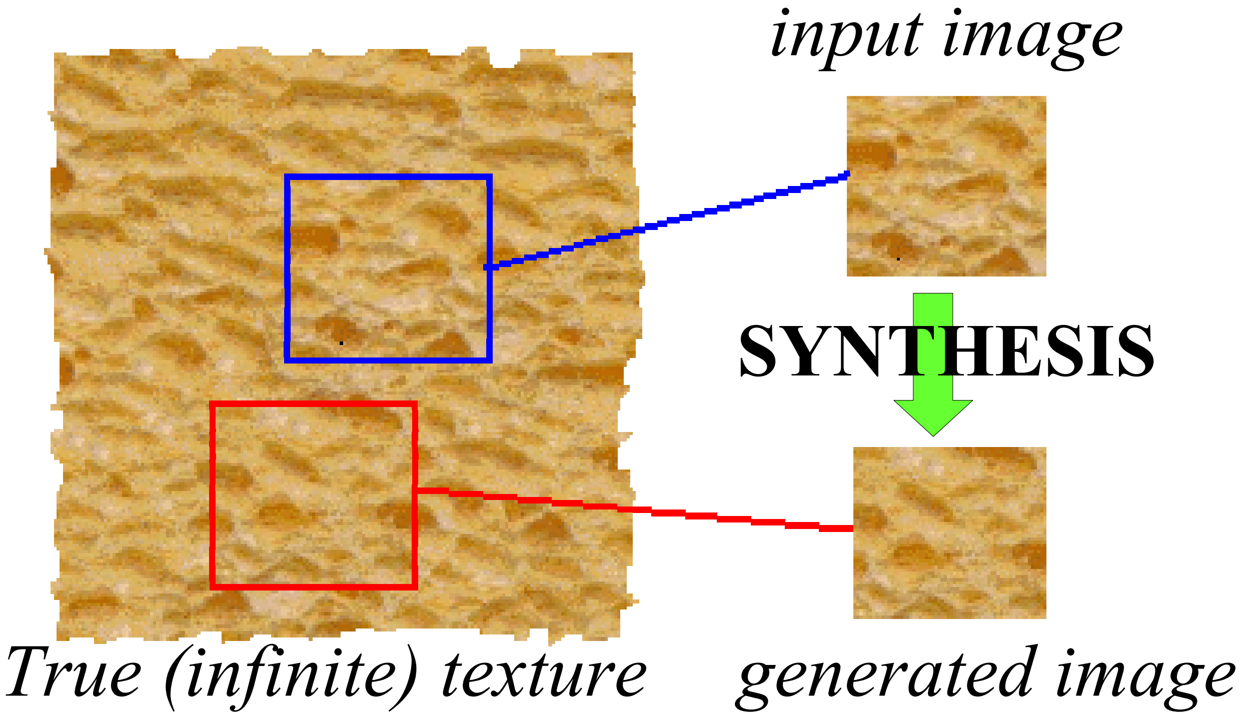
stochastic



Both?

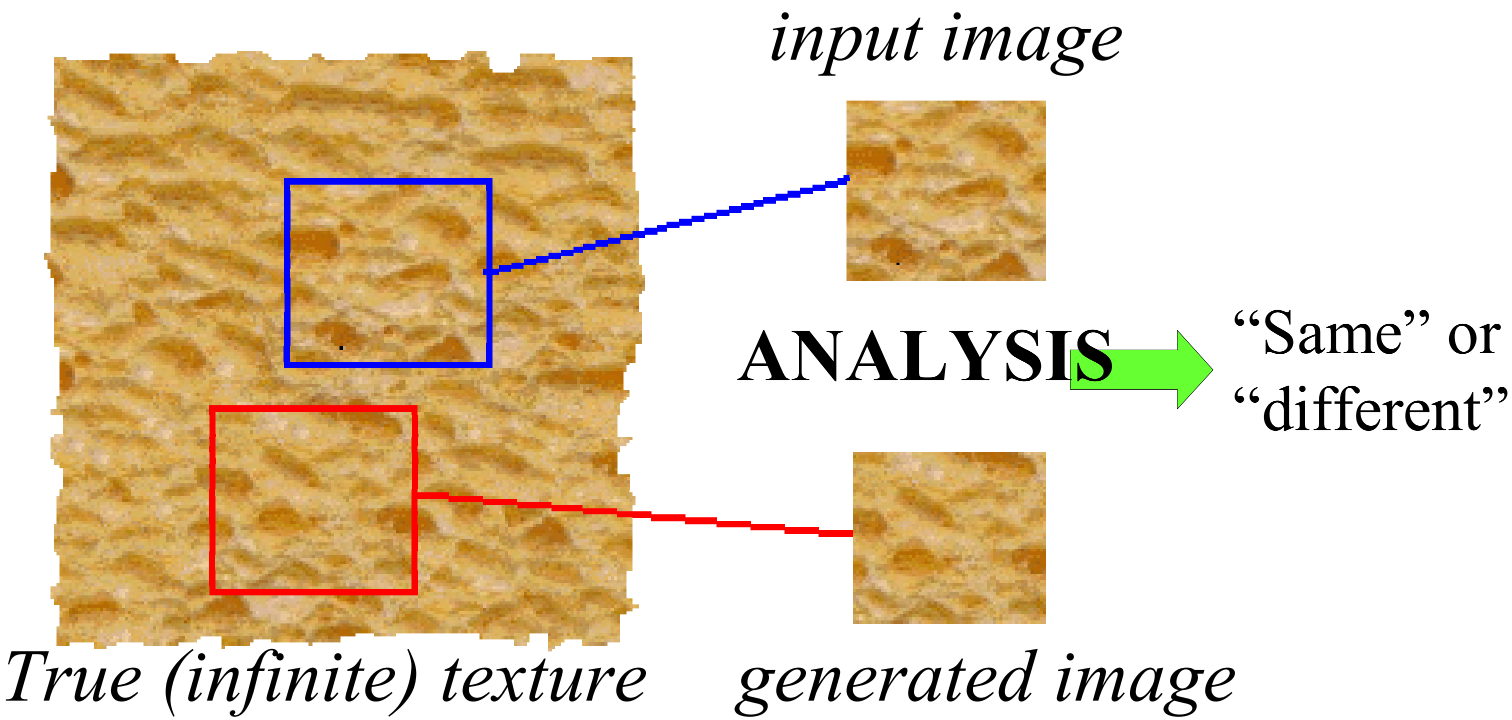
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

The Goal of Texture Synthesis



- Given a finite sample of some texture, the goal is to synthesize other samples from that same texture
  - The sample needs to be "large enough“

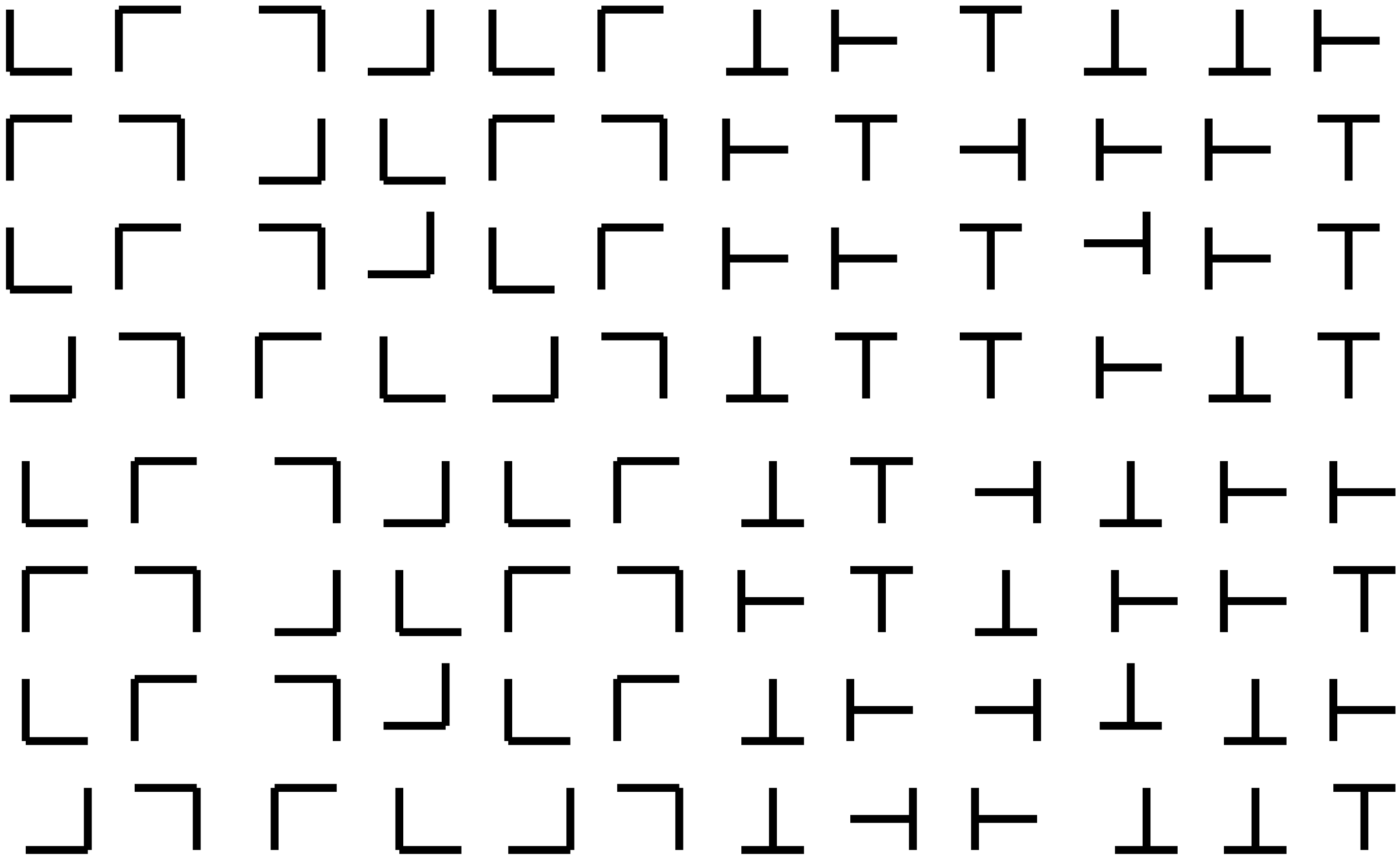
The Goal of Texture Analysis



Compare textures and decide if they're made of the same "stuff".

Pre-attentive texture discrimination

Pre-attentive texture discrimination

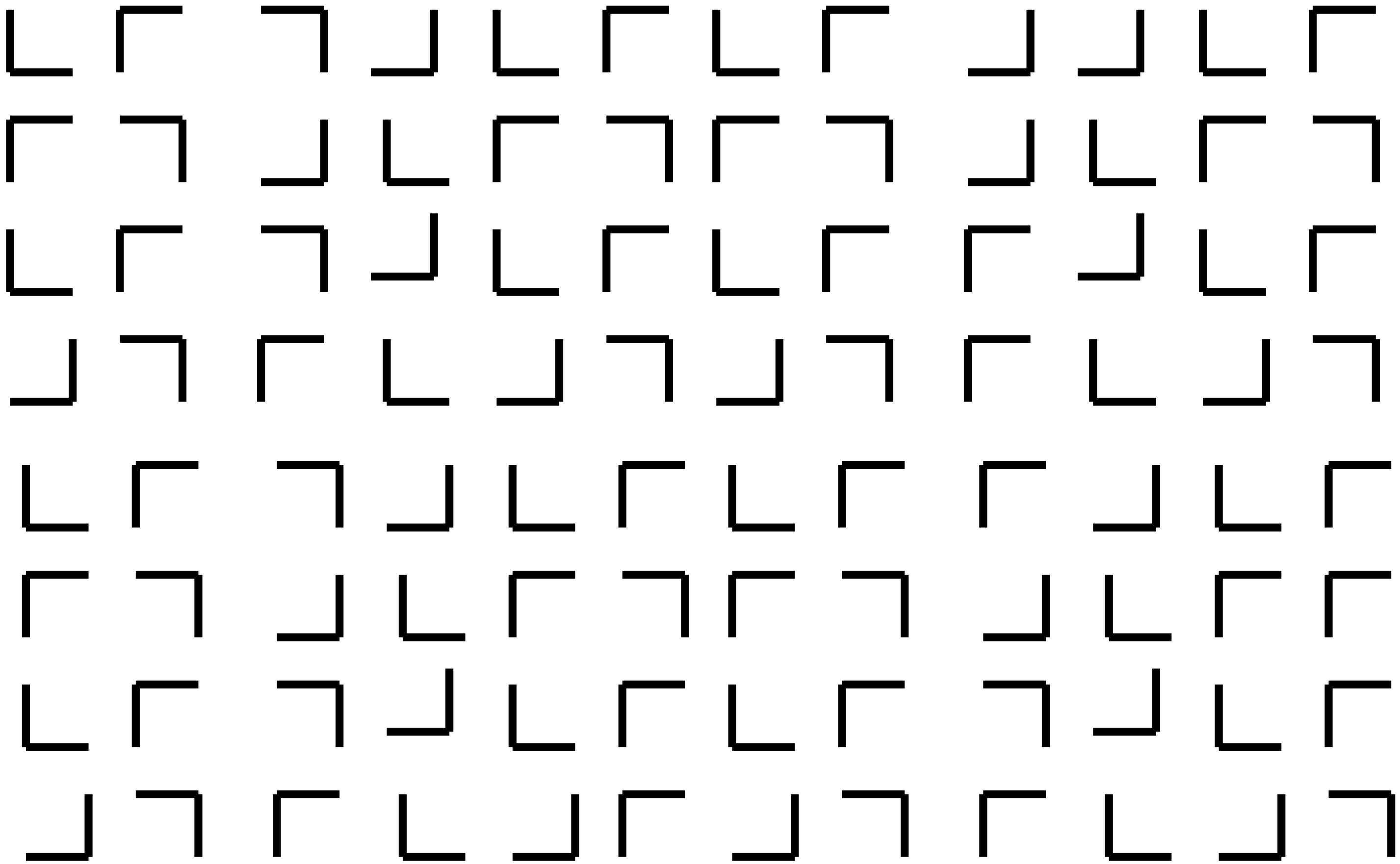


Pre-attentive texture discrimination

Same or different textures?

Pre-attentive texture discrimination

Pre-attentive texture discrimination

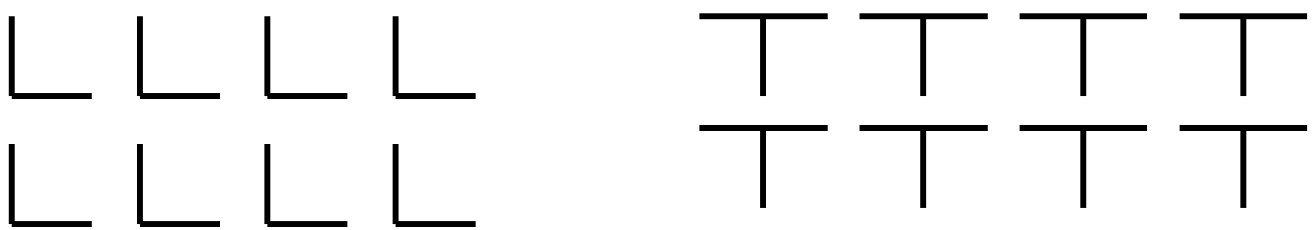


Pre-attentive texture discrimination

Same or different textures?

Julesz

- Textons: analyze the texture in terms of statistical relationships between fundamental texture elements, called "textons".
- It generally required a human to look at the texture in order to decide what those fundamental units were...



Representing textures

- Textures are made up of quite stylized subelements, repeated in meaningful ways
- Representation:
  - find the subelements, and represent their statistics
- But what are the subelements, and how do we find them?
  - recall normalized correlation
  - find subelements by applying filters, looking at the magnitude of the response

Influential early paper:

Early vision and texture perception

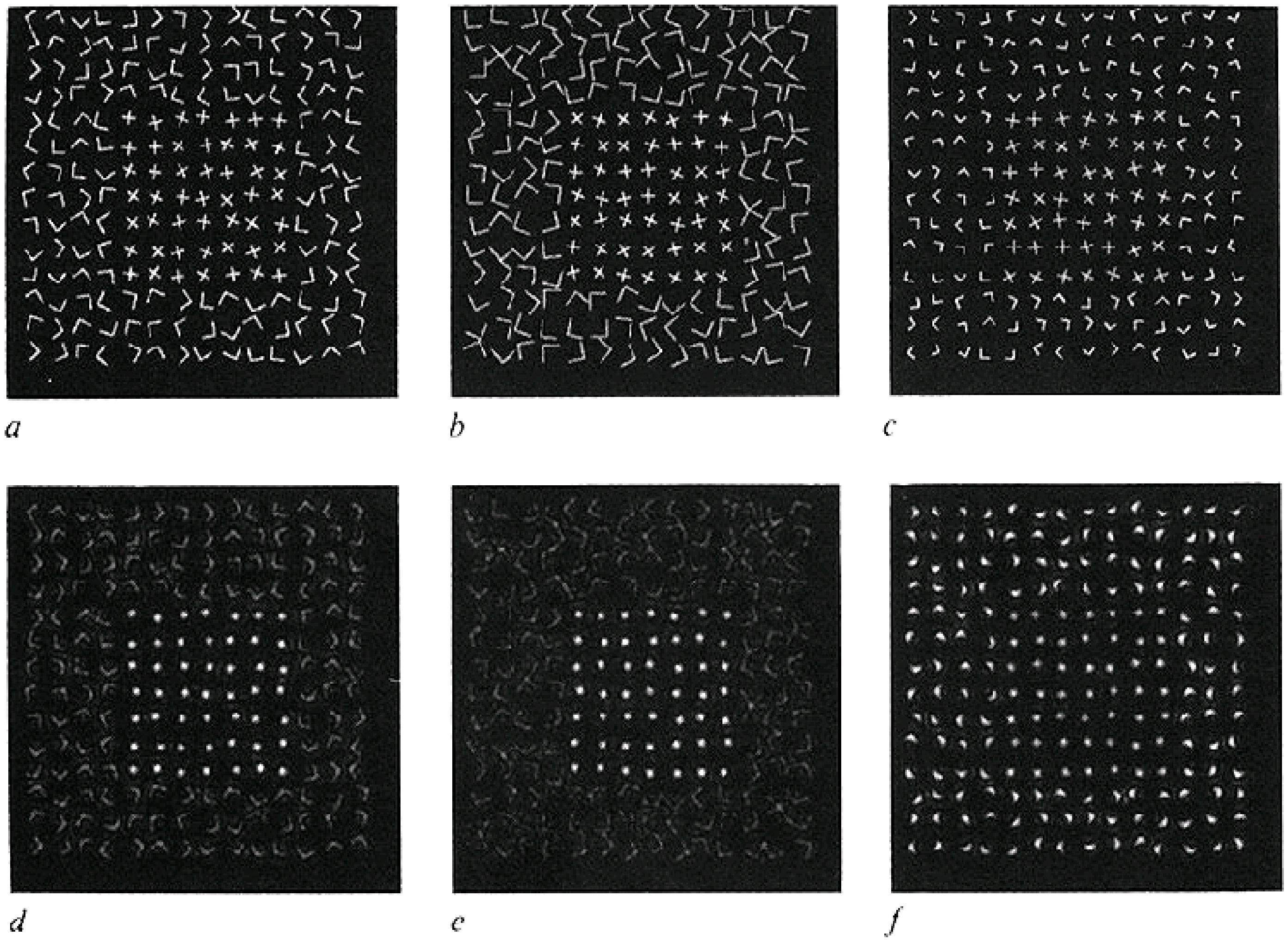
James R. Bergen* & Edward H. Adelson**

* SRI David Sarnoff Research Center, Princeton,
New Jersey 08540, USA
** Media Lab and Department of Brain and Cognitive Science,
Massachusetts Institute of Technology, Cambridge,
Massachusetts 02139, USA

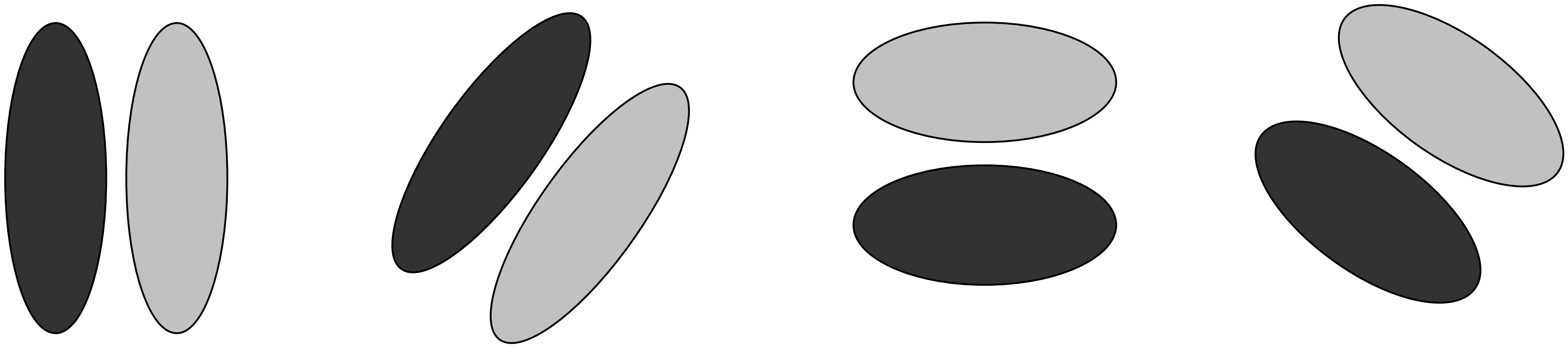
Bergen and Adelson, Nature 1988

Learn size-tuned filter responses.

Fig. 1 *Top row*, Textures consisting of Xs within a texture composed of Ls. The micropatterns are placed at random orientations on a randomly perturbed lattice. *a*, The bars of the Xs have the same length as the bars of the Ls. *b*, The bars of the Ls have been lengthened by 25%, and the intensity adjusted for the same mean luminance. Discriminabitity is enhanced. *c*, The bars of the Ls have been shortened by 25%, and the intensity adjusted for the same mean luminance. Discriminabitity is impaired. *Bottom row*: the responses of a size-tuned mechanism *d*, response to image *a*; *e*, response to image *b*; *f*, response to image *c*.

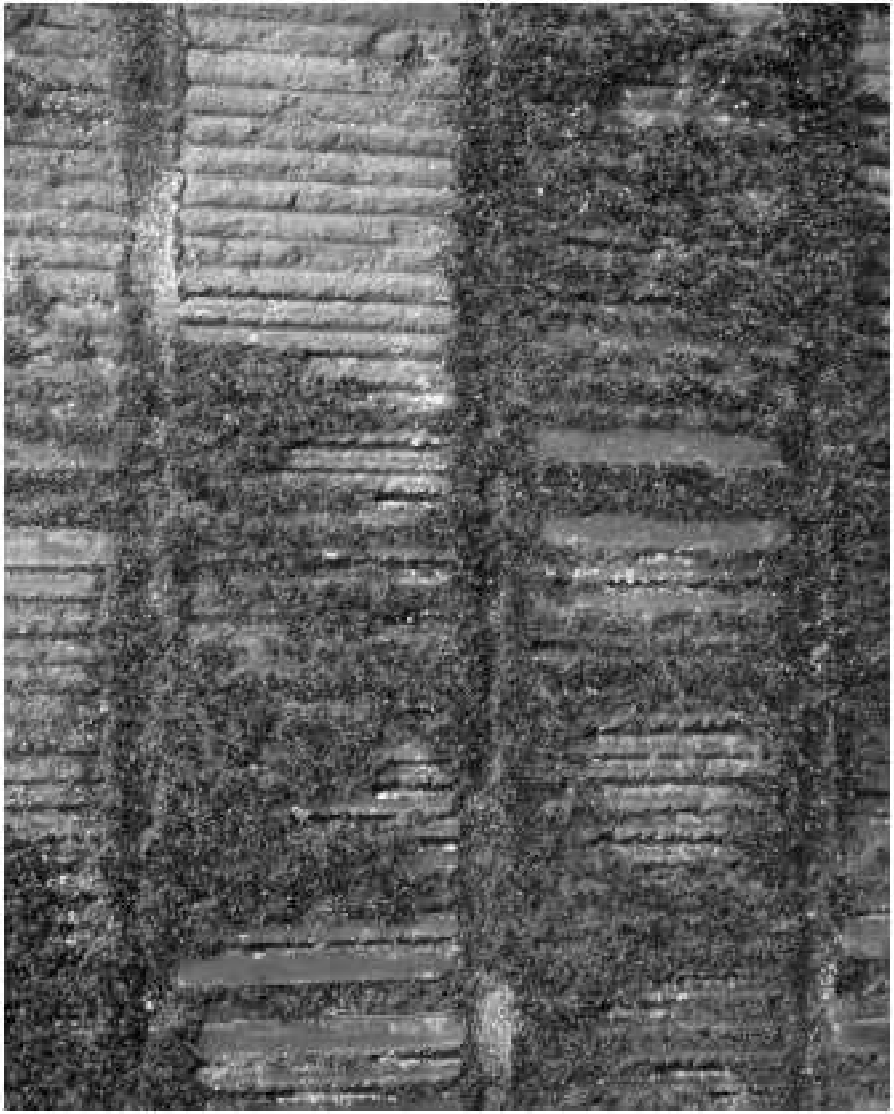


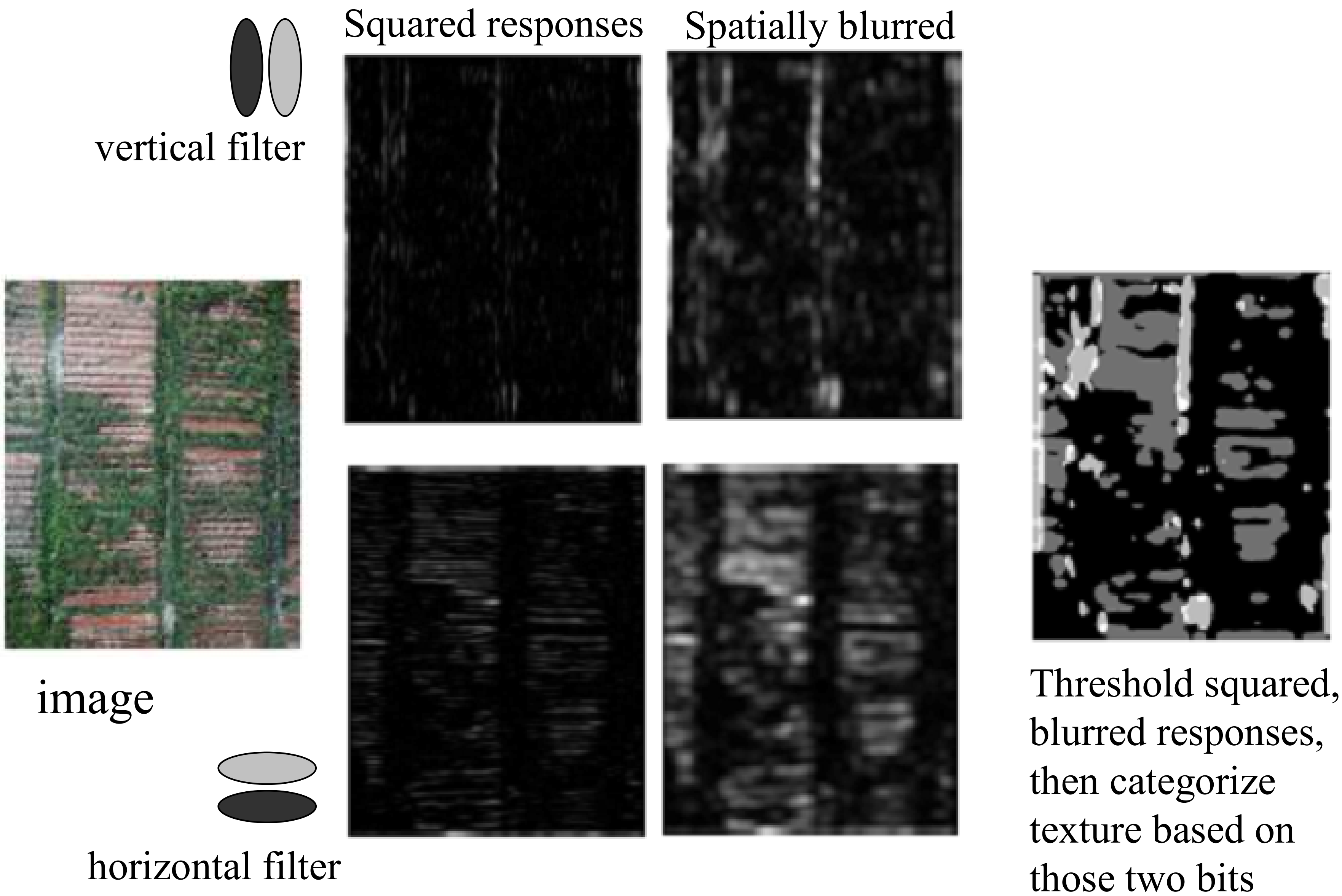
Malik and Perona

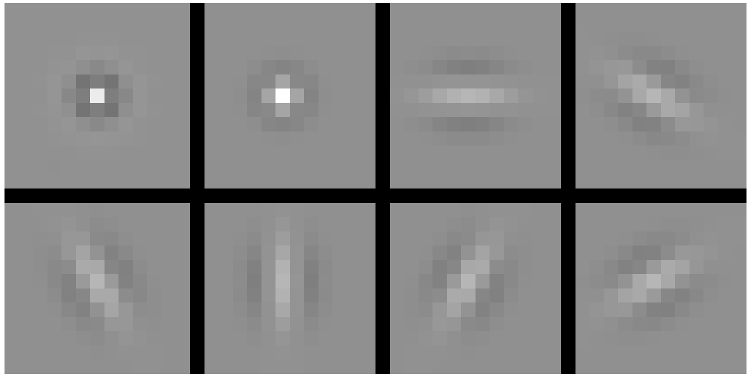


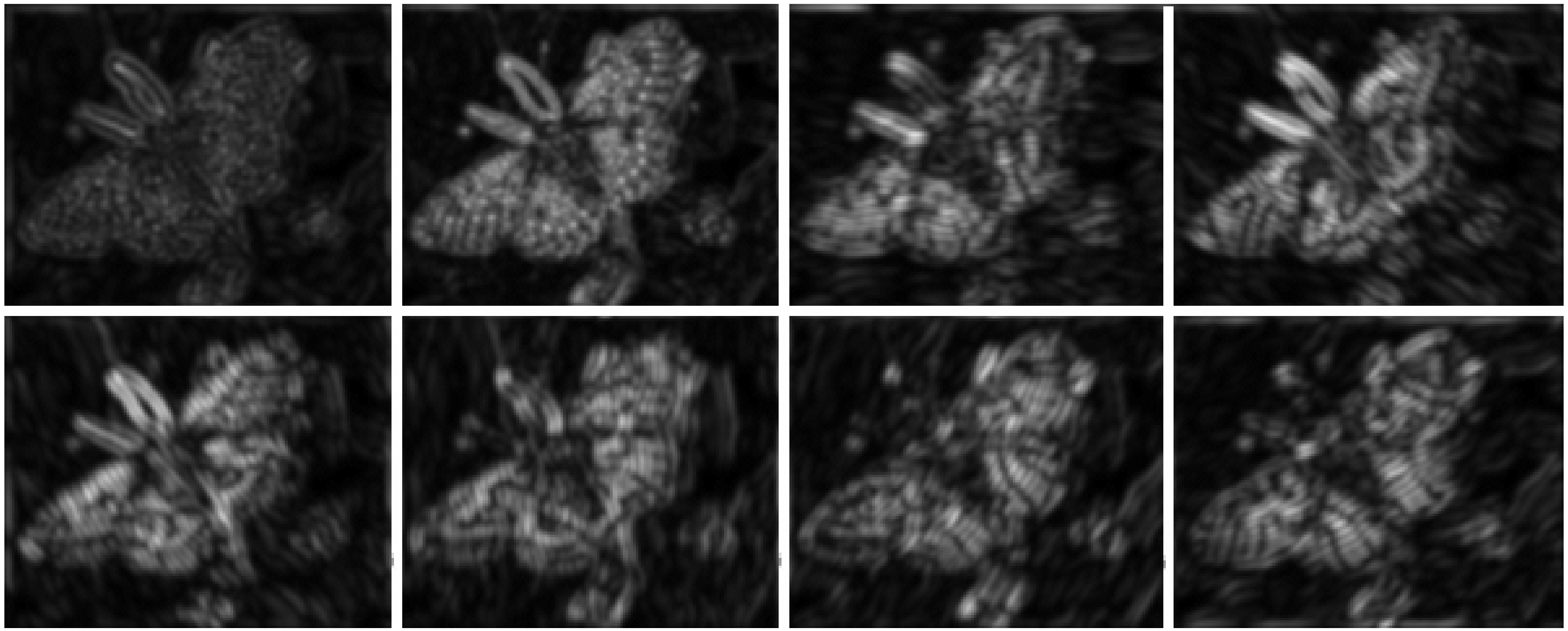
Learn: use lots of filters, multi-ori&scale.

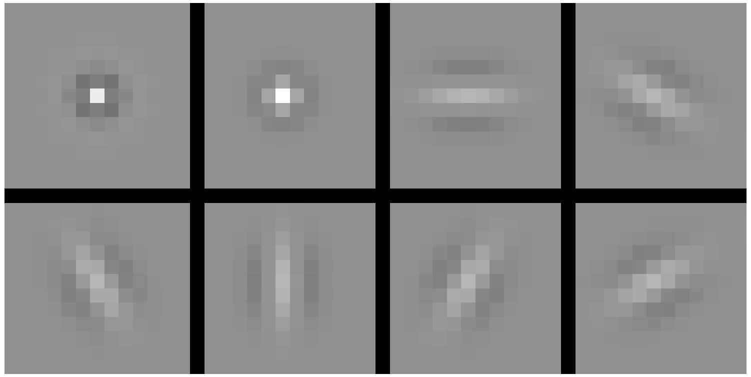
Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms. J OPT SOC AM A 7: (5) 923-932 MAY 1990

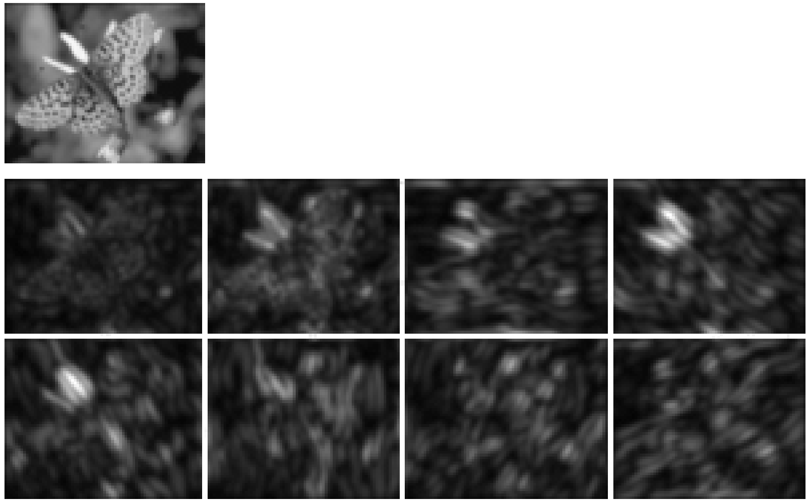


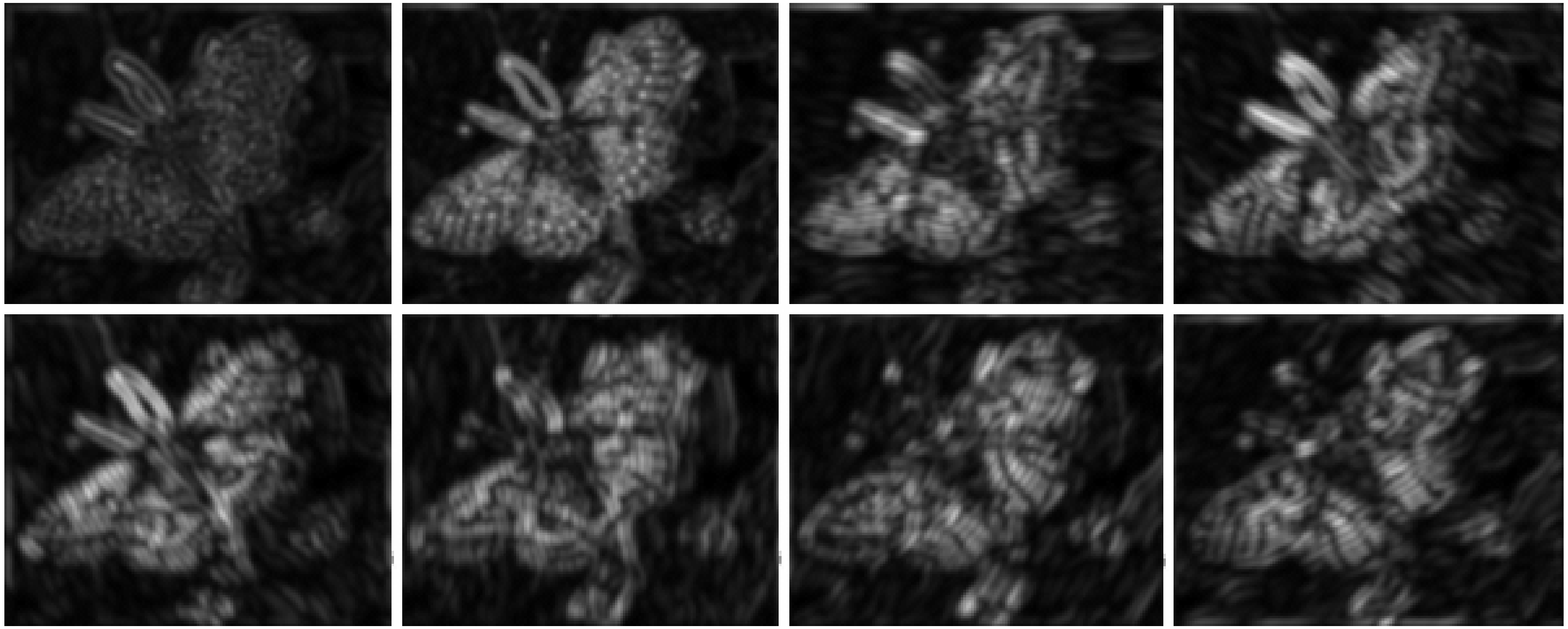

Squared responses
Spatially blurred
vertical filter
image
horizontal filter
Threshold squared,
blurred responses,
then categorize
texture based on
those two bits

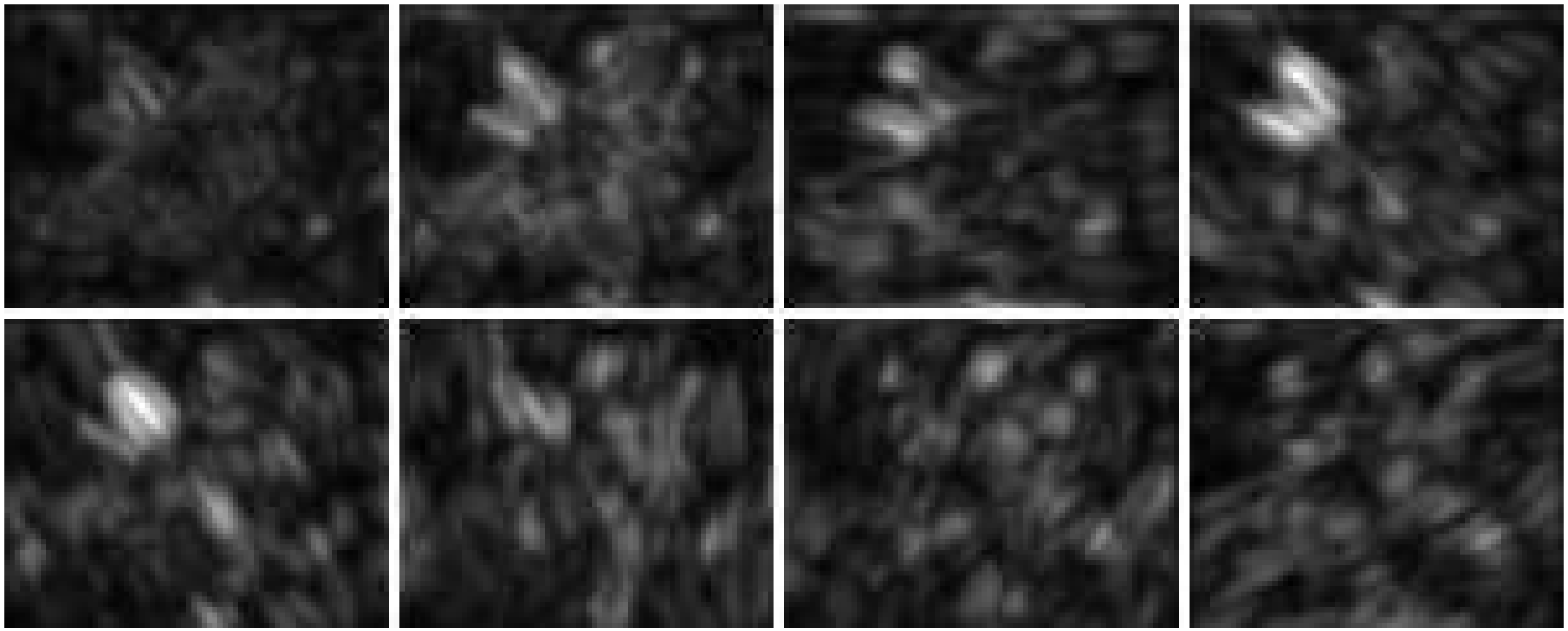








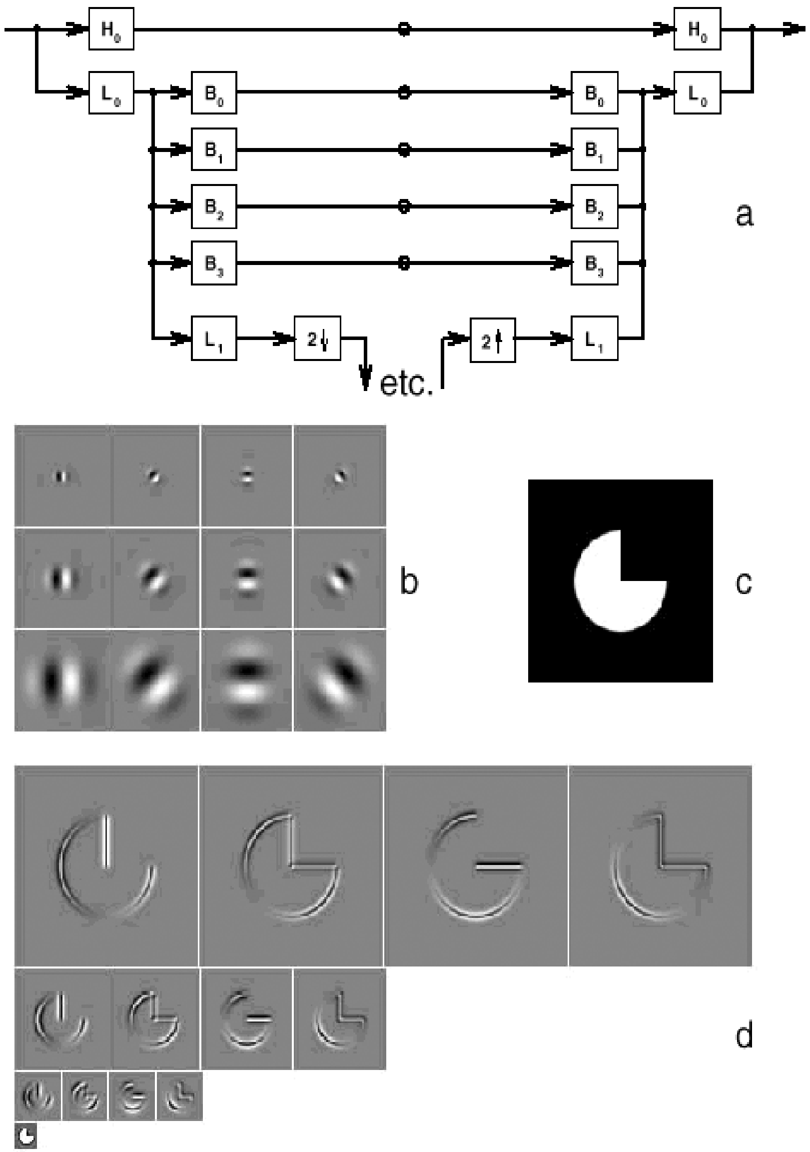




Pyramid-Based Texture Analysis/Synthesis

David J. Heeger[*]
Stanford University

James R. Bergen[†]
SRI David Sarnoff Research Center



SIGGRAPH 1994

Bergen and Heeger

Idea: Learn filter marginal statistics.

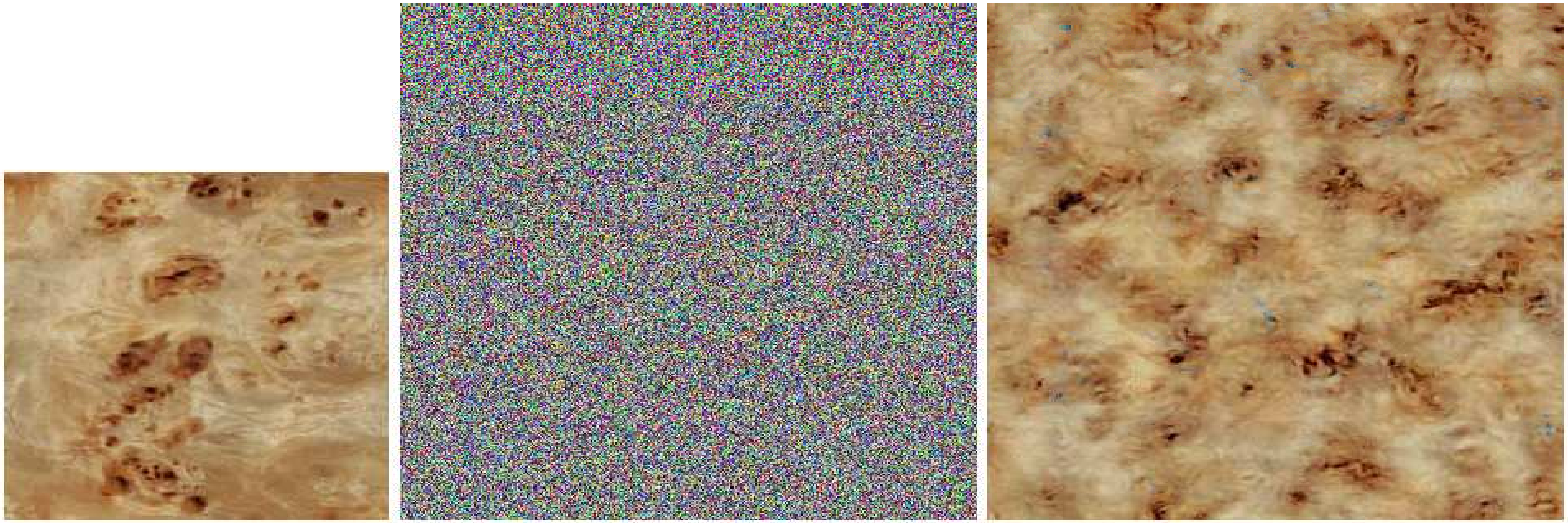


Figure 2: (Left) Input digitized sample texture: burled mappa wood. (Middle) Input noise. (Right) Output synthetic texture that matches the appearance of the digitized sample. Note that the synthesized texture is larger than the digitized sample; our approach allows generation of as much texture as desired. In addition, the synthetic textures tile seamlessly.

Bergen and Heeger results

Figure 3: In each pair left image is original and right image is synthetic: stucco, iridescent ribbon, green marble, panda fur, slag stone, figured yew wood.

Bergen and Heeger failures

Figure 8: Examples of failures: wood grain and red coral.

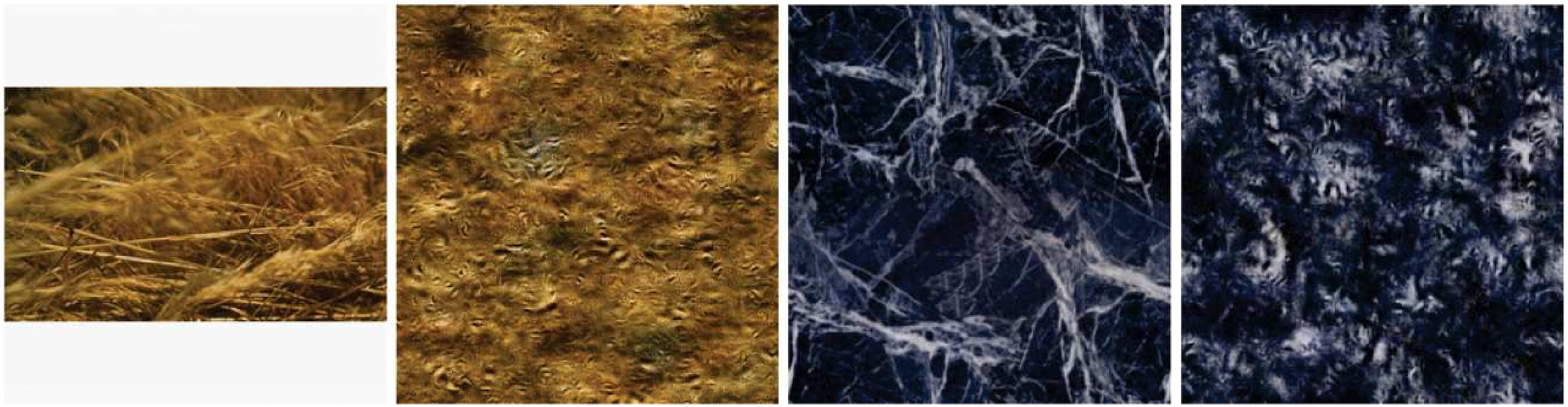


Figure 9: More failures: hay and marble.

DeBonet

Learn filter conditional statistics across scale.

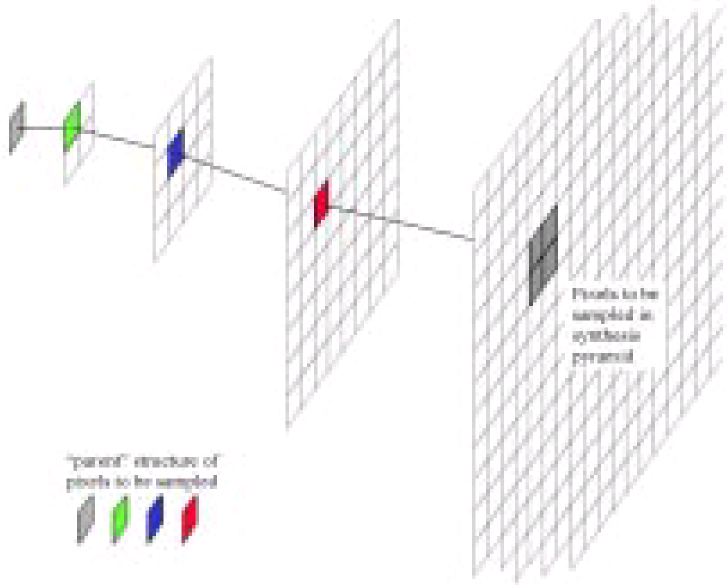


Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the "parent" structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

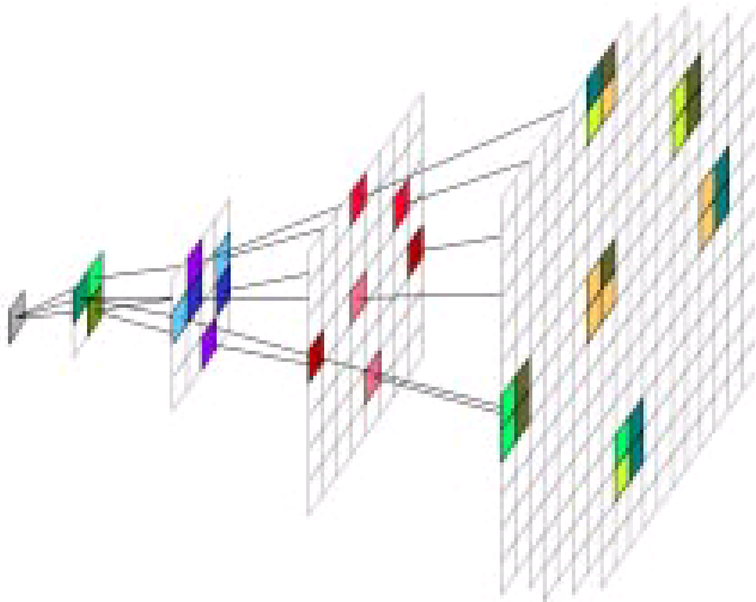
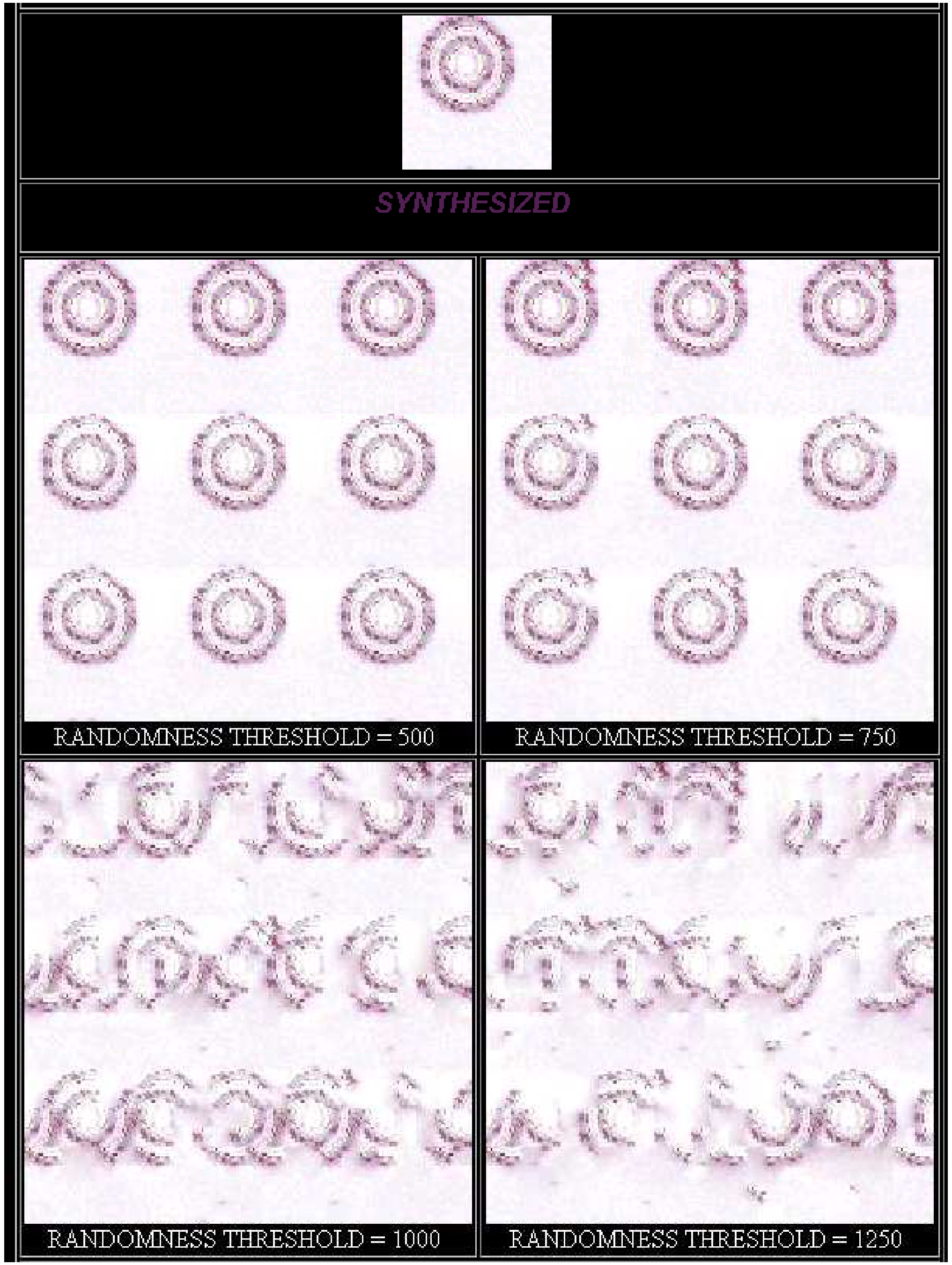
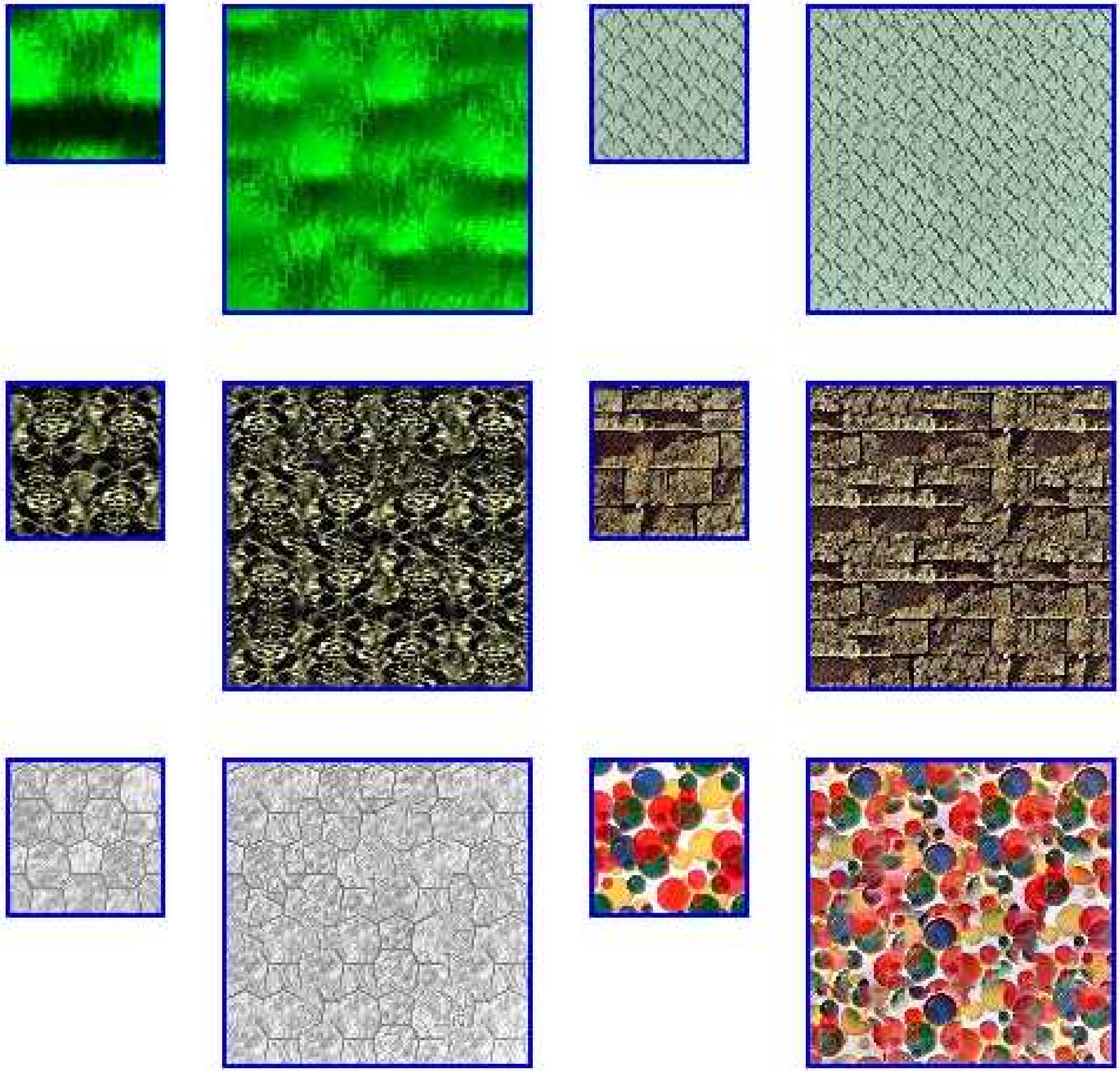


Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.


SYNTHESIZED
RANDOMNESS THRESHOLD = 500
RANDOMNESS THRESHOLD = 750
RANDOMNESS THRESHOLD = 1000
RANDOMNESS THRESHOLD = 1250

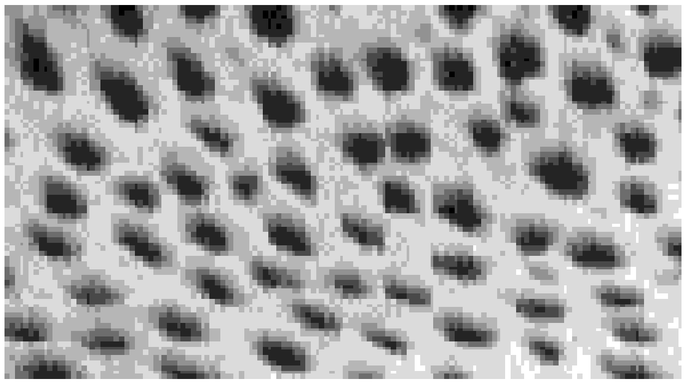
DeBonet

DeBonet



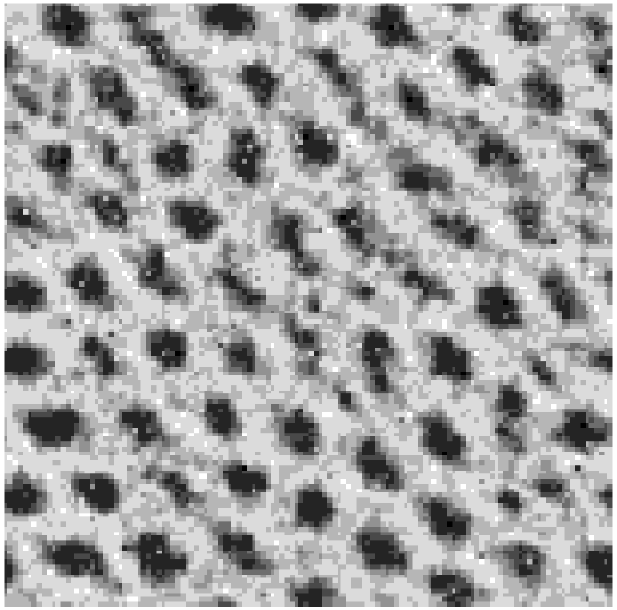
Zhu, Wu, & Mumford, 1998

Gibbs sampling of Markov Random Field model:



a

Cheetah



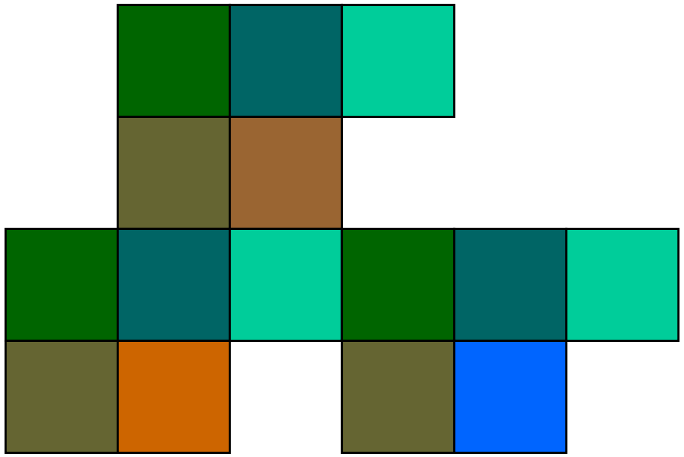
b

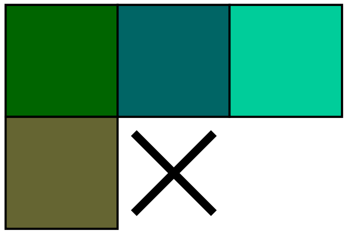
Synthetic

IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

Texture Synthesis by Non-parametric Sampling

Alexei A. Efros and Thomas K. Leung
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, U.S.A.
{efros,leungt}@cs.berkeley.edu



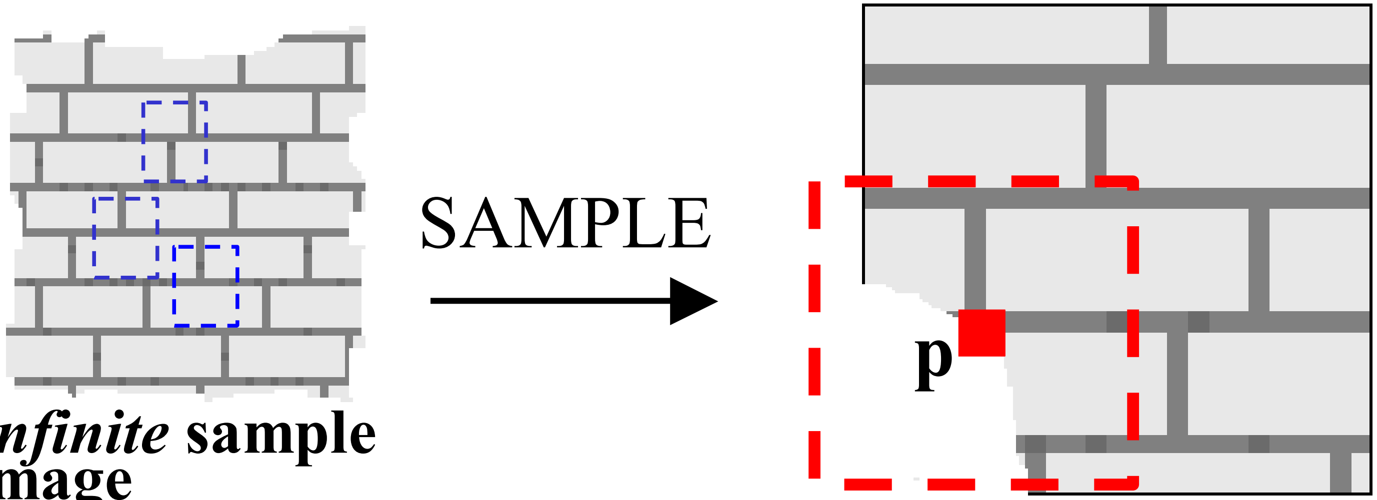


Efros and Leung ‘99

- preserve local structure
- model wide range of real textures
- ability to do constrained synthesis
- method:
  - Texture is “grown” one pixel at a time
  - conditional pdf of pixel given its neighbors synthesized thus far is computed directly from the sample image

http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Synthesizing One Pixel



Infinite **sample image**

Generated image

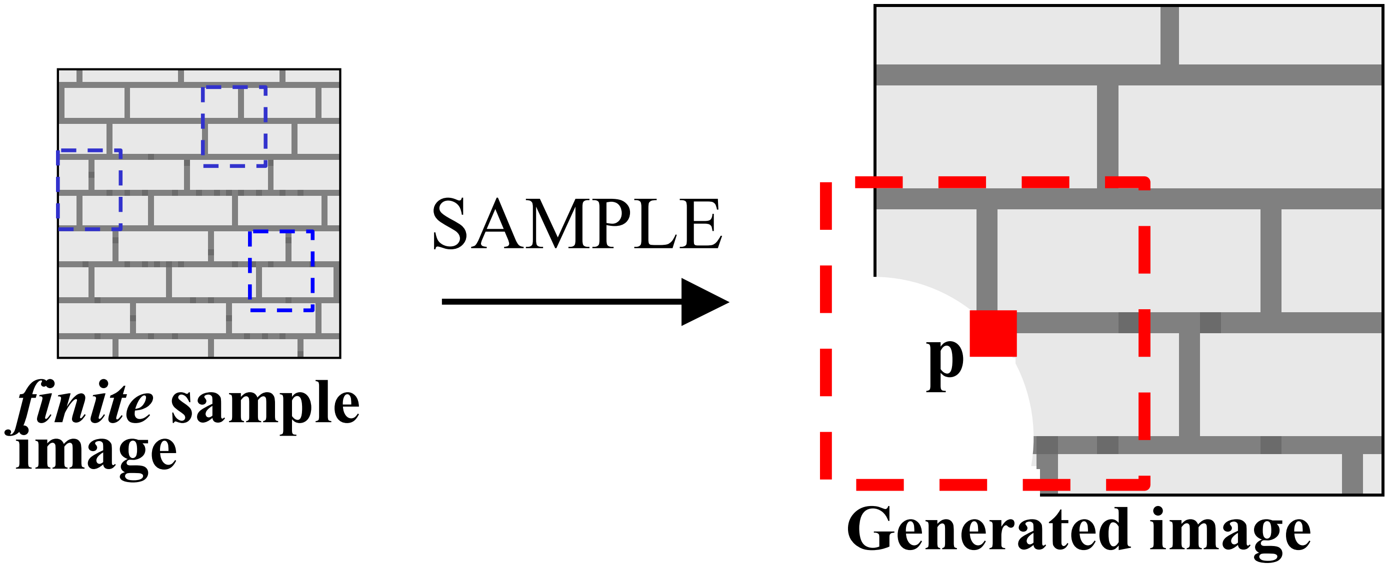
– Assuming Markov property, what is conditional probability distribution of p, given the neighbourhood window?

– Instead of constructing a model, let's directly search the input image for all such neighbourhoods to produce a histogram for p

– To synthesize p, just pick one match at random

http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Really Synthesizing One Pixel



– However, since our sample image is finite, an exact neighbourhood match might not be present

– So we find the **best** match using SSD error (weighted by a Gaussian to emphasize local structure), and take all samples within some distance from that match

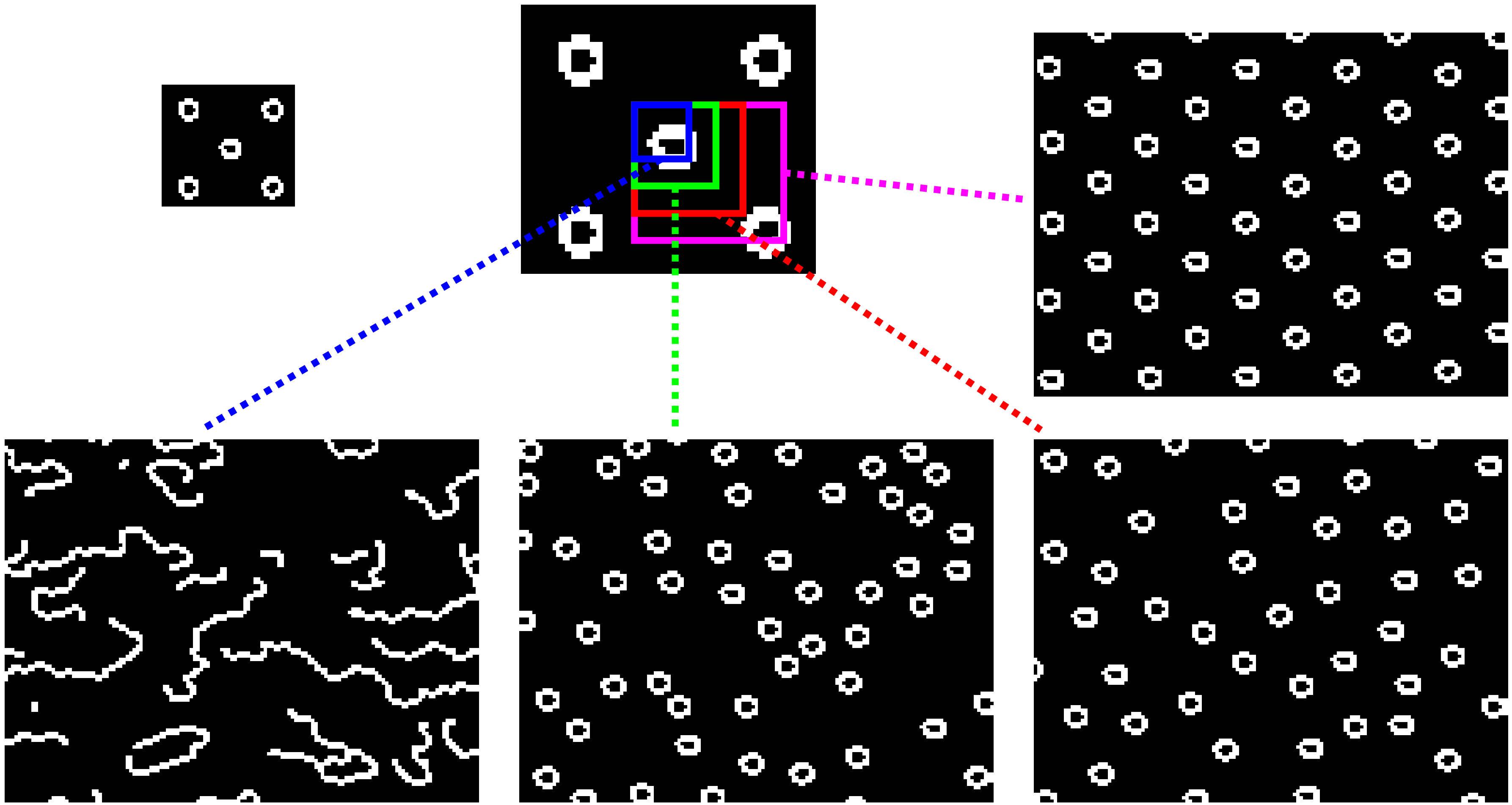
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Growing Texture

– Starting from the initial configuration, we "grow" the texture one pixel at a time
– The size of the neighbourhood window is a parameter that specifies how stochastic the user believes this texture to be
– To grow from scratch, we use a random 3x3 patch from input image as seed

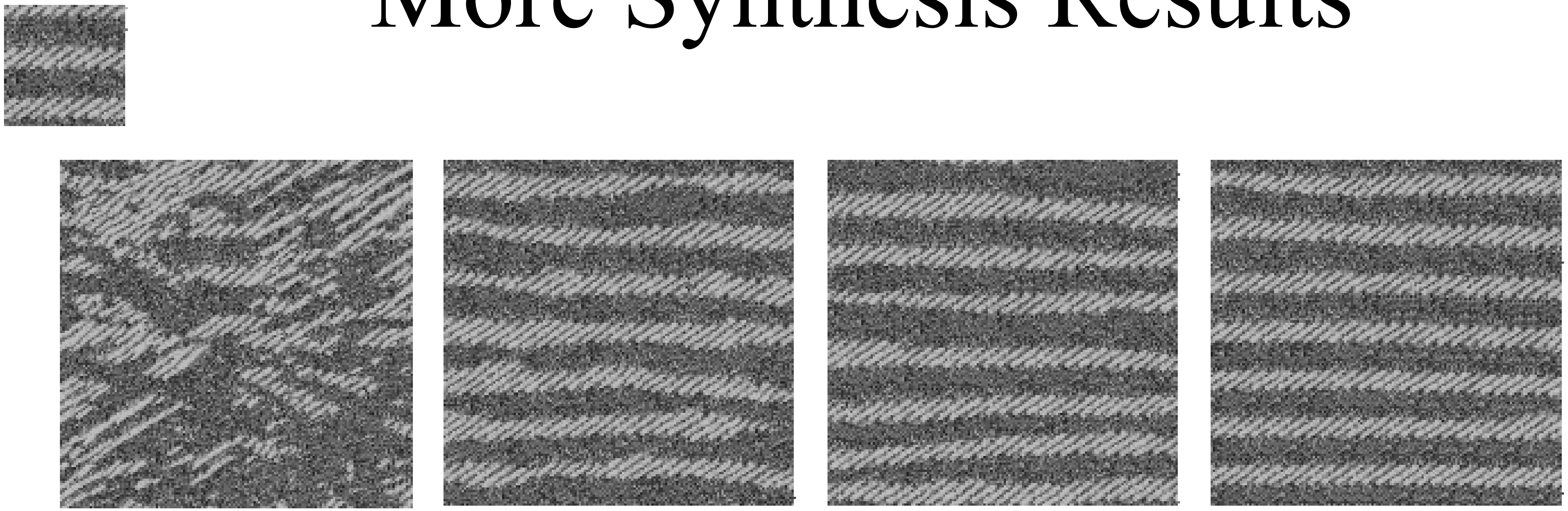
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

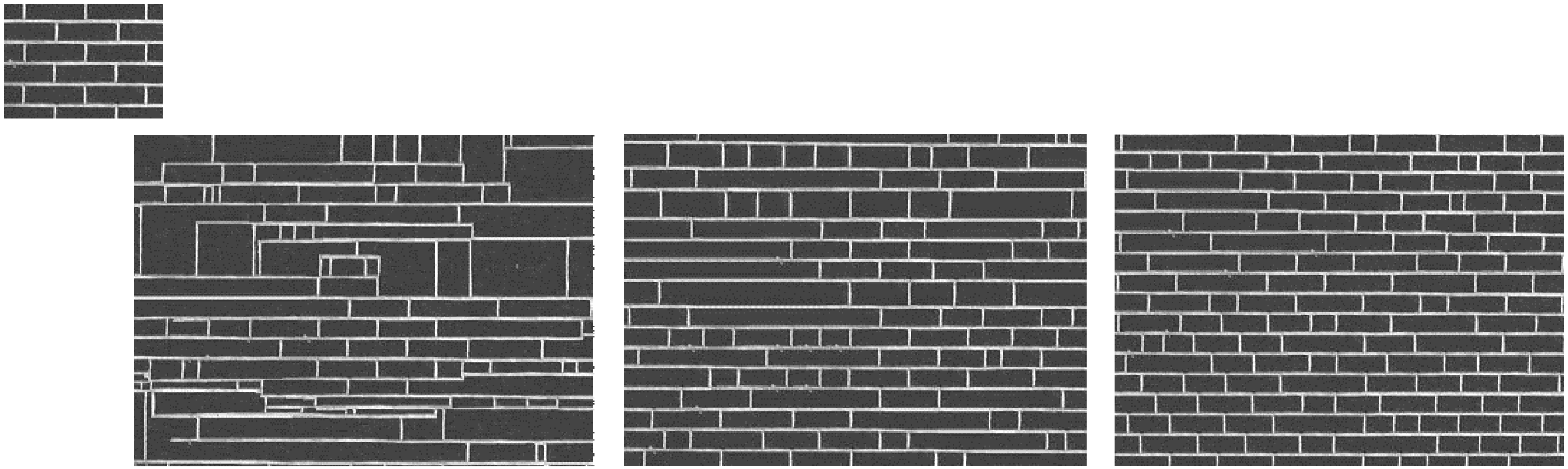
Randomness Parameter



http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

More Synthesis Results



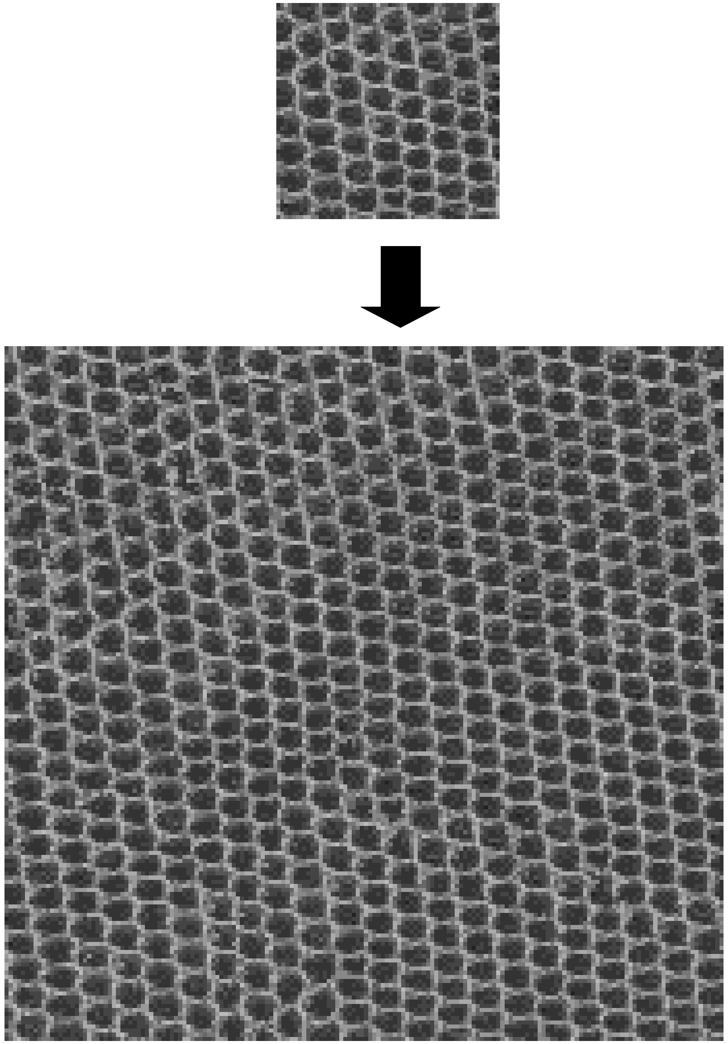


Increasing window size

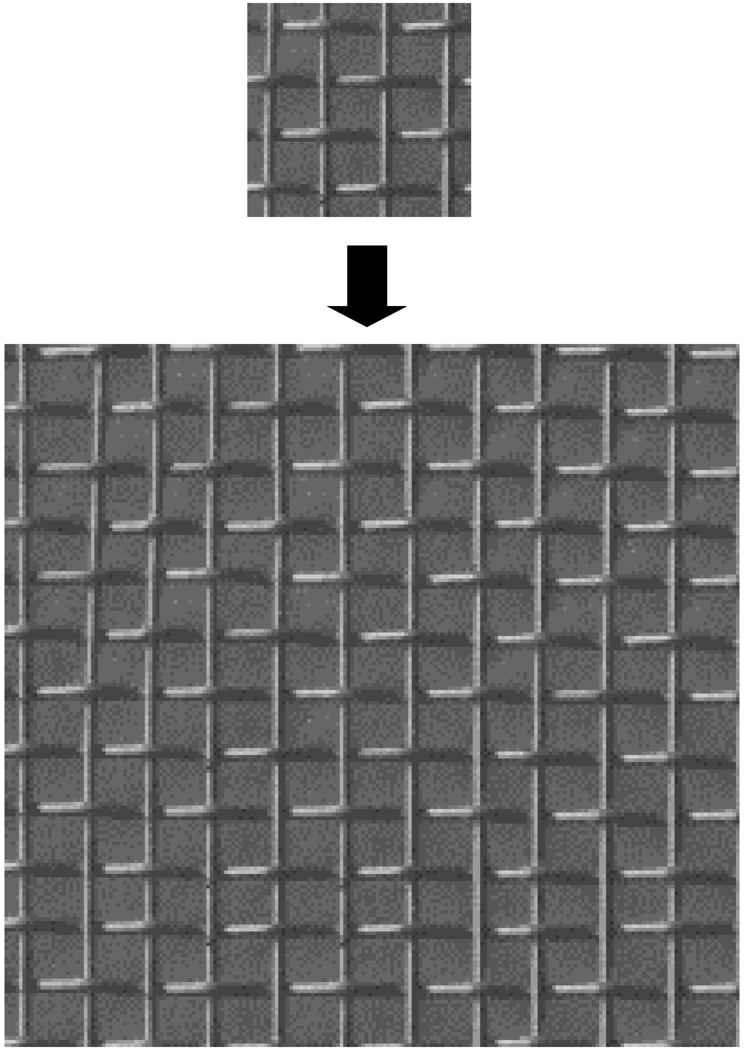
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Brodatz Results

reptile skin



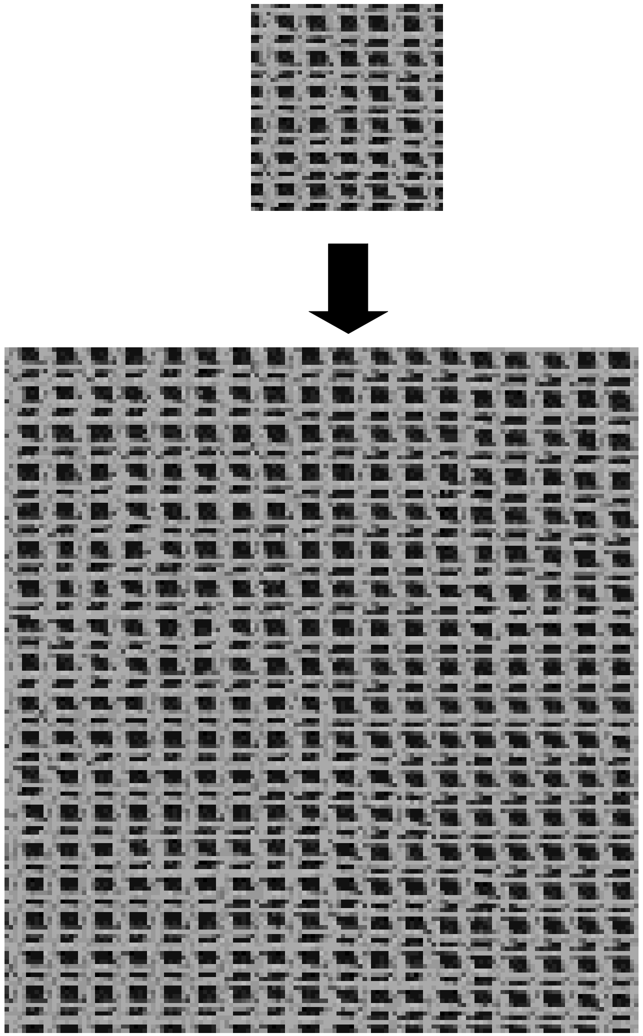
aluminum wire



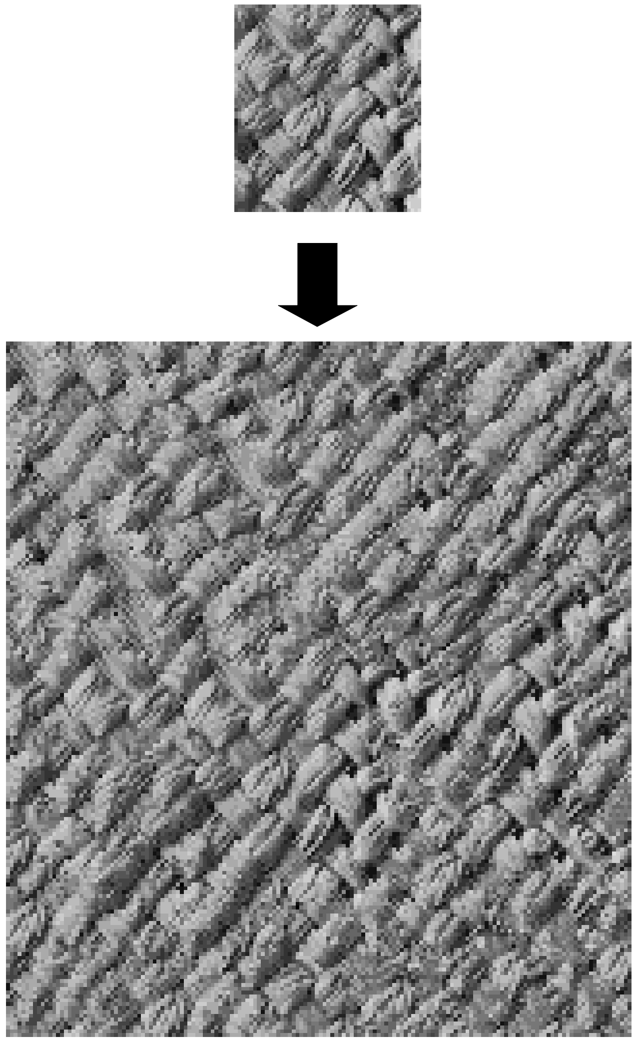
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

More Brodatz Results

french canvas



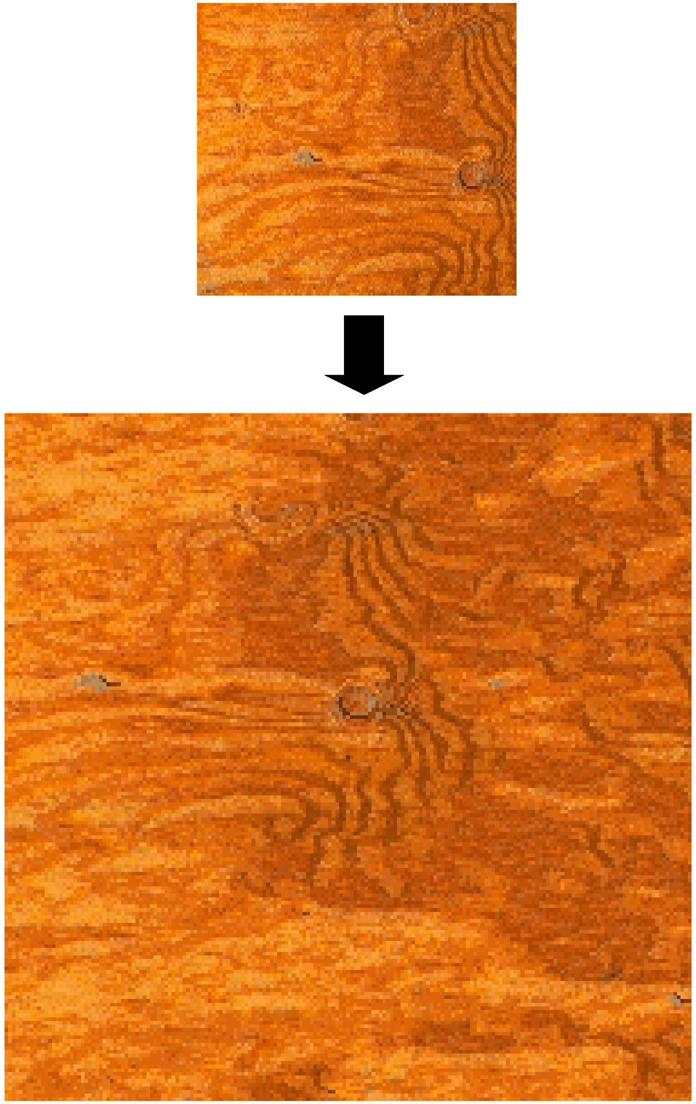
rafia weave



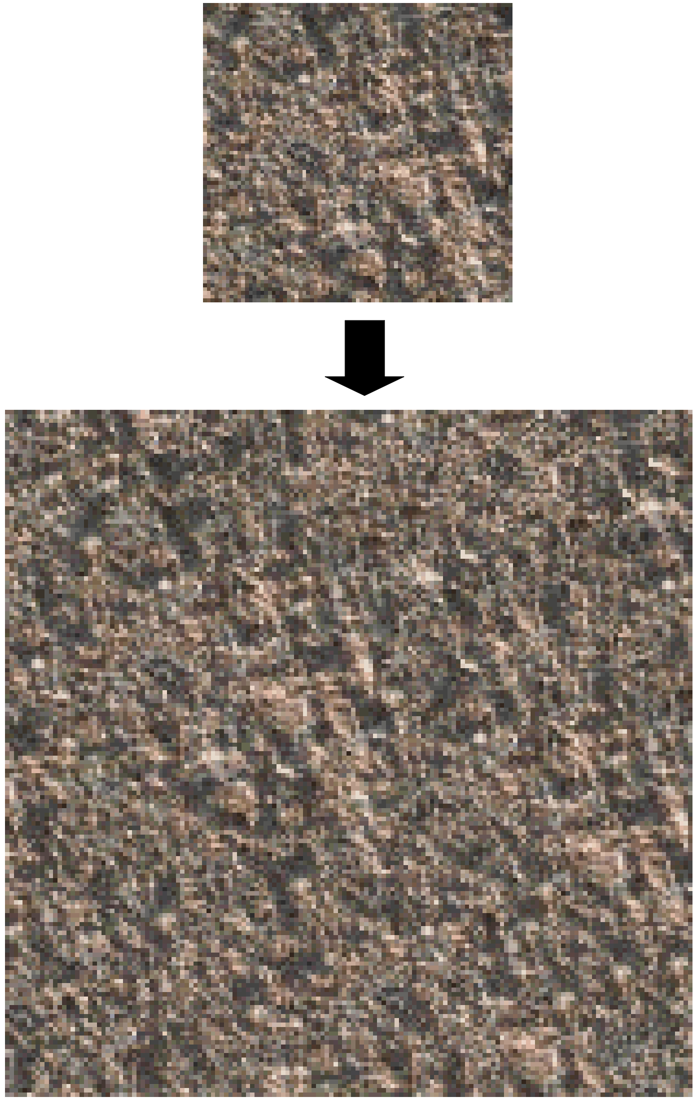
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

More Results

wood



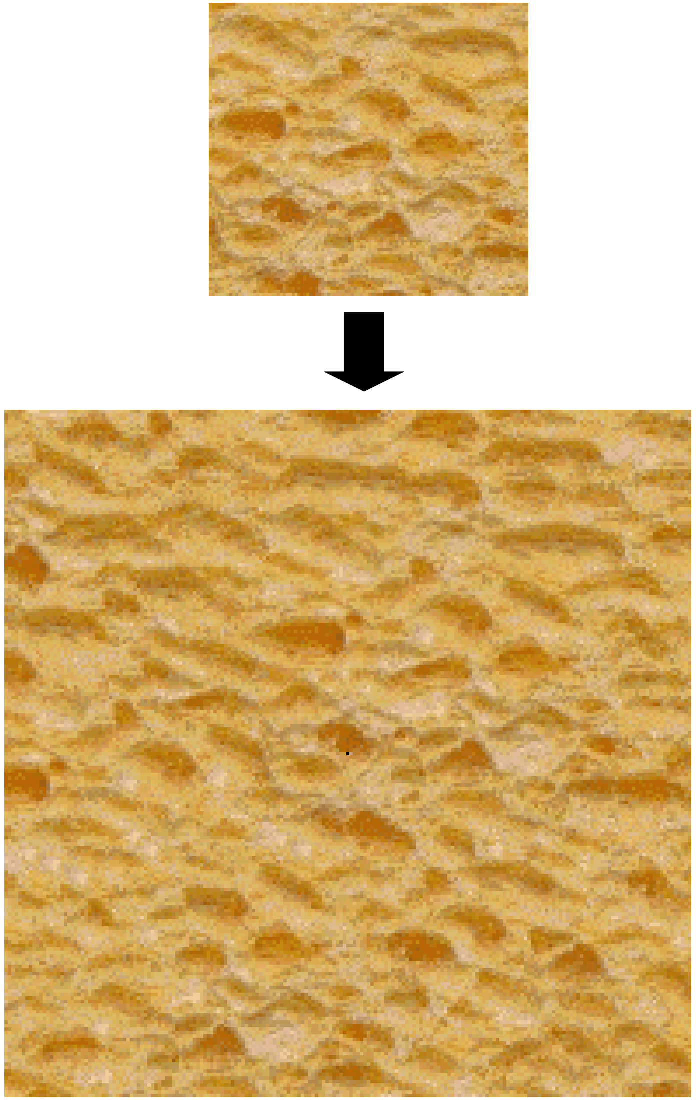
granite



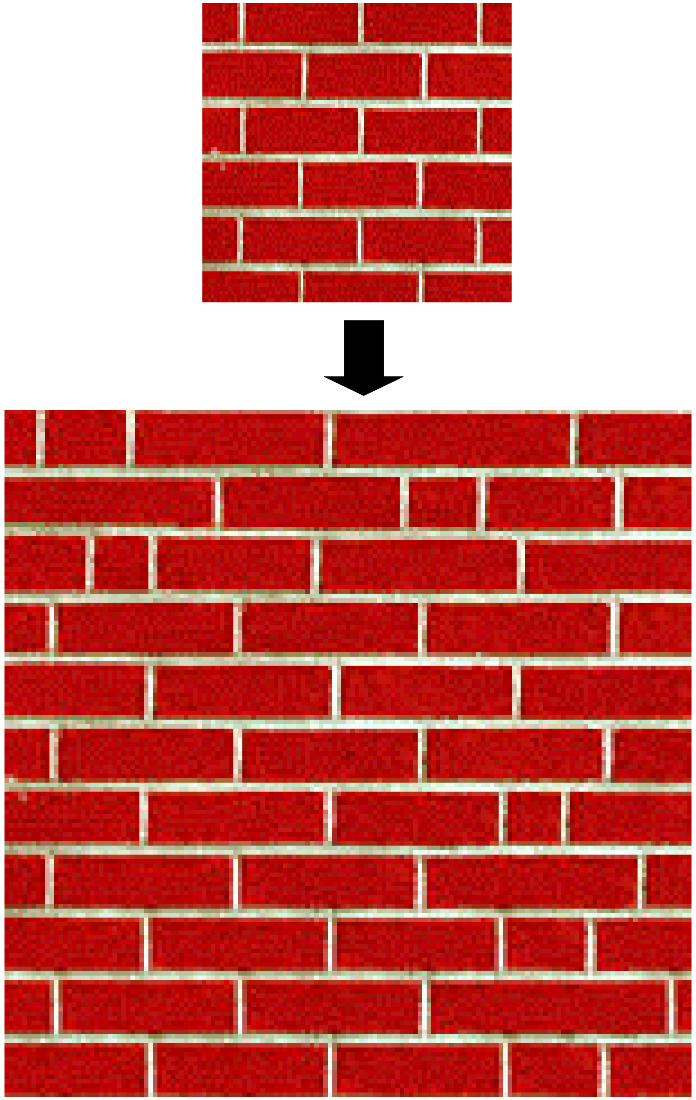
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

More Results

white bread

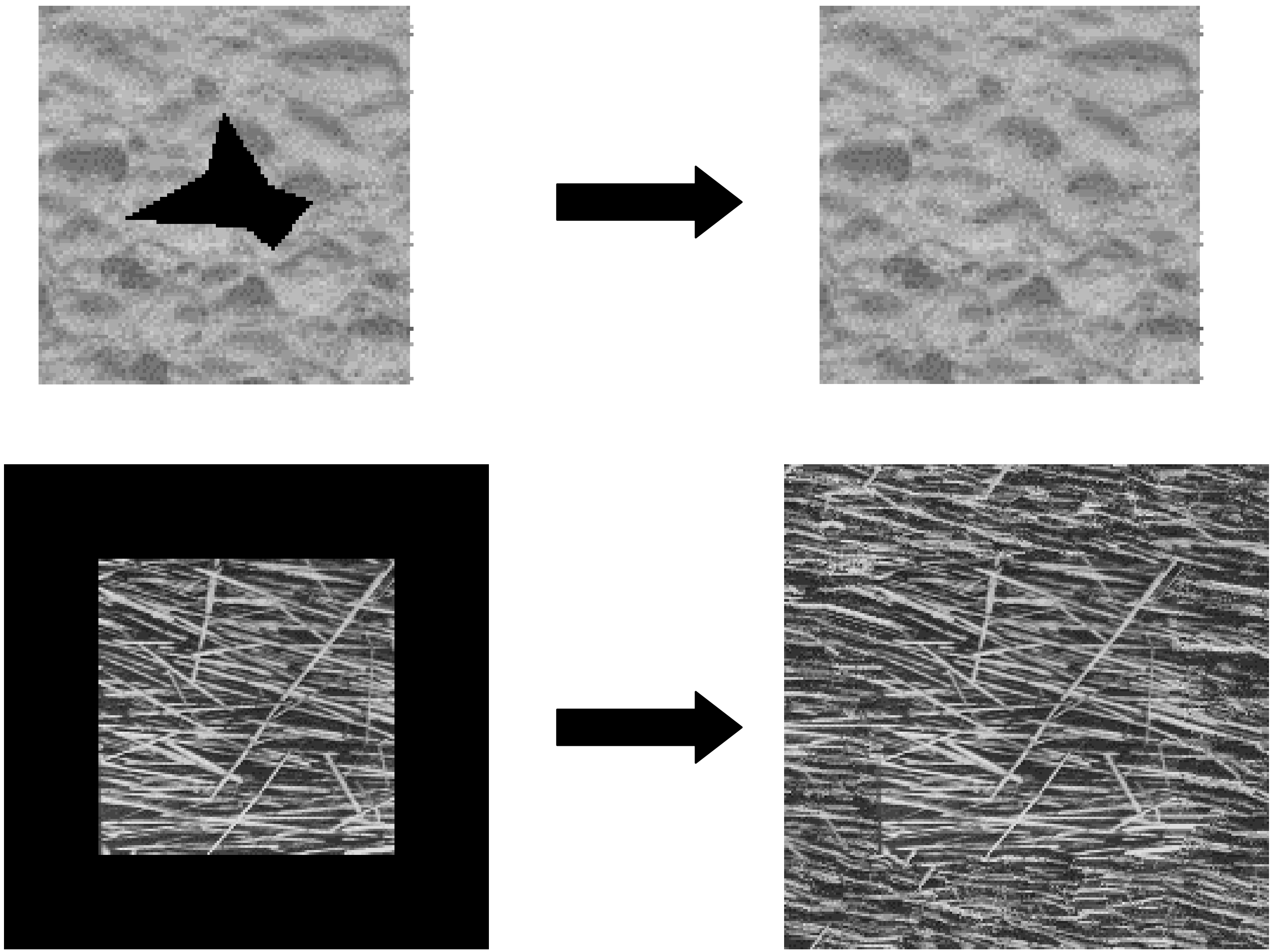


brick wall



http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

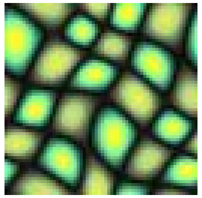
Constrained Synthesis

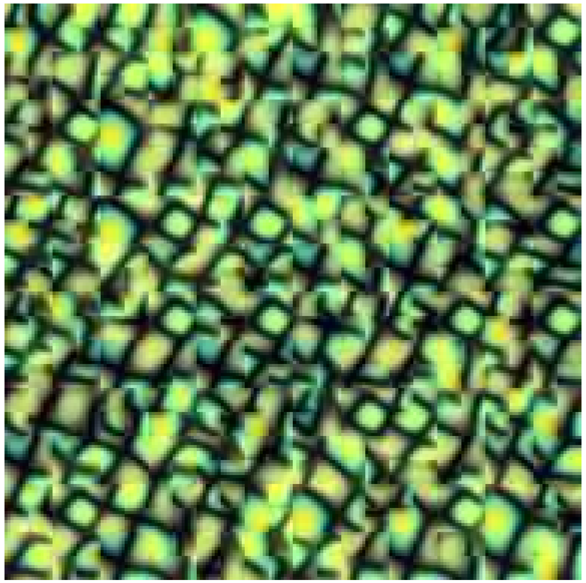


http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

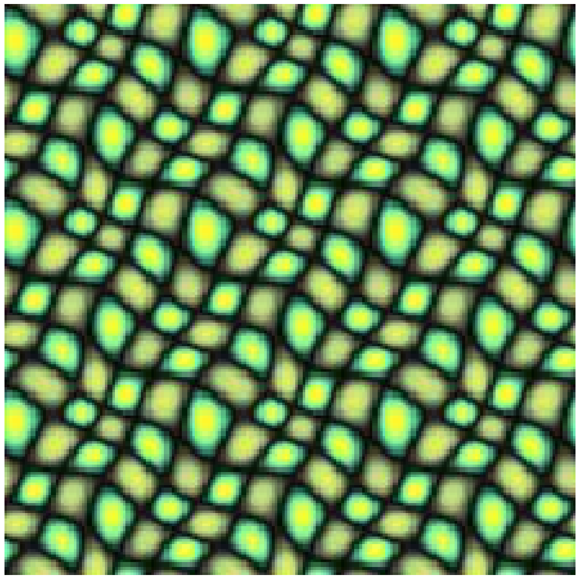
Visual Comparison

Synthetic tilable texture

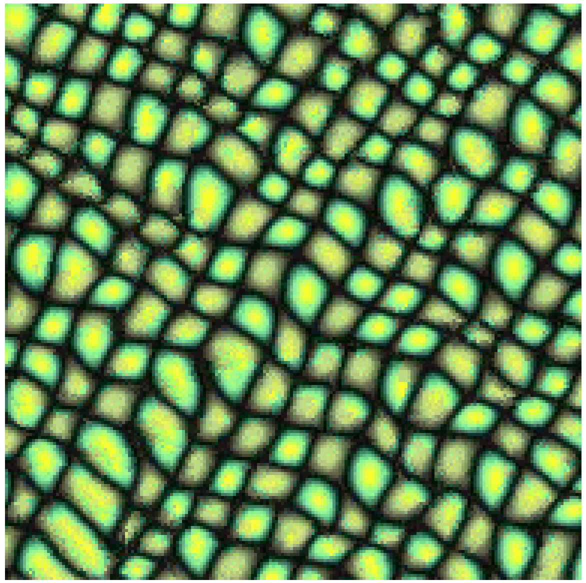




[DeBonet, '97]



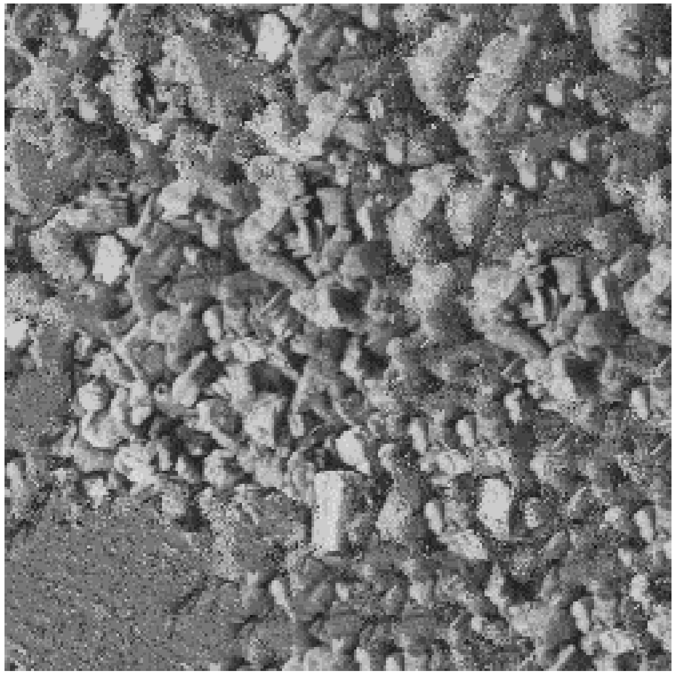
Simple tiling



Our approach

http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Failure Cases



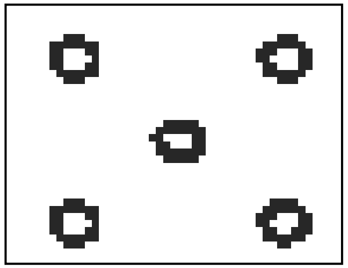
Growing garbage

Verbatim copying

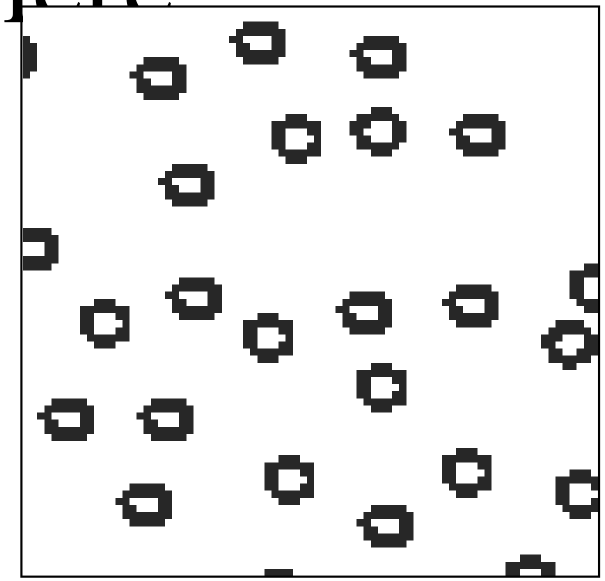
http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Texturing a sphere

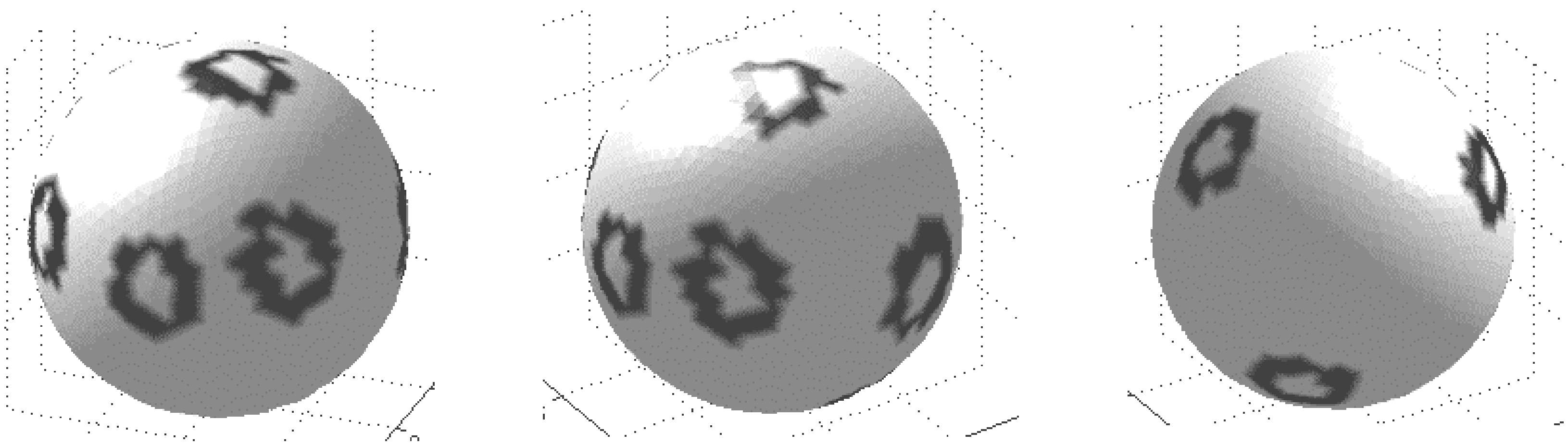
Sample image



2D

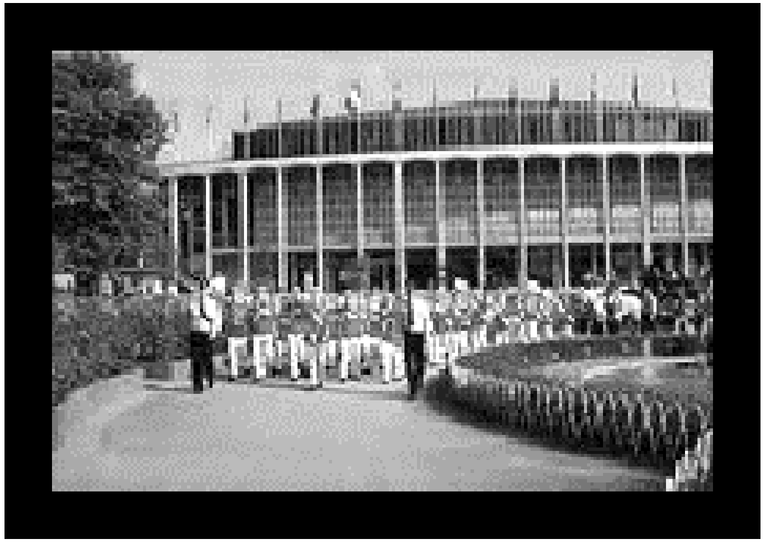


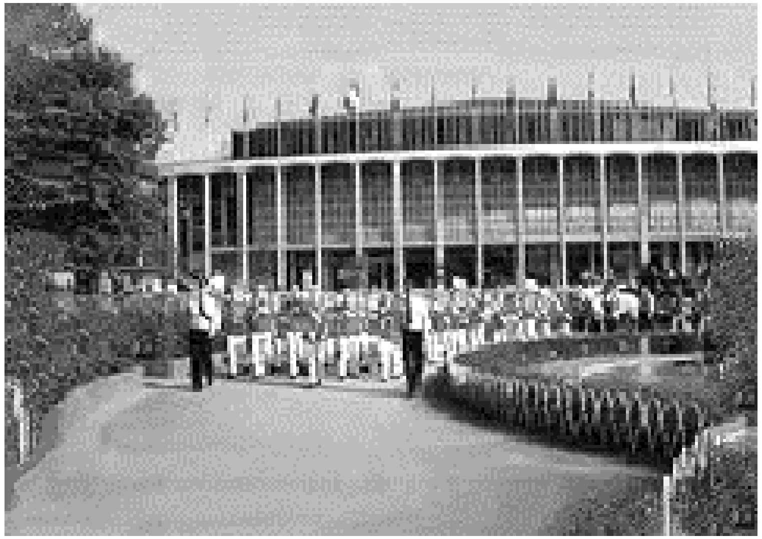
3D



http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

Image Extrapolation





http://www.cs.berkeley.edu/~efros/research/NPS/efros-iccv99.ppt

What we learned from Efros and Leung regarding texture synthesis

- Don’t need conditional filter responses across scale
- Don’t need marginal statistics of filter responses.
- Don’t need multi-scale, multi-orientation filters.
- Don’t need filters.

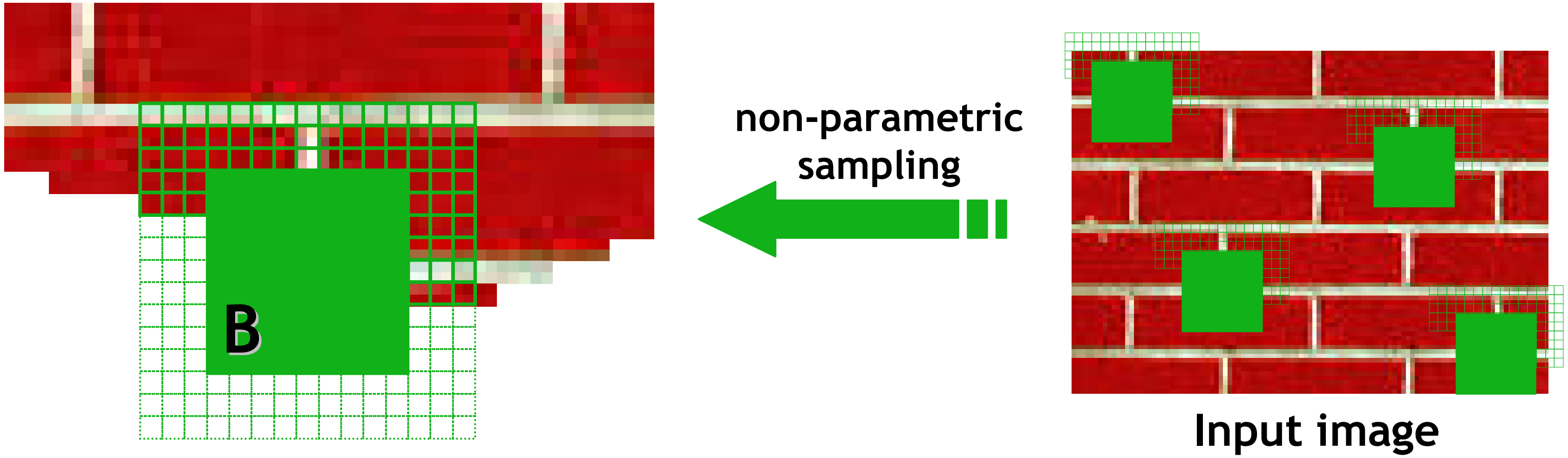
Efros & Leung

- The algorithm
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow
- Optimizations and Improvements
  - [Wei & Levoy,'00] (based on [Popat & Picard,'93])
  - [Harrison,'01]
  - [Ashikhmin,'01]

Quilting

- The “Corrupt Professor’s Algorithm” - Freeman:
  - Plagiarize as much of the source image as you can
  - Then try to cover up the evidence
- Rationale:
  - Texture blocks are by definition correct samples of texture so problem only connecting them together

Quilting: Efros & Freeman

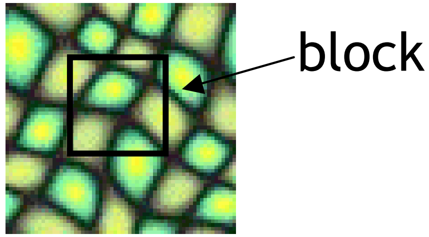


Synthesizing a block

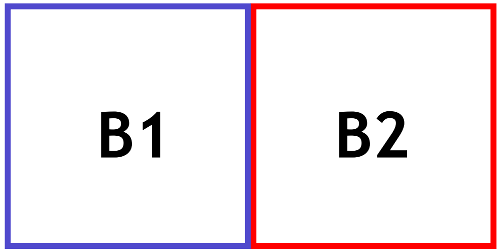
- Observation: neighbor pixels are highly correlated

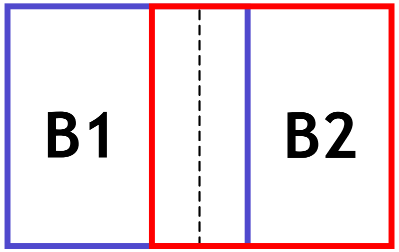
Idea: unit of synthesis = block

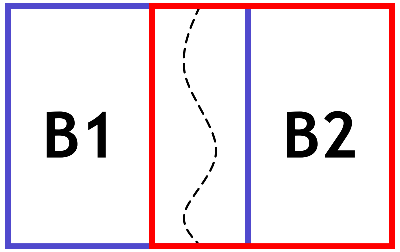
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!



Input texture



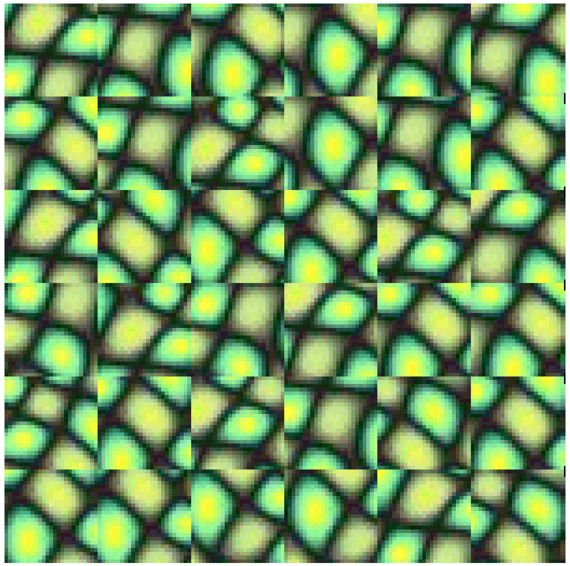


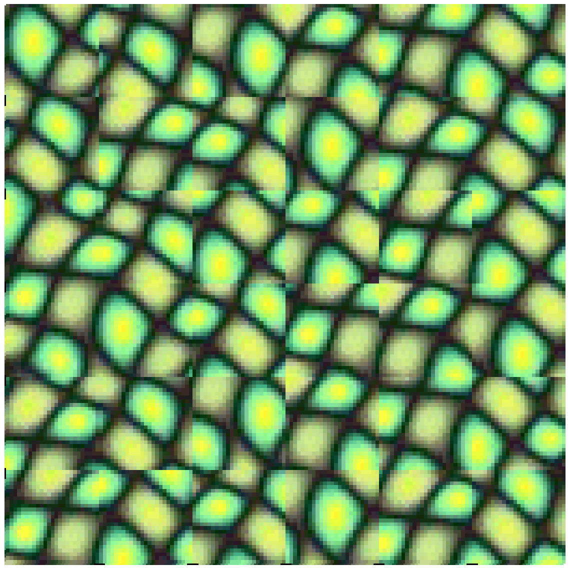


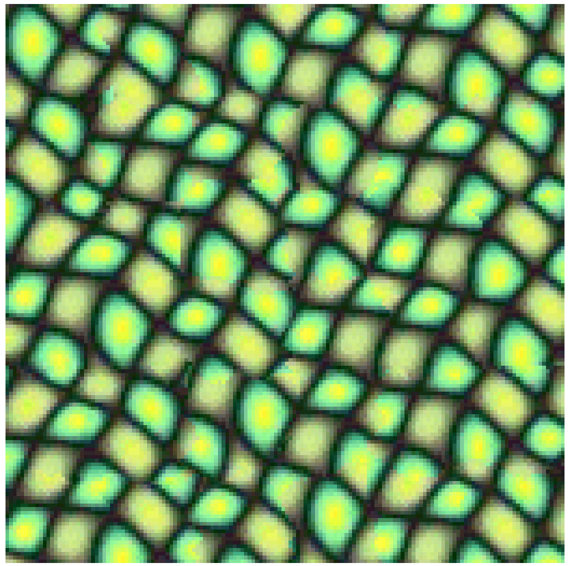
Random placement of blocks

Neighboring blocks constrained by overlap

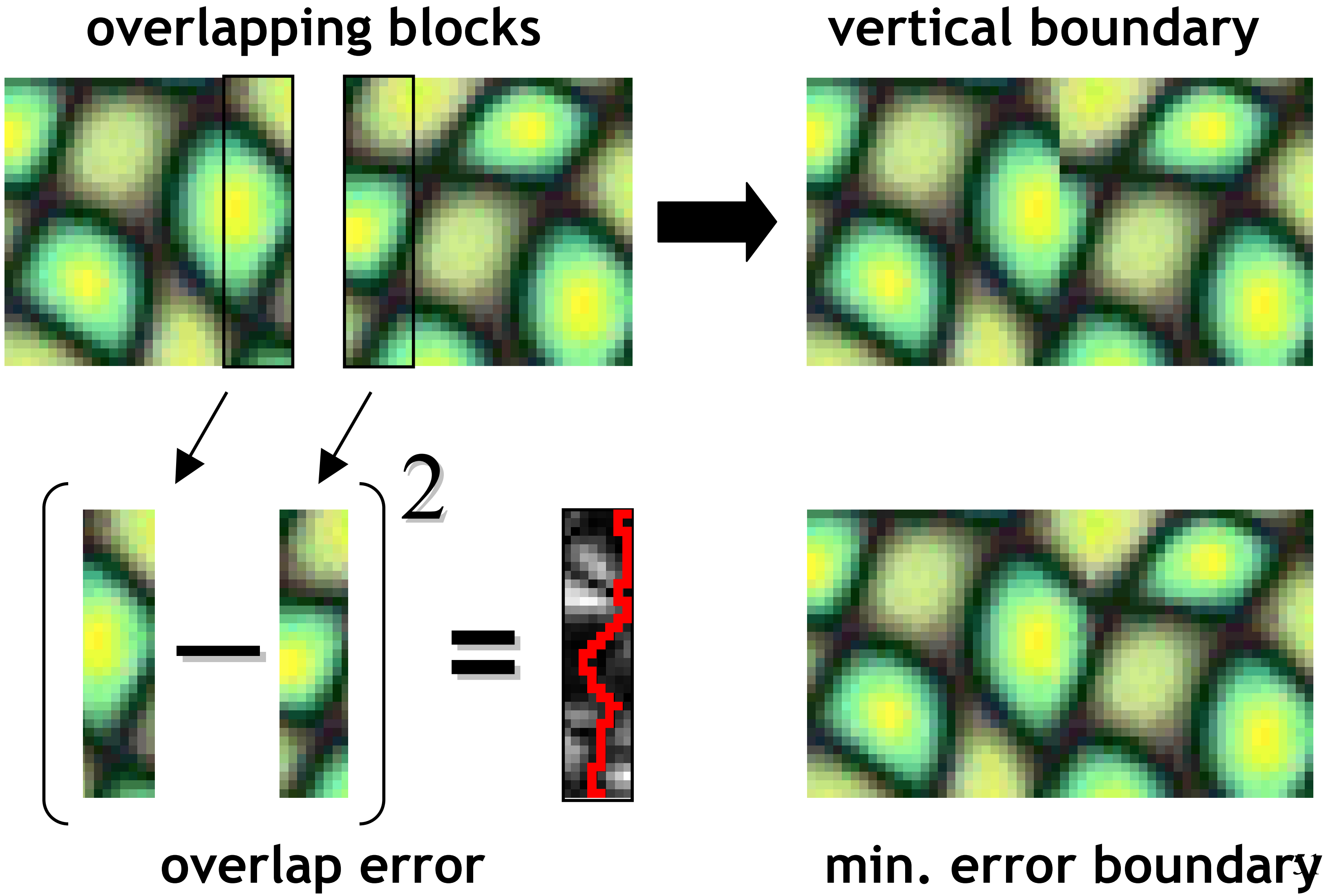
Minimal error boundary cut







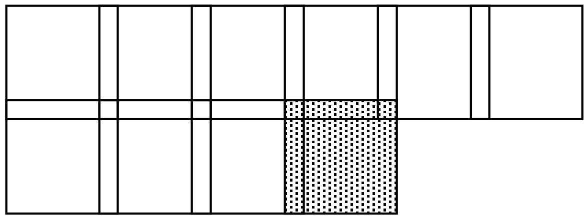
Minimal error boundary



Algorithm

– Pick size of block and size of overlap

– Synthesize blocks in raster order

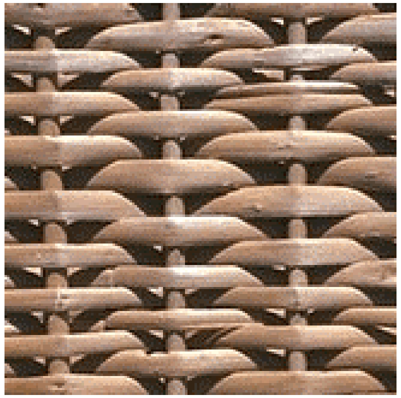


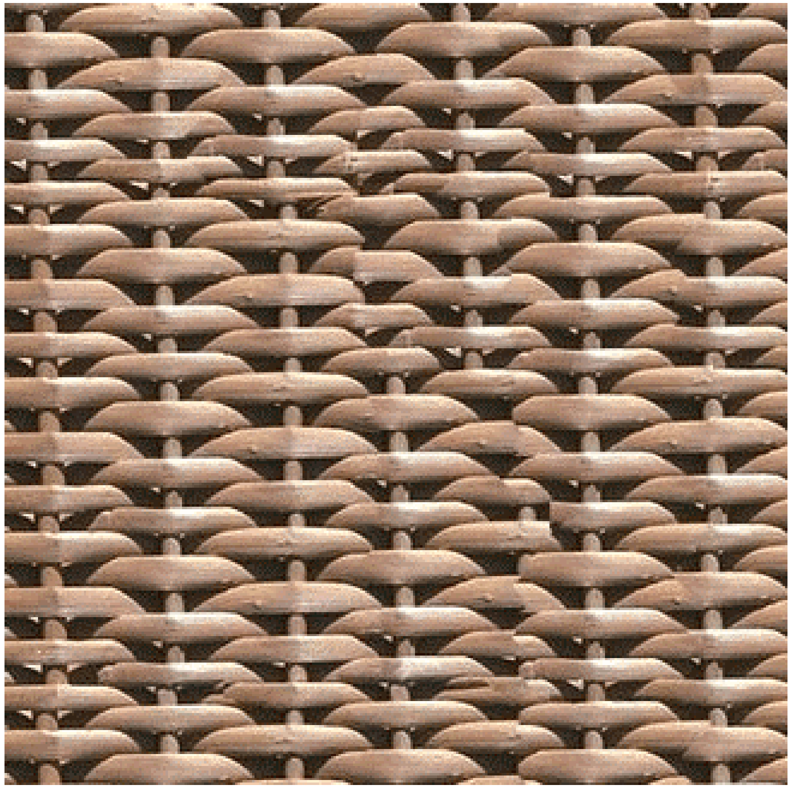
– Search input texture for block that satisfies overlap constraints (above and left)

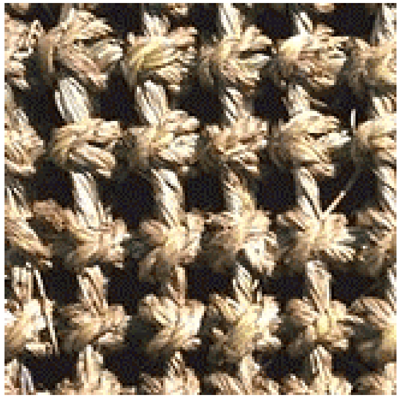
  - Easy to optimize using NN search [Liang et.al., ’01]

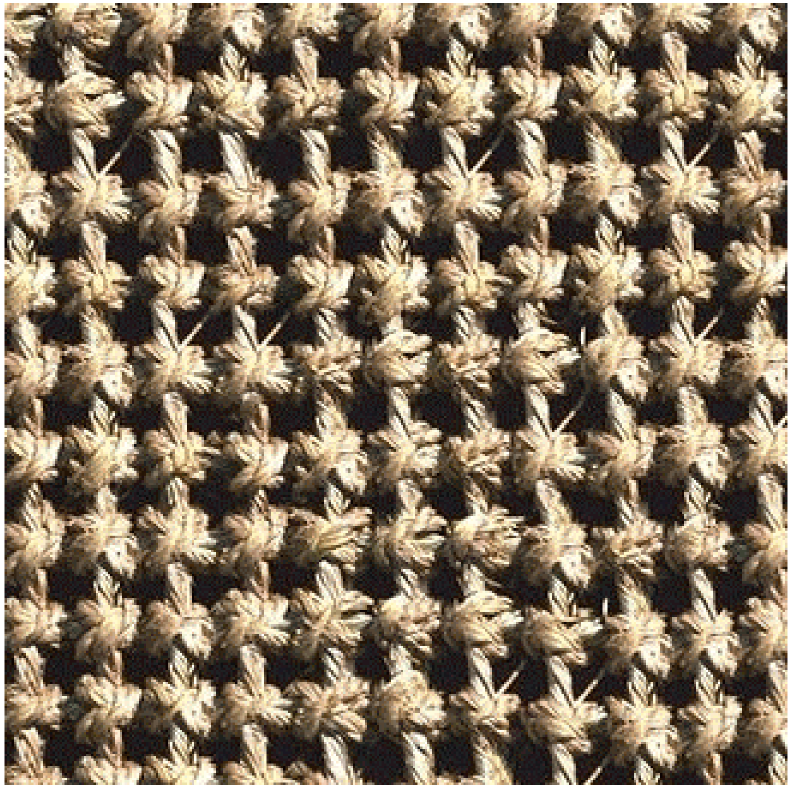
– Paste new block into resulting texture

  - use dynamic programming to compute minimal error boundary cut

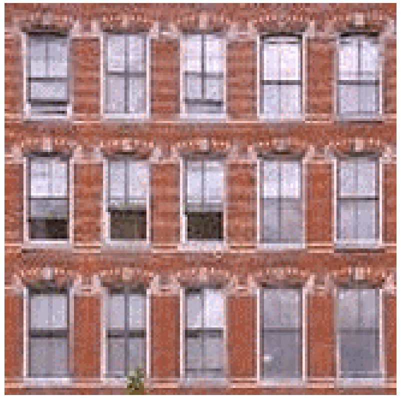




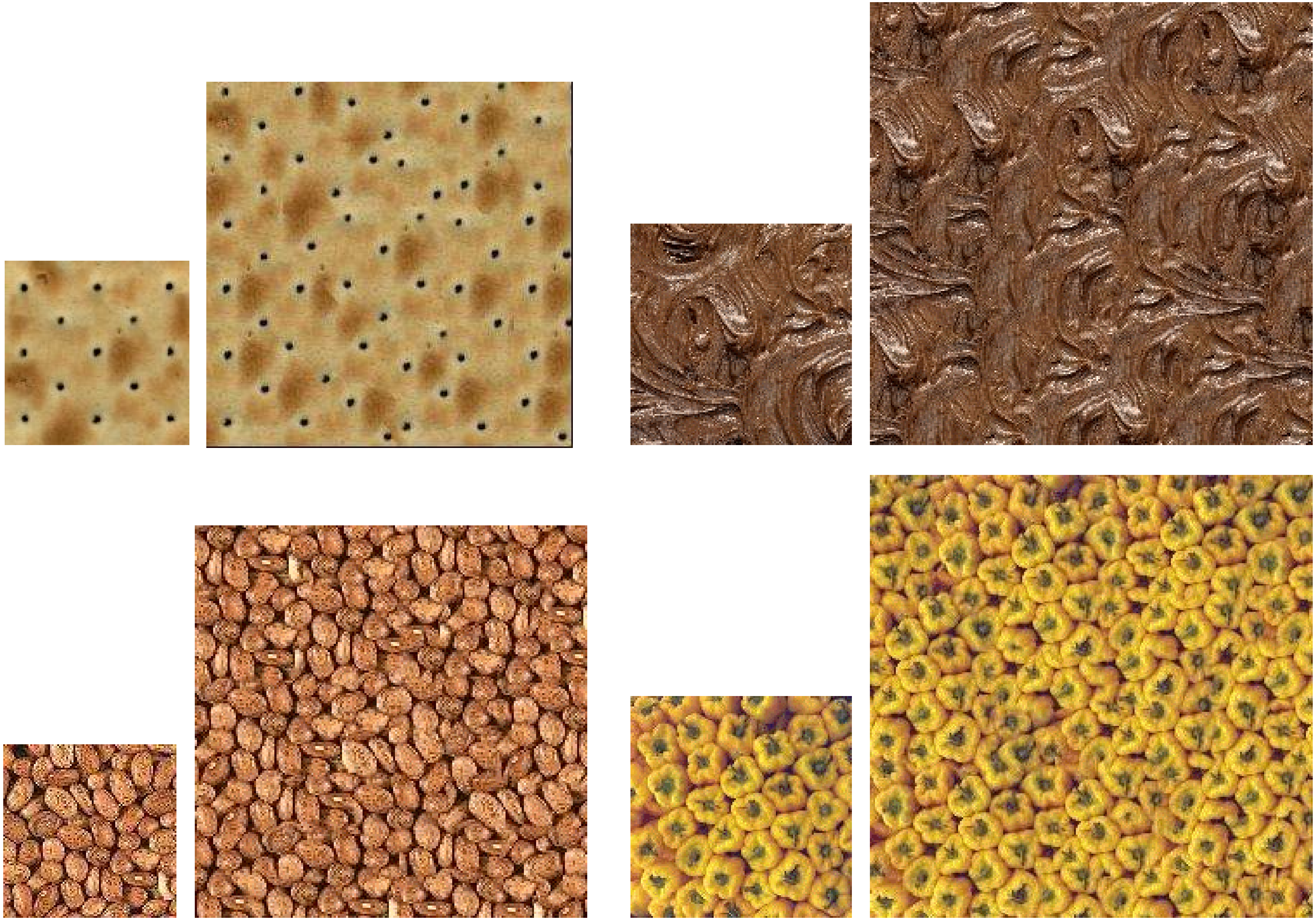


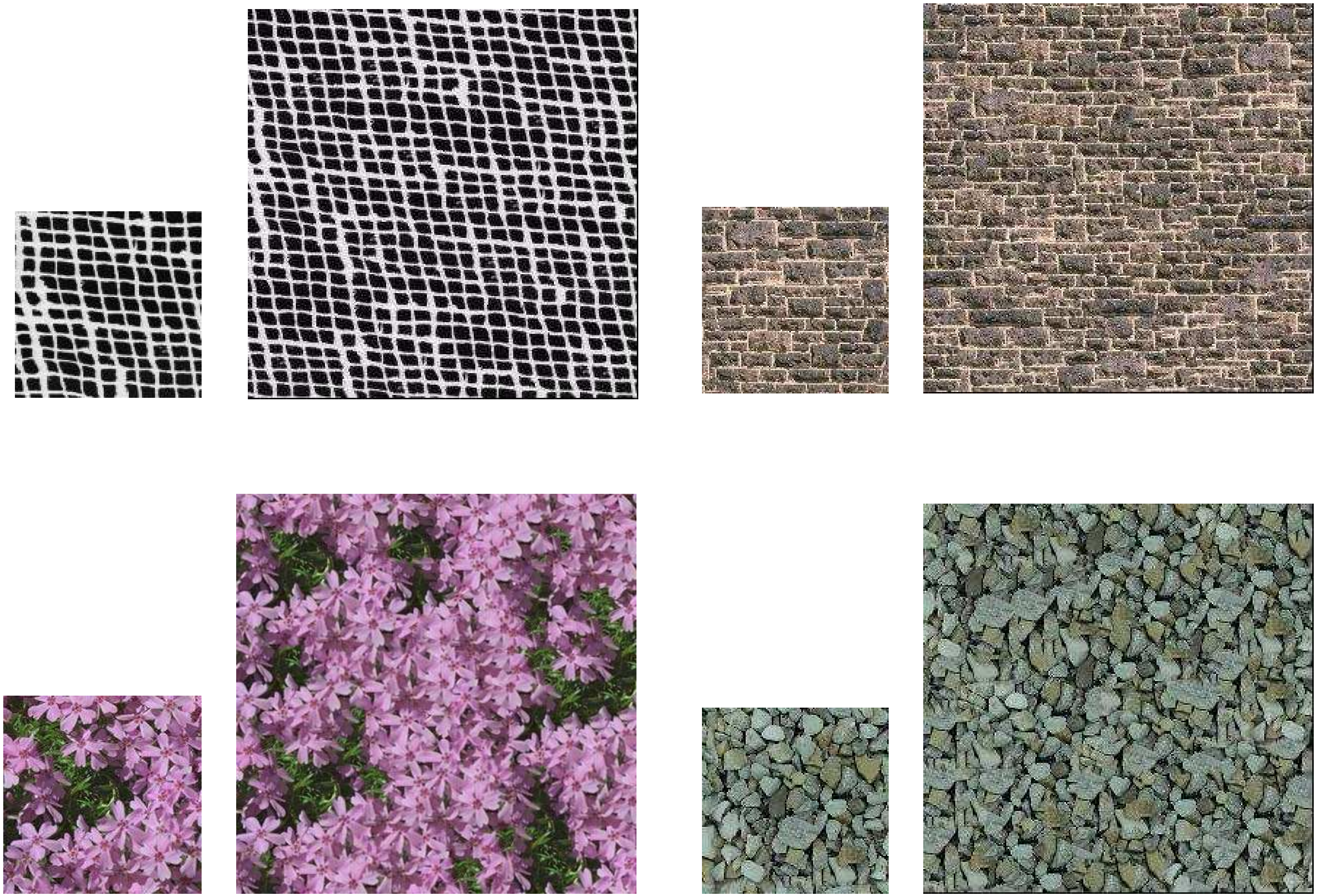




College Inn


College Inn

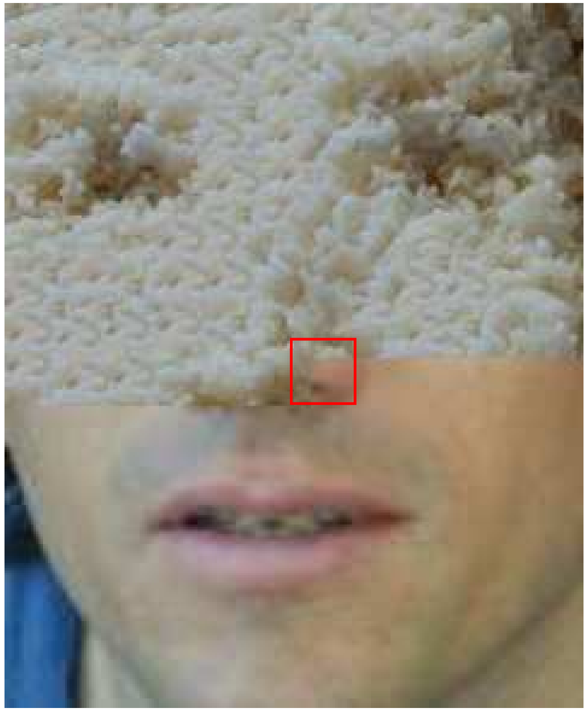


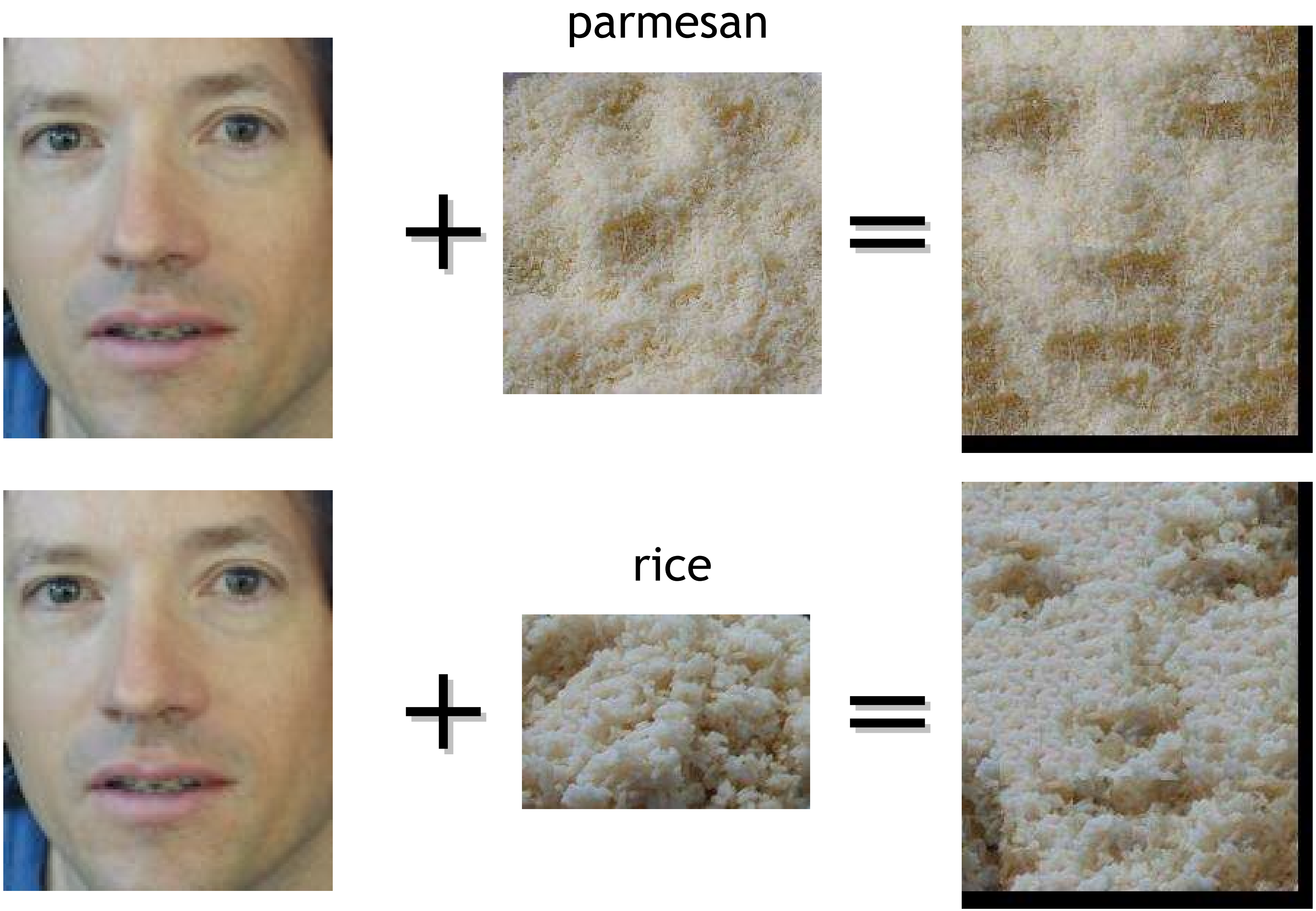
Failures
(Chernobyl Harvest)

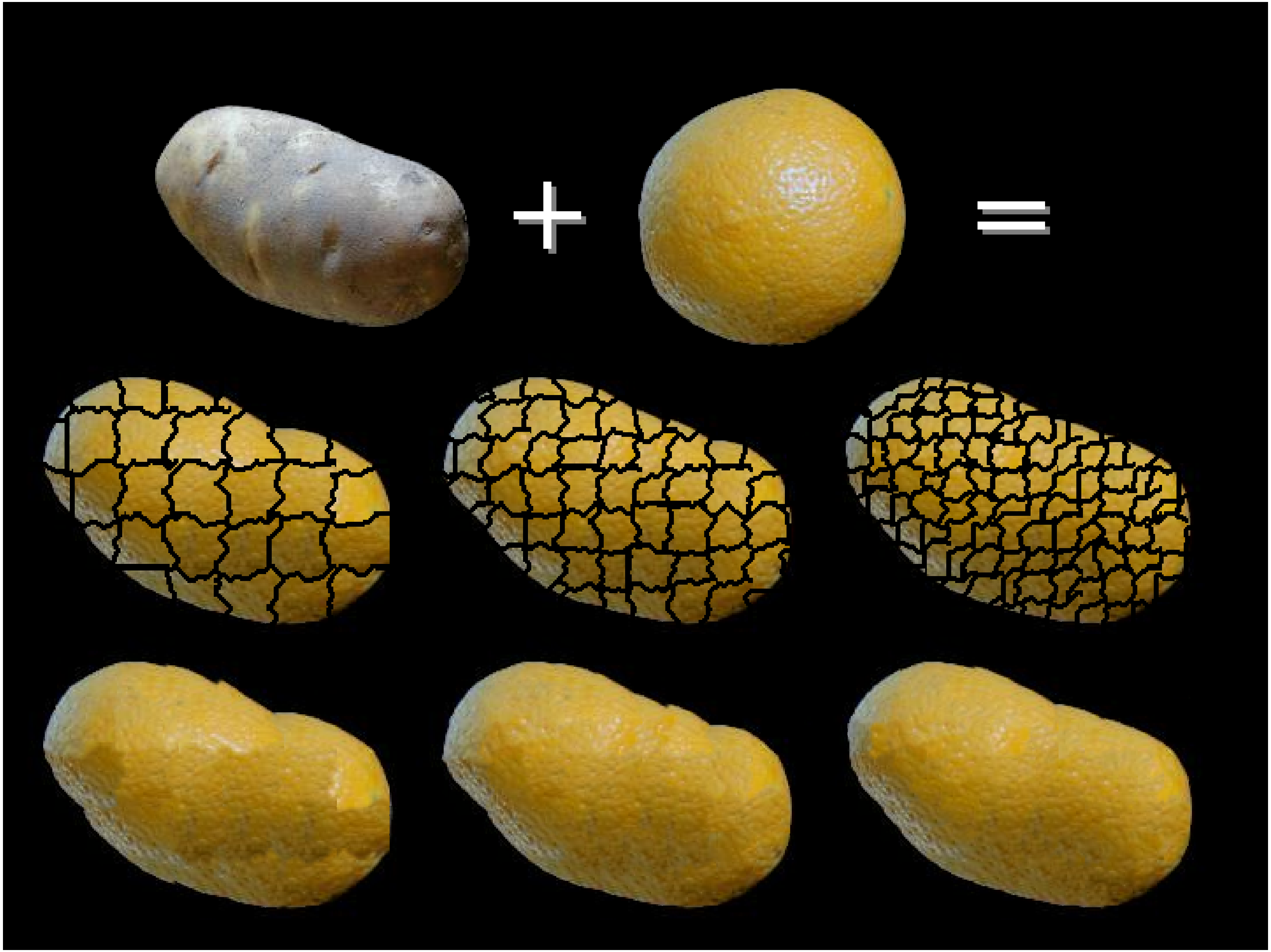
Texture Transfer

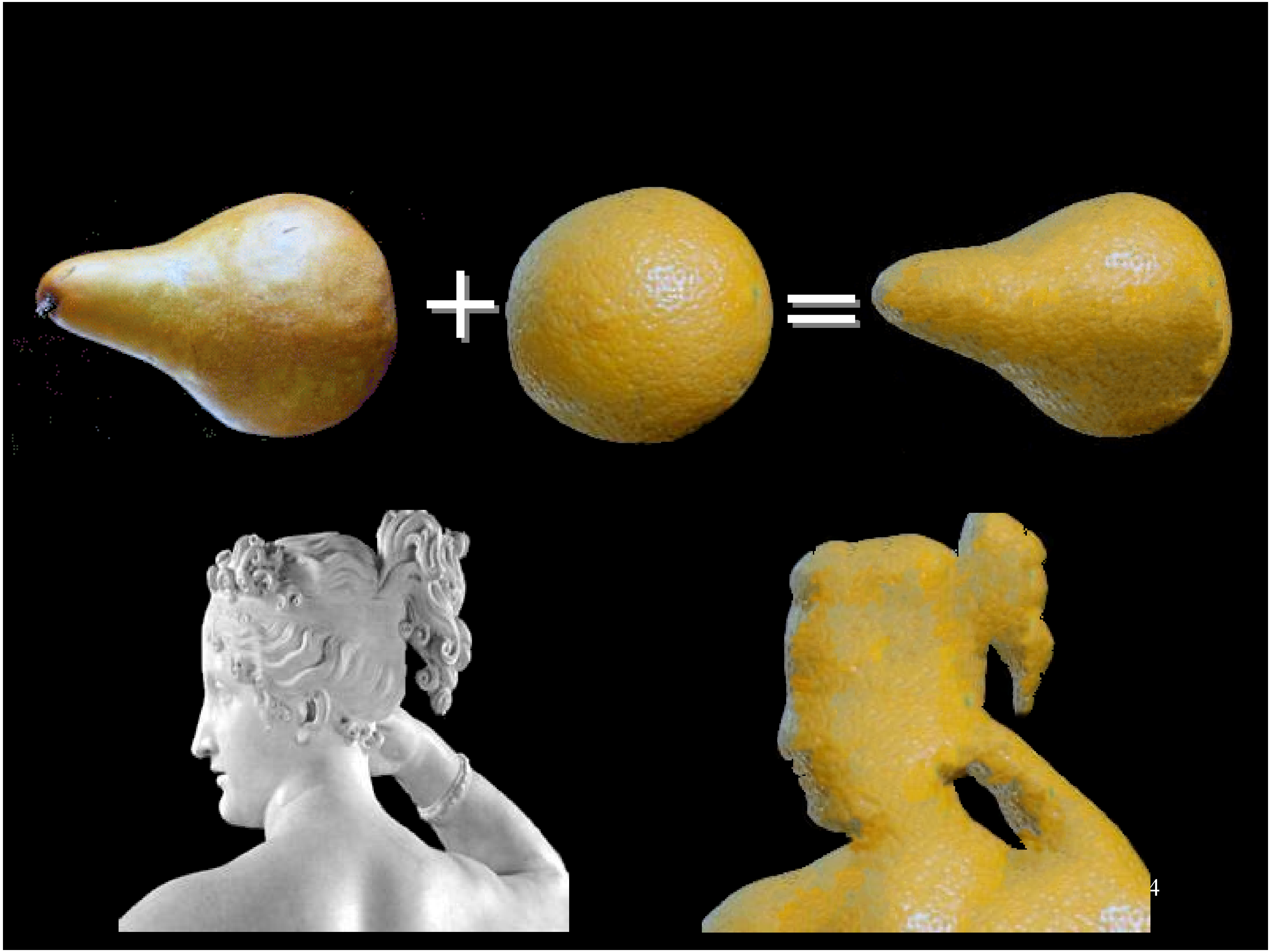
- Take the texture from one object and "paint" it onto another object
  - This requires separating texture and shape
  - That's HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading
-

Then, just add another constraint when sampling: similarity to underlying image at that spot

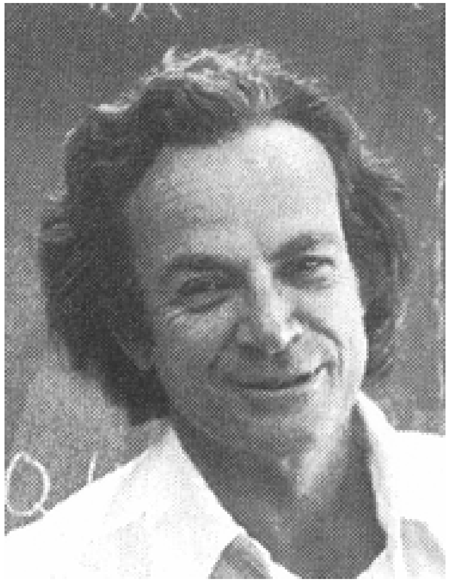



parmesan
rice




4

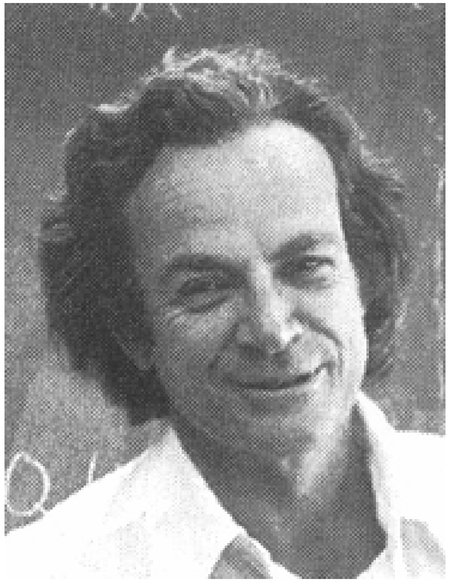
Source texture



Target image

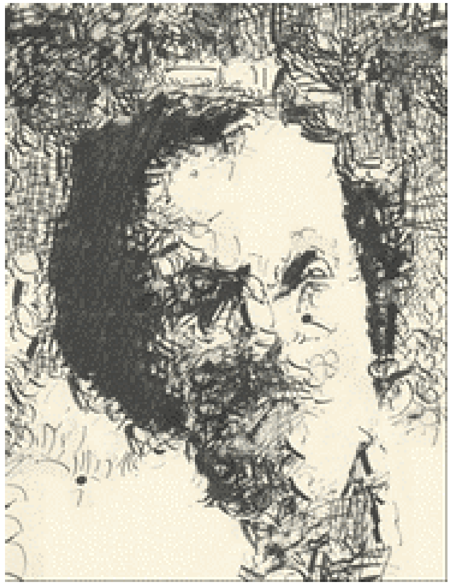
Source correspondence image

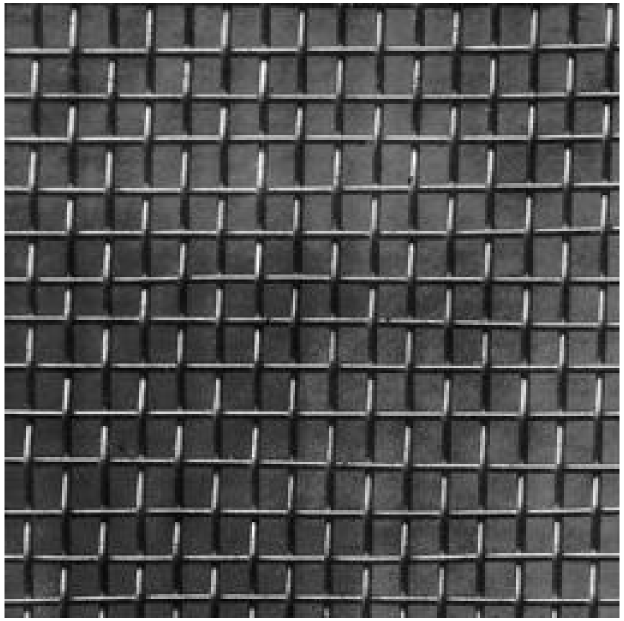
Target correspondence image



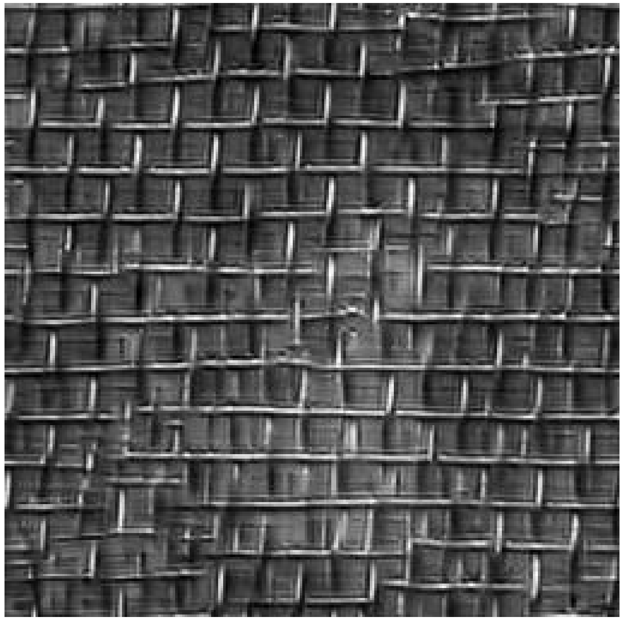
+

=

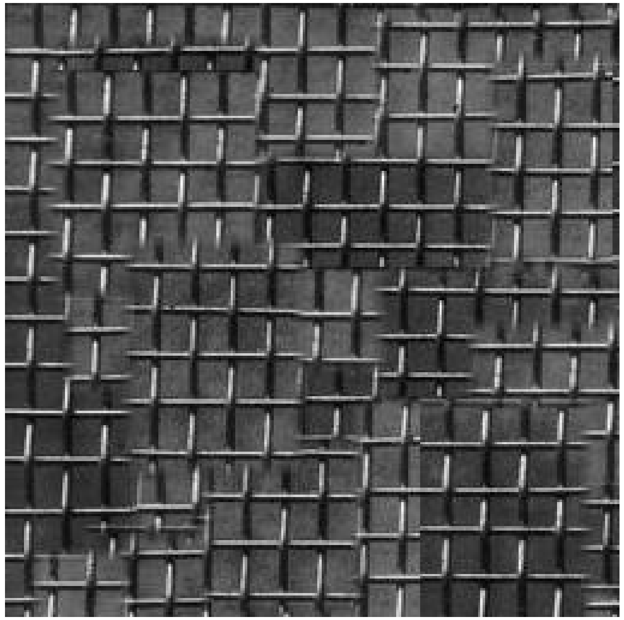




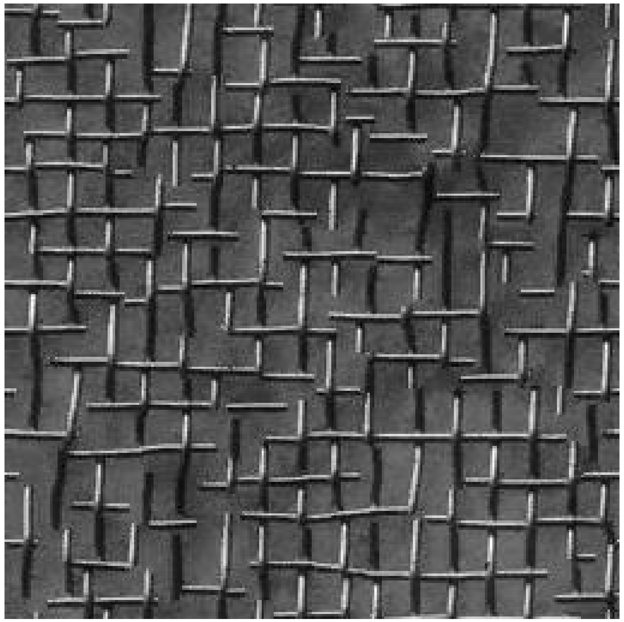
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

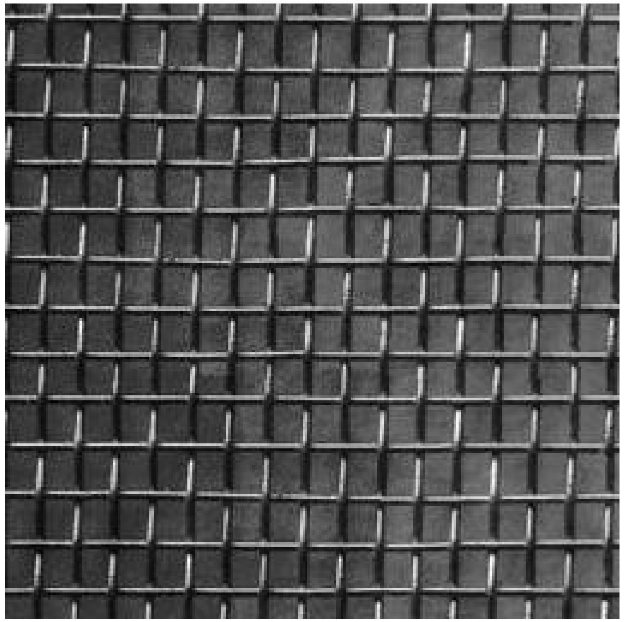
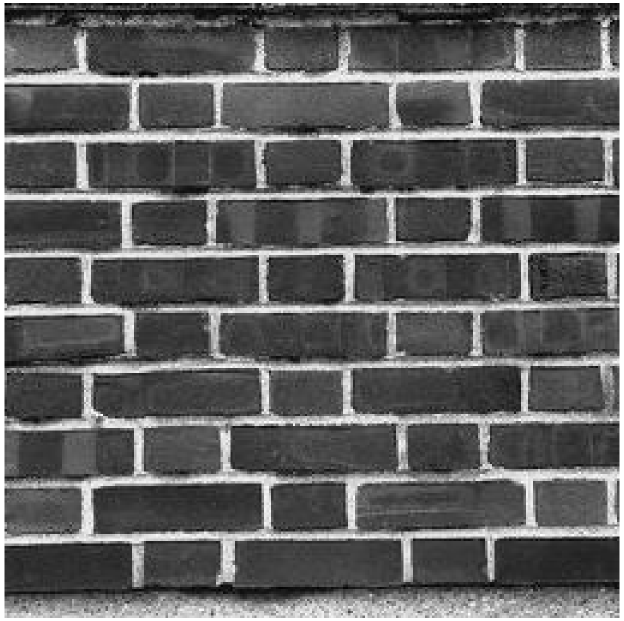


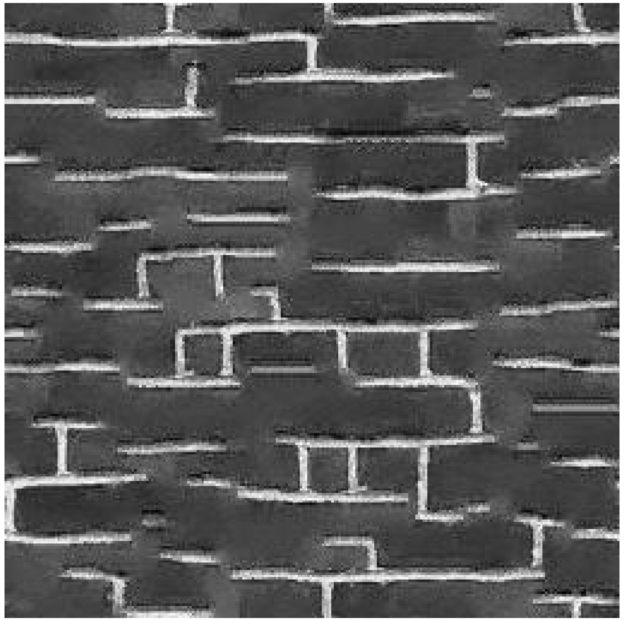
Image Quilting



input image

Portilla & Simoncelli

Xu, Guo & Shum



Wei & Levoy

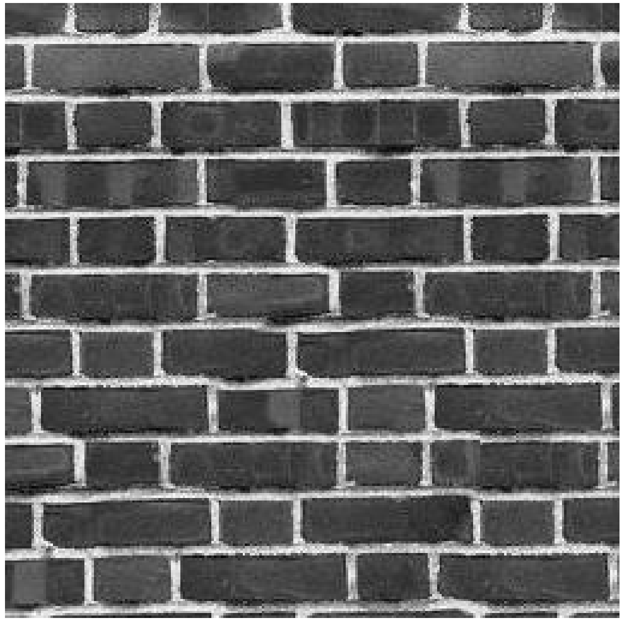
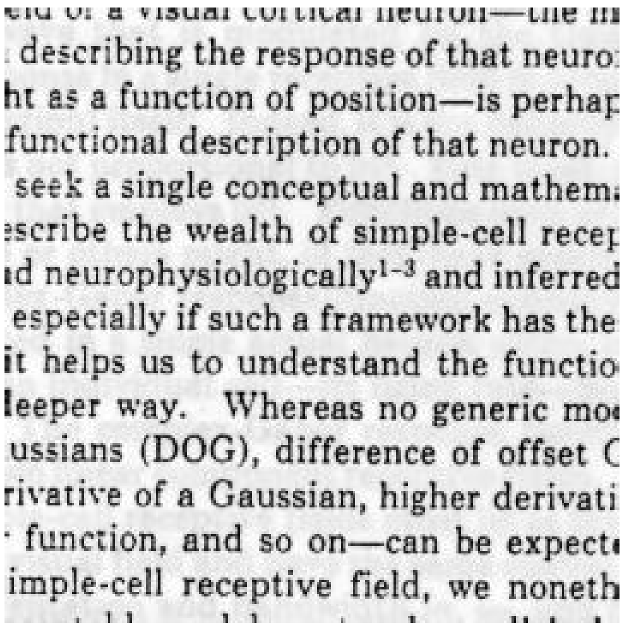
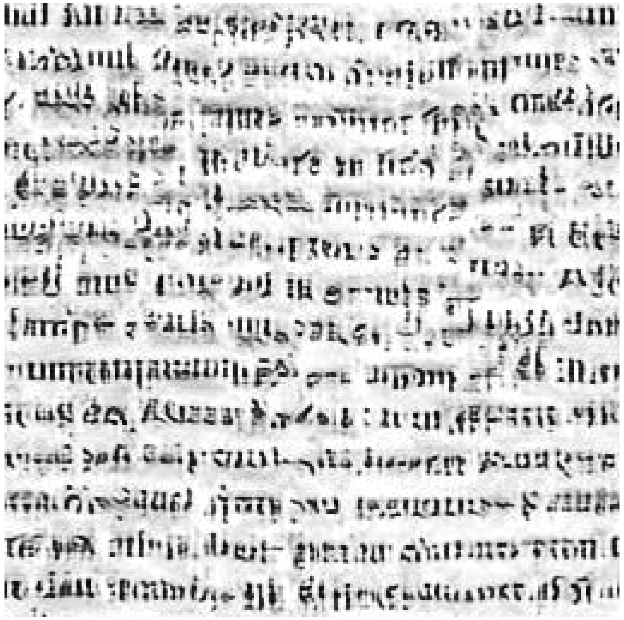


Image Quilting

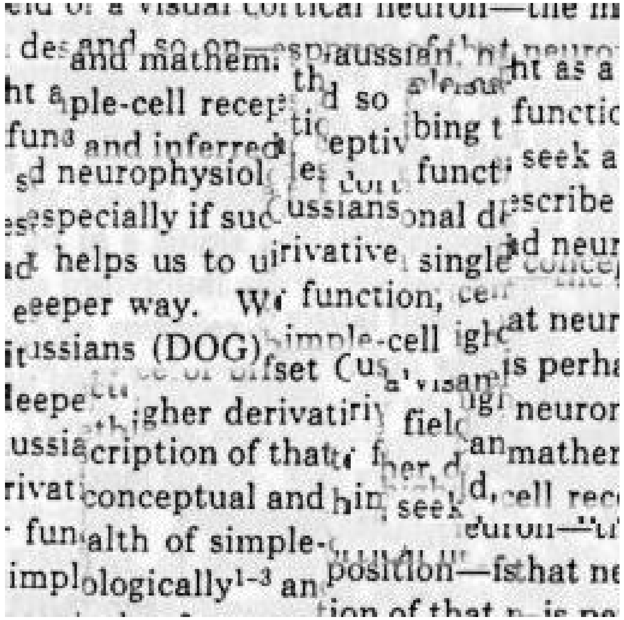
Homage to Shannon!



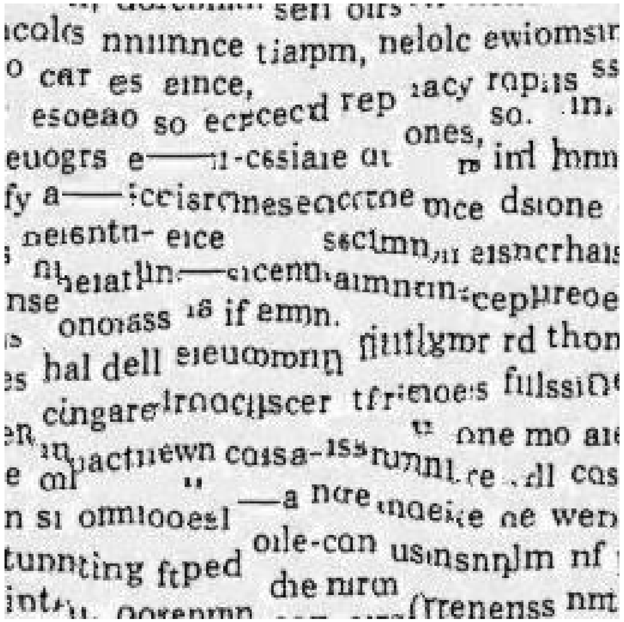
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

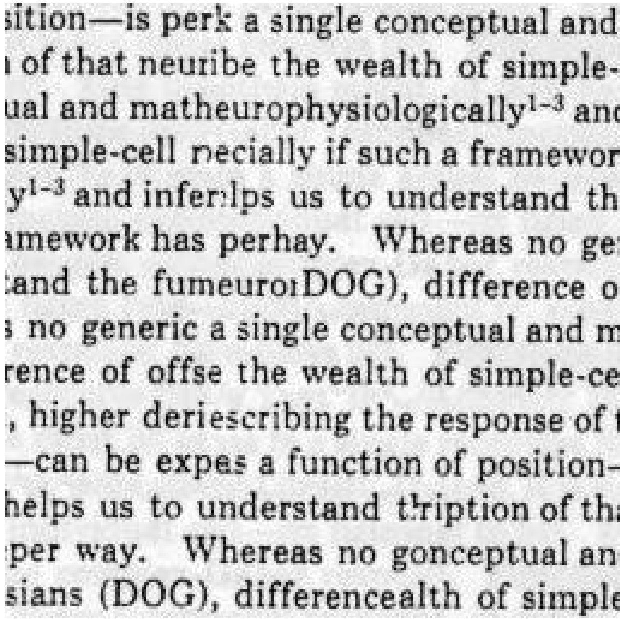


Image Quilting

Summary of image quilting

- Quilt together patches of input image
  - randomly (texture synthesis)
  - constrained (texture transfer)
- Image Quilting
  - No filters, no multi-scale, no one-pixel-at-a-time!
  - fast and very simple
  - Results are not bad

Example-based model

A set of image patches

Input image

Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

Compressed example-based model

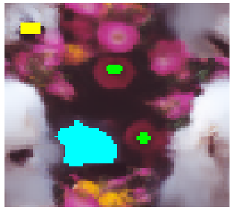
A set of image patches

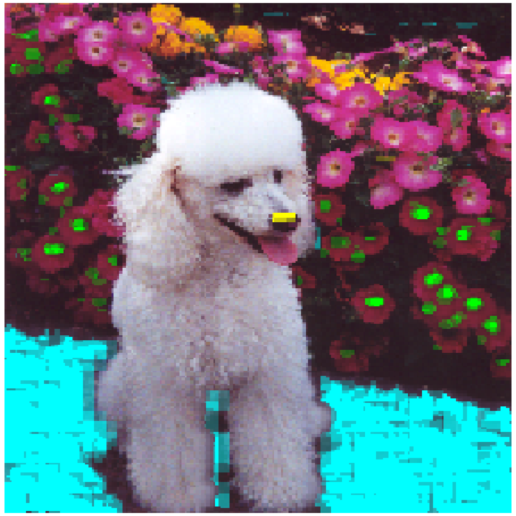
Input image

Epitome

Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

Compact representation

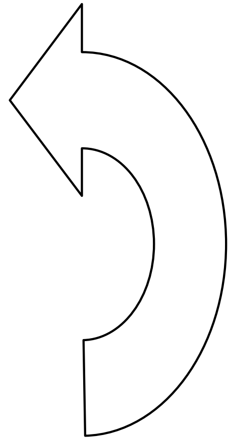


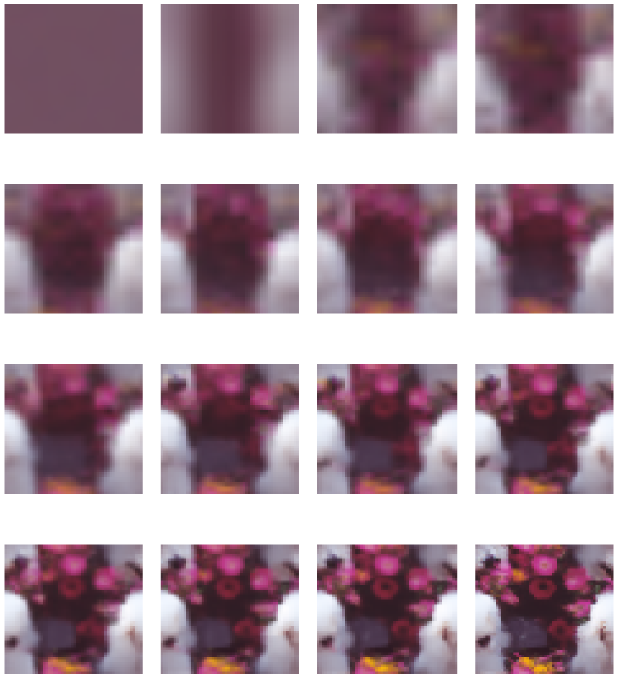


Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

Learning the epitome

- For each patch, infer the posterior over the mappings
- Average all patches using the posterior as a weight
- Estimate the variance

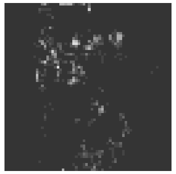




Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

More examples

mean



variance

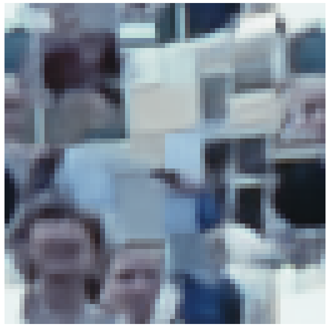
Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003

www.research.microsoft.com/~jojic/epitome.htm

More examples

Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

More examples



Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

What is epitome good for?

- A better way to learn a library of patches (for SR, texture synthesis and analysis, …)
- A tool for easy editing
- Organizing visual memory for recognition
- An alternative both to templates and low-order statistics (e.g., histograms) in vision systems

Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003
www.research.microsoft.com/~jojic/epitome.htm

Denoising

Original image

SNR=13dB

Noisy image

SNR=18.4dB

Reconstruction using a mixture of 1000 patches learned from the noisy image

SNR=19.2dB

Reconstruction using an 80x80 epitome

(in both cases, the patch size was 8x8)

Nebojsa Jojic, Brendan Frey and Anitha Kannan, ICCV 2003

www.research.microsoft.com/~jojic/epitome.htm