

C280, Computer Vision

Prof. Trevor Darrell

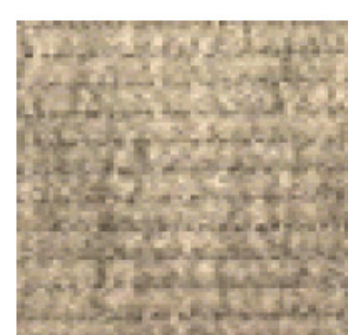
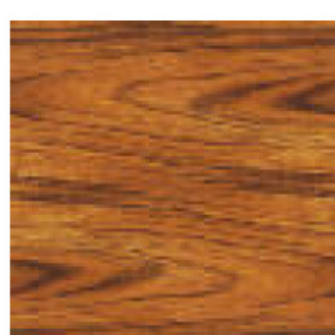
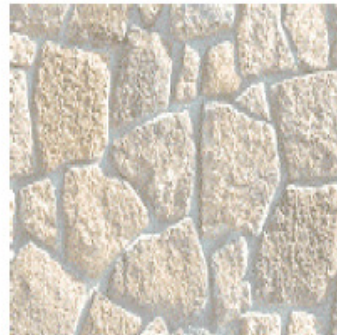
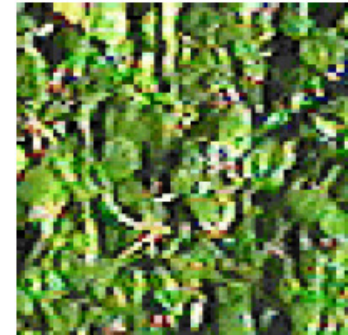
trevor@eecs.berkeley.edu

Lecture 7: Texture

Last Time: Feature Detection and Matching

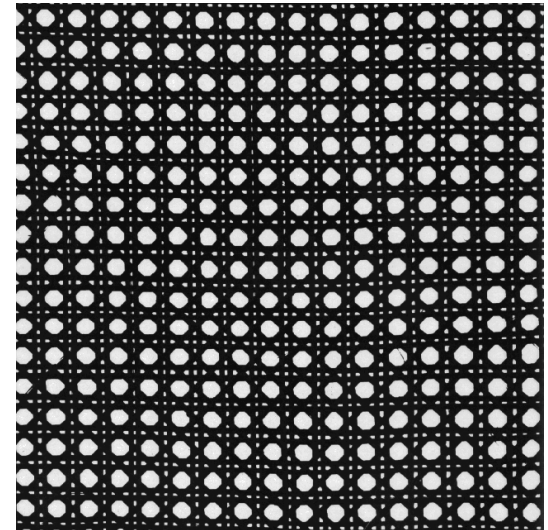
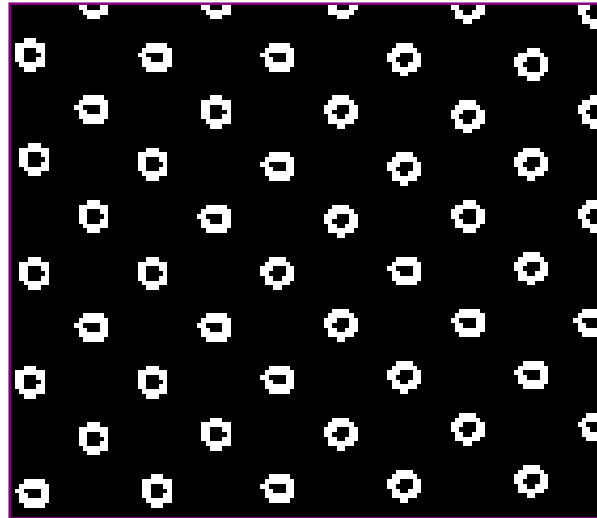
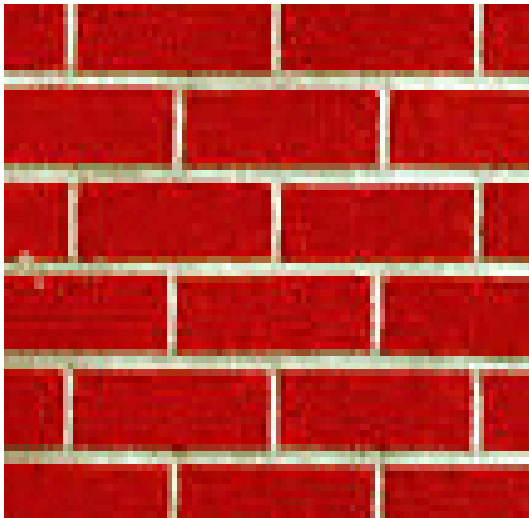
- Local features
- Pyramids for invariant feature detection
- Invariant descriptors
- Matching

Today: Texture

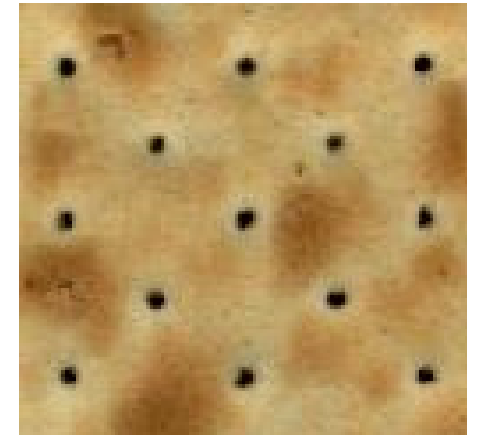


What defines a texture?

Includes: more regular patterns



Includes: more random patterns



Scale: objects vs. texture



Often the same thing in the world can occur as texture or an object, depending on the scale we are considering.

Why analyze texture?

Importance to perception:

- Often indicative of a material's properties
- Can be important appearance cue, especially if shape is similar across objects
- Aim to distinguish between shape, boundaries, and texture

Technically:

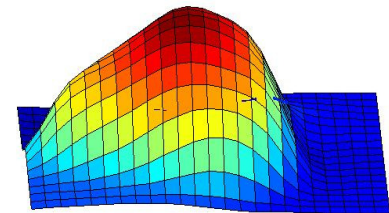
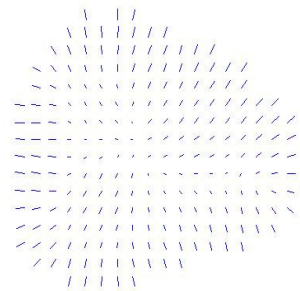
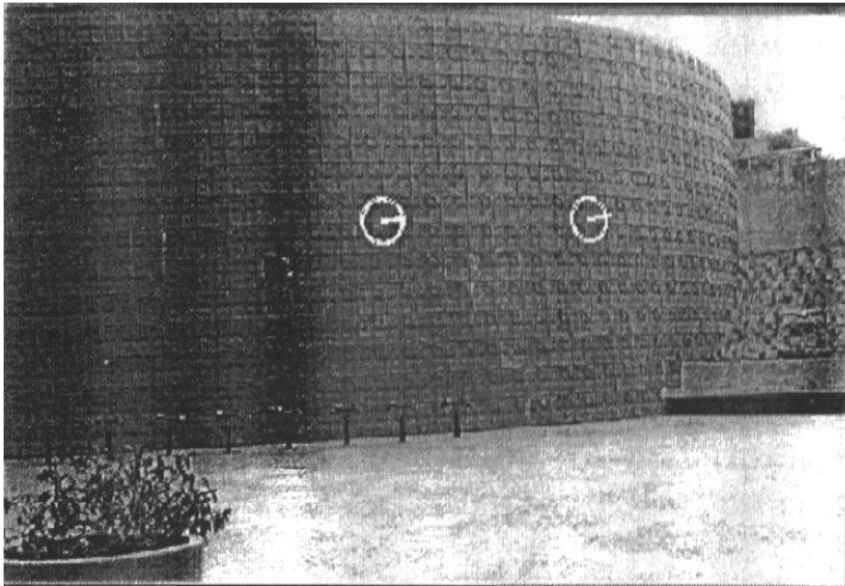
- Representation-wise, we want a feature one step above “building blocks” of filters, edges.

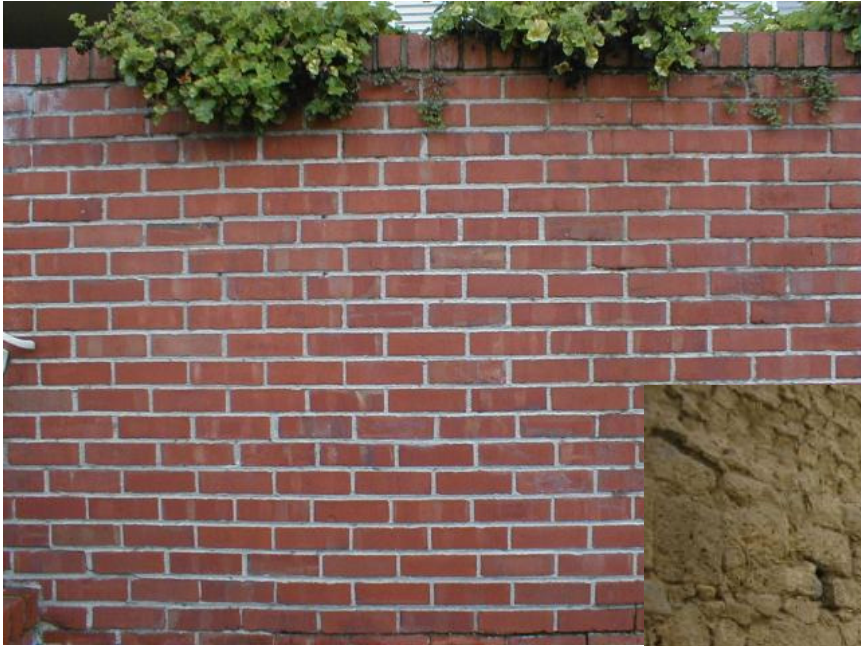
Texture-related tasks

- **Shape from texture**
 - Estimate surface orientation or shape from image texture

Shape from texture

- Use deformation of texture from point to point to estimate surface shape





Texture-related tasks

- **Shape from texture**

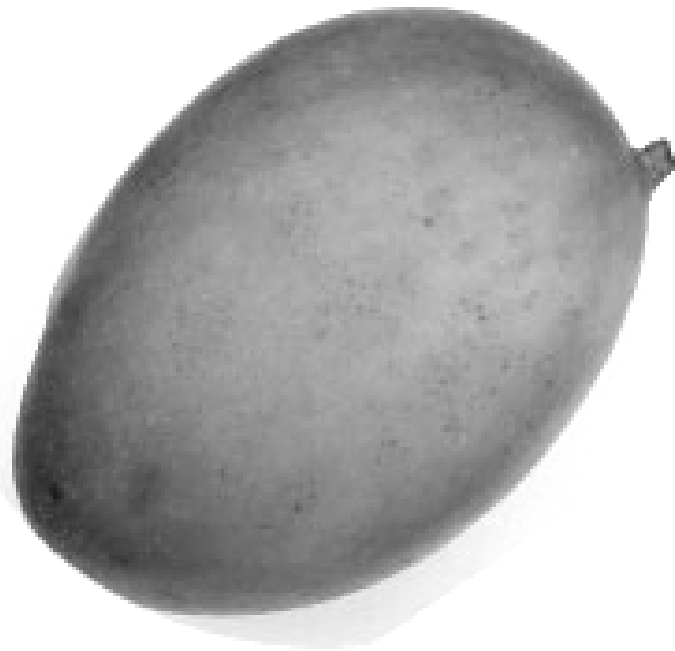
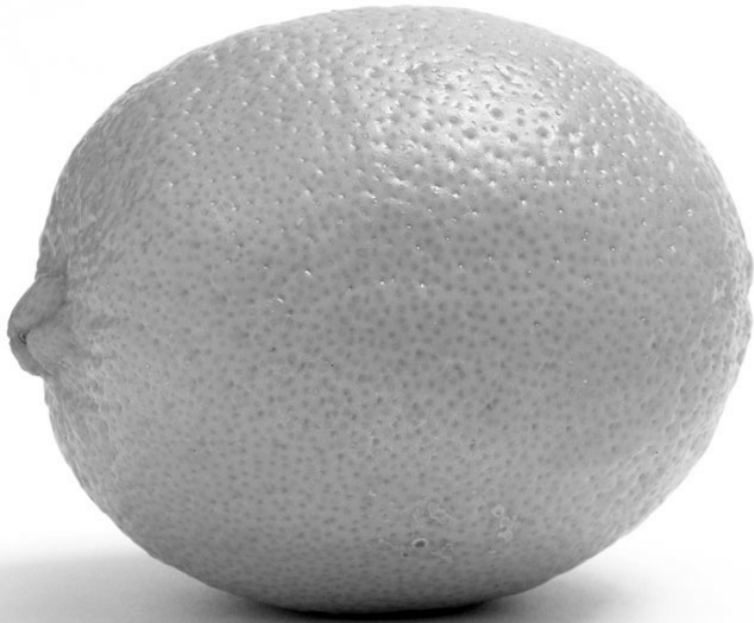
- Estimate surface orientation or shape from image texture

- **Segmentation/classification** from texture cues

- Analyze, represent texture
- Group image regions with consistent texture

- **Synthesis**

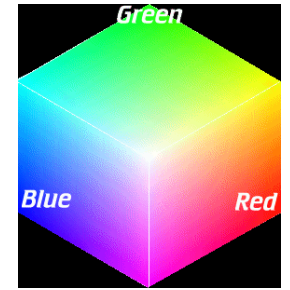
- Generate new texture patches/images given some examples





<http://animals.nationalgeographic.com/>

Color vs. texture



query



query



Recall: These looked very similar in terms of their color distributions (when our features were R-G-B)

But how would their *texture* distributions compare?

Psychophysics of texture

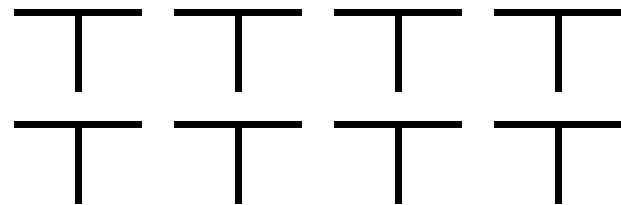
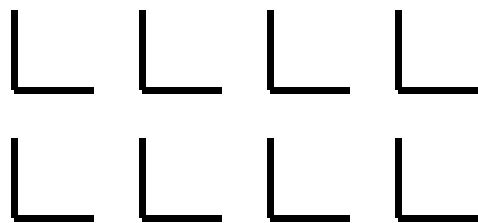
- Some textures distinguishable with *preattentive* perception—without scrutiny, eye movements [Julesz 1975]

Same or different?

L R R L L L L L L L L L
L R L L L L L L L L L L
L R R L L L L L L L L L L
L R L L L L L L L L L L L
L R L L L L L L L L L L L
L R L L L L L L L L L L L
L R L L L L L L L L L L L

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...



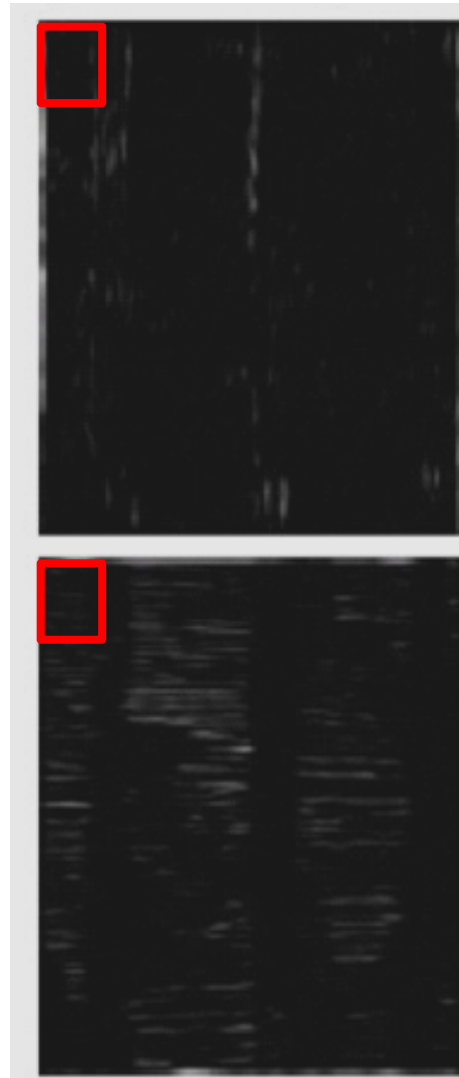
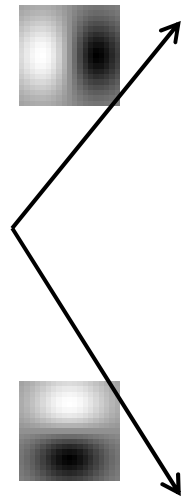
Texture representation

- Textures are made up of repeated local patterns, so:
 - Find the patterns
 - Use filters that look like patterns (spots, bars, raw patches...)
 - Consider magnitude of response
 - Describe their statistics within each local window
 - Mean, standard deviation
 - Histogram
 - Histogram of “prototypical” feature occurrences

Texture representation: example



original image

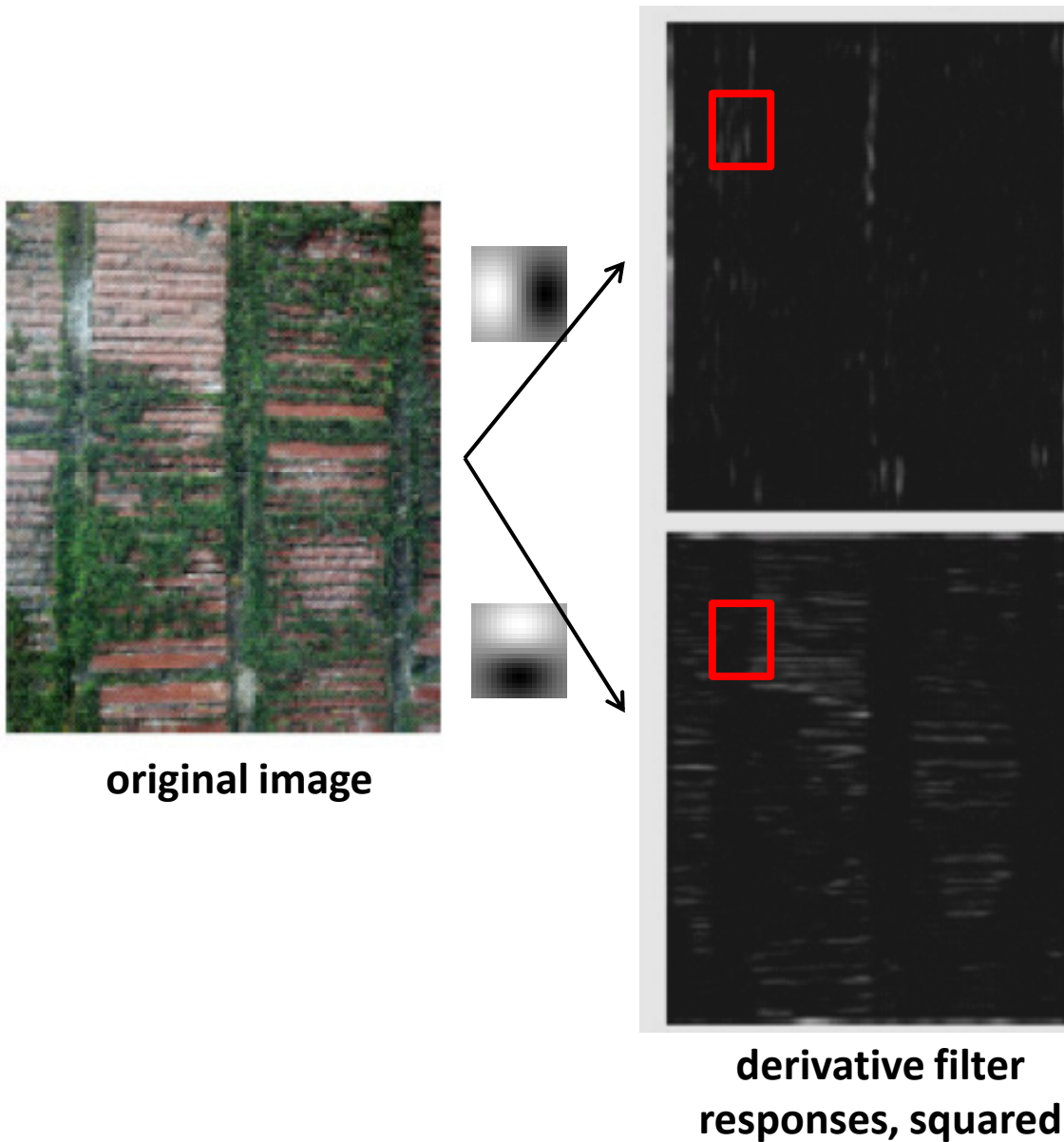


derivative filter responses, squared

	<u>mean</u> <u>d/dx</u> <u>value</u>	<u>mean</u> <u>d/dy</u> <u>value</u>
Win. #1	4	10
	⋮	

statistics to summarize patterns in small windows

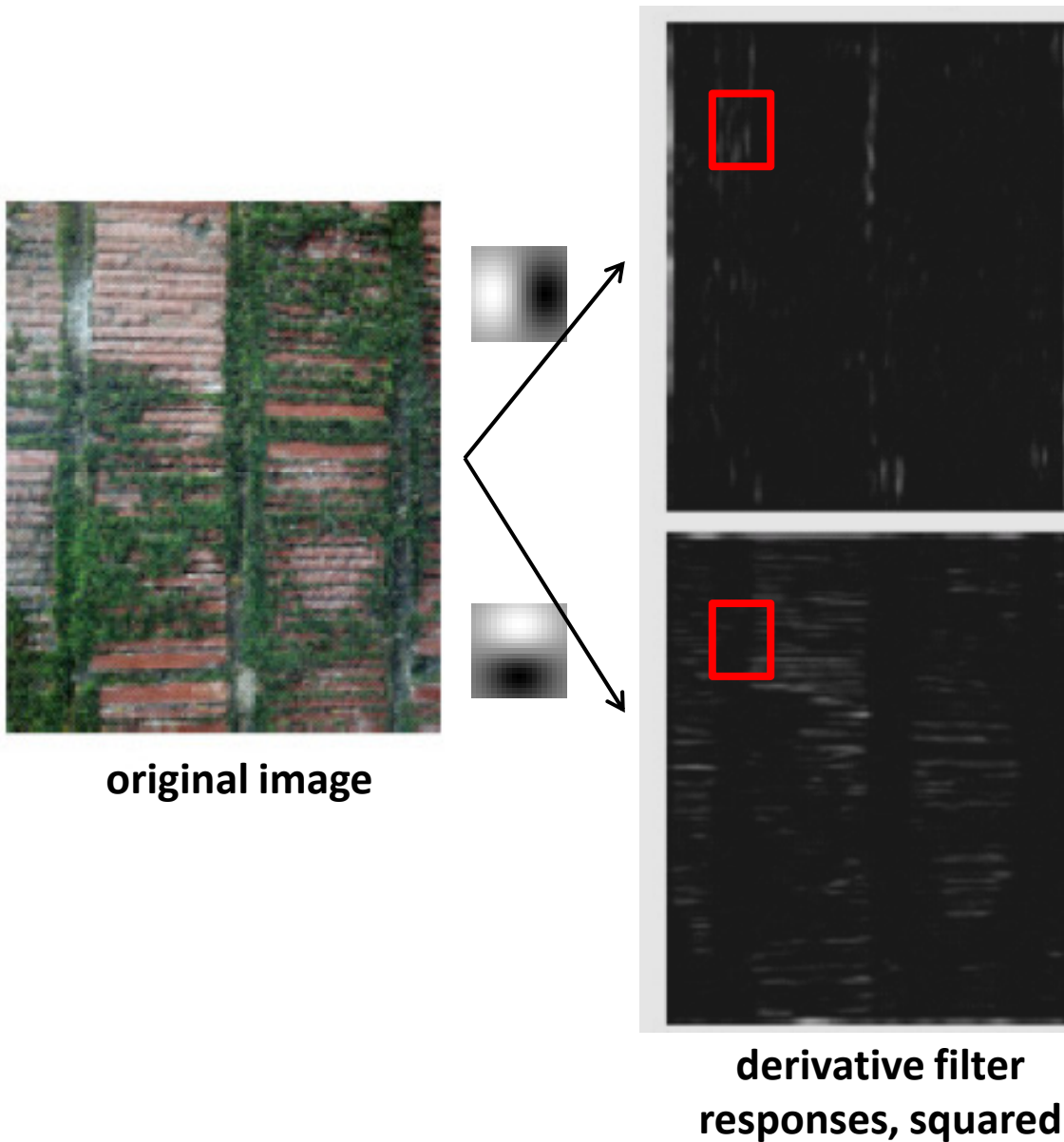
Texture representation: example



	<u>mean</u> <u>d/dx</u> <u>value</u>	<u>mean</u> <u>d/dy</u> <u>value</u>
Win. #1	4	10
Win.#2	18	7
	⋮	

statistics to summarize
patterns in small
windows

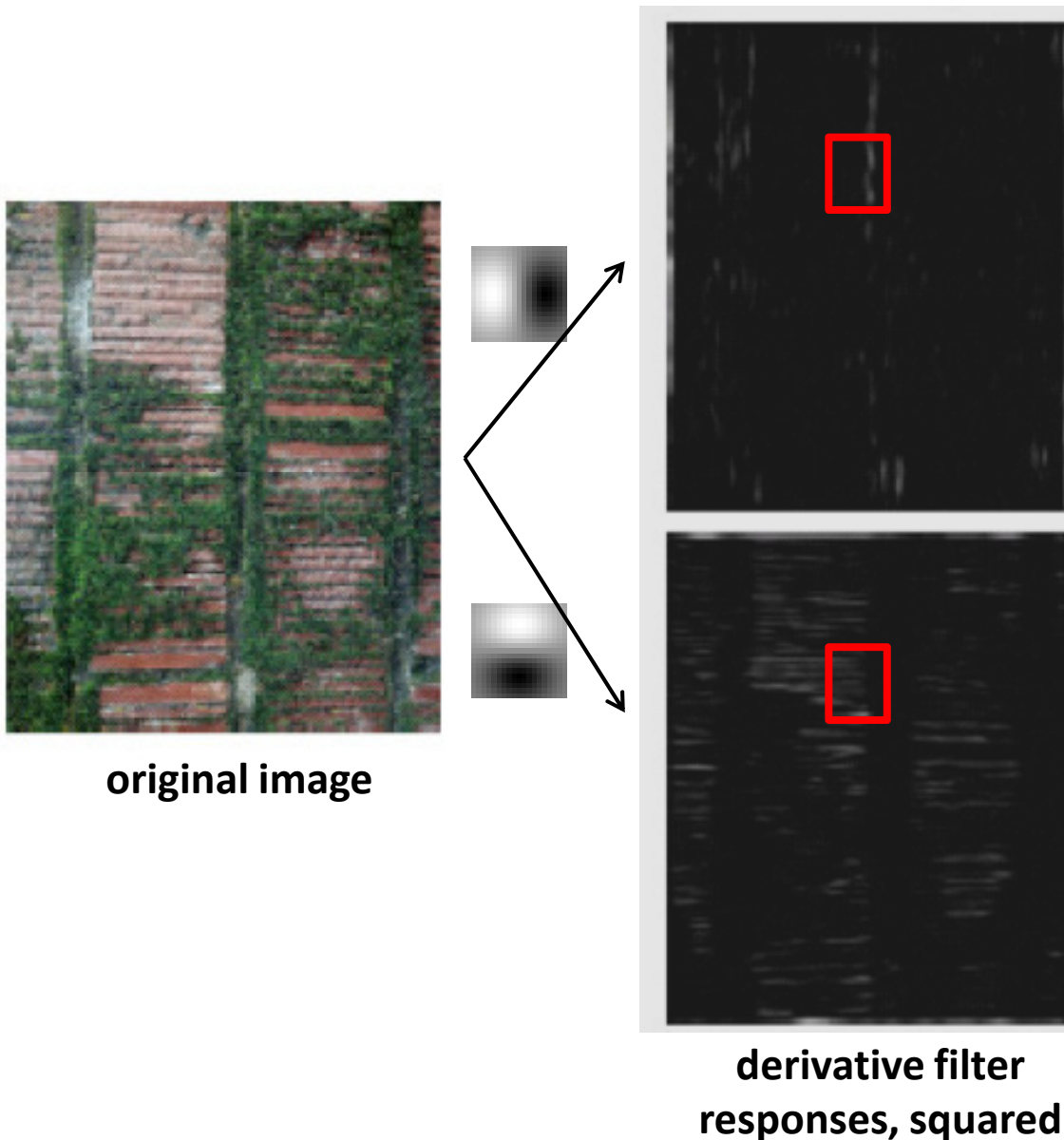
Texture representation: example



	<u>mean</u> <u>d/dx</u> <u>value</u>	<u>mean</u> <u>d/dy</u> <u>value</u>
Win. #1	4	10
Win.#2	18	7
	⋮	

statistics to summarize
patterns in small
windows

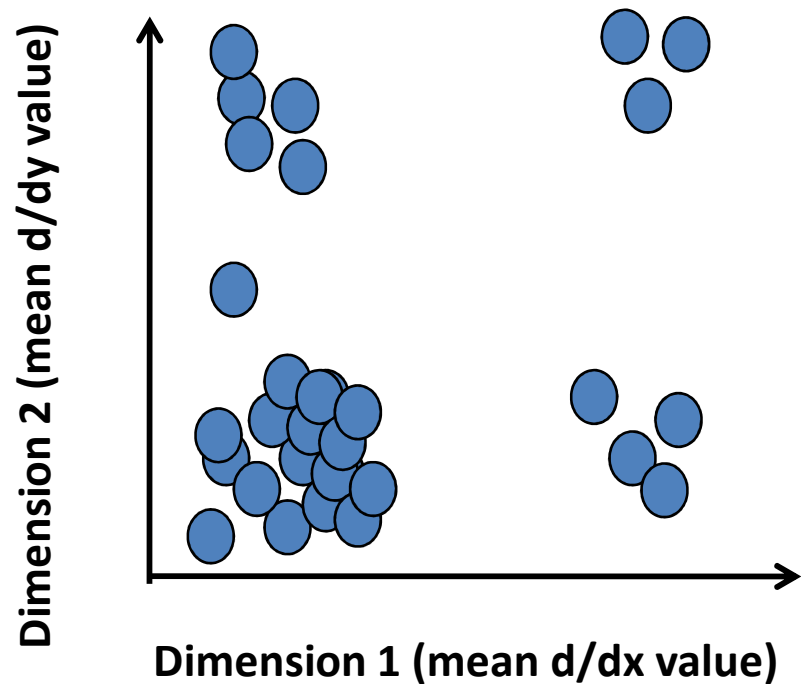
Texture representation: example



	<u>mean d/dx value</u>	<u>mean d/dy value</u>
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20
	⋮	

statistics to summarize
patterns in small
windows

Texture representation: example



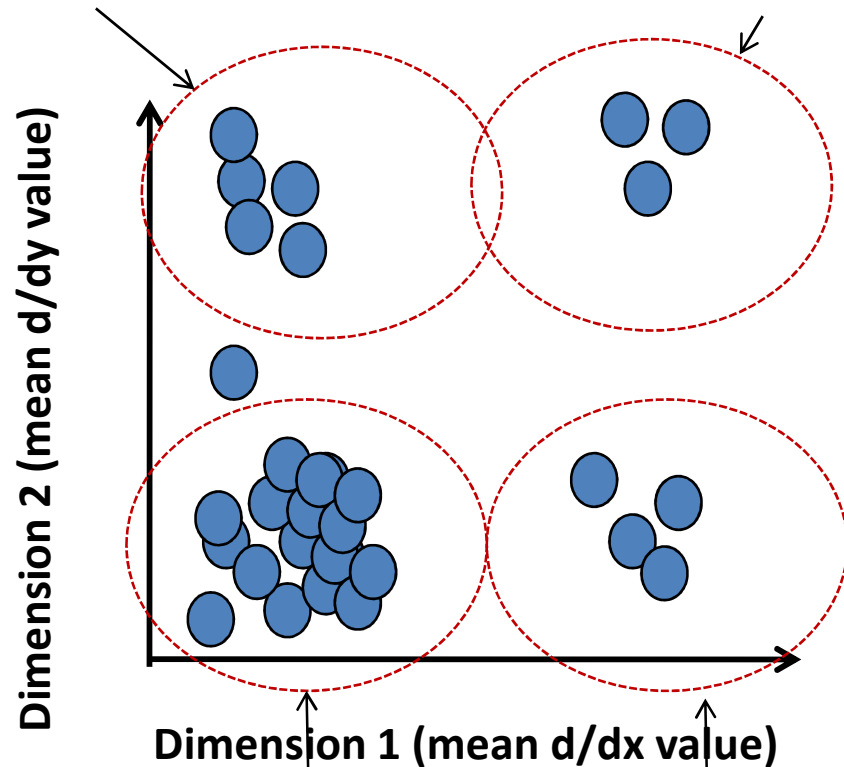
	<u>mean d/dx value</u>	<u>mean d/dy value</u>
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20
	⋮	

statistics to summarize
patterns in small
windows

Texture representation: example

Windows with primarily horizontal edges

Both



Windows with small gradient in both directions

Windows with primarily vertical edges

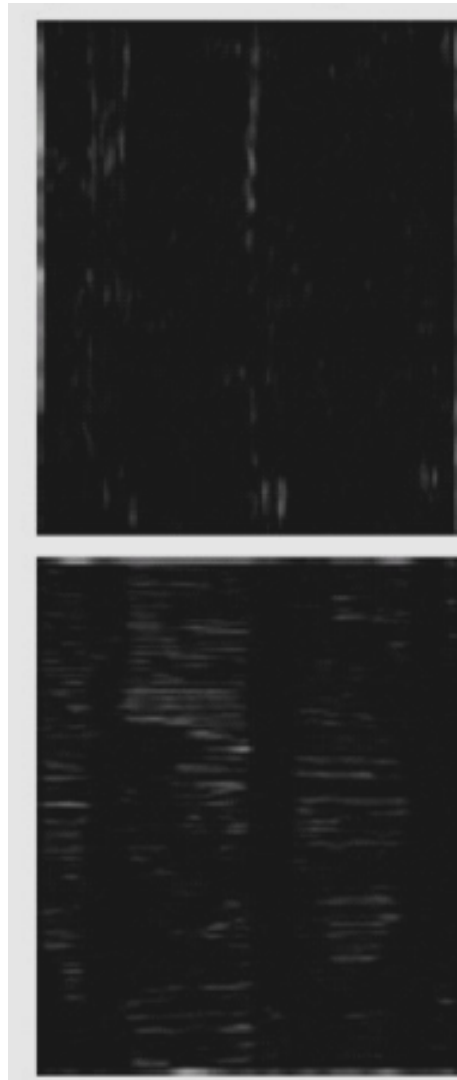
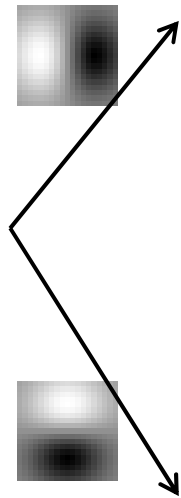
	<u>mean d/dx value</u>	<u>mean d/dy value</u>
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20
	⋮	

statistics to summarize patterns in small windows

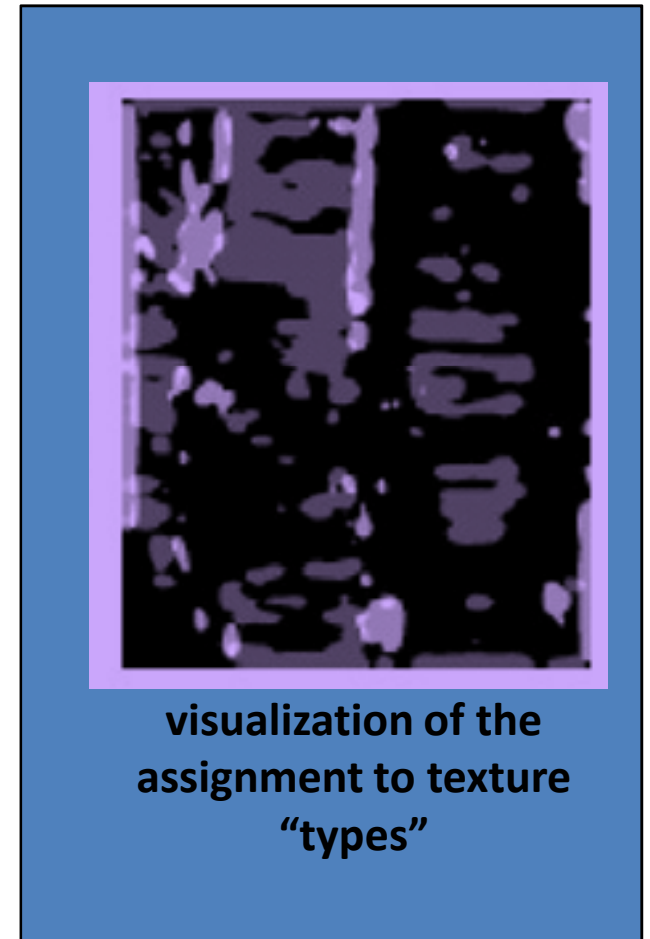
Texture representation: example



original image

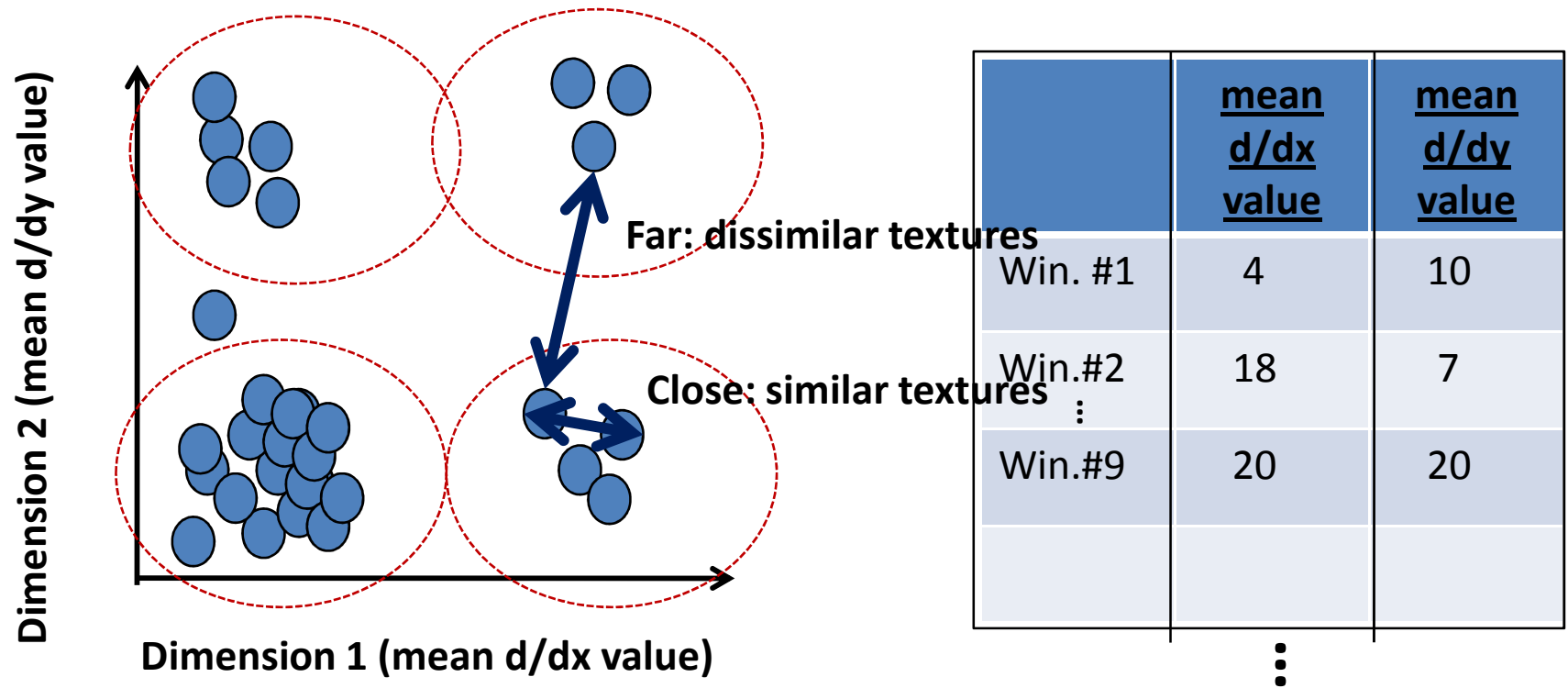


derivative filter
responses, squared



visualization of the
assignment to texture
"types"

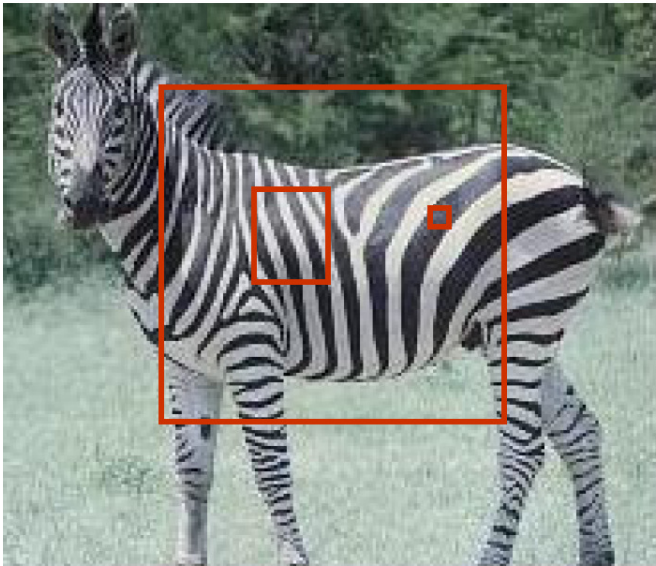
Texture representation: example



statistics to summarize
patterns in small
windows

Texture representation: window scale

- We're assuming we know the relevant window size for which we collect these statistics.

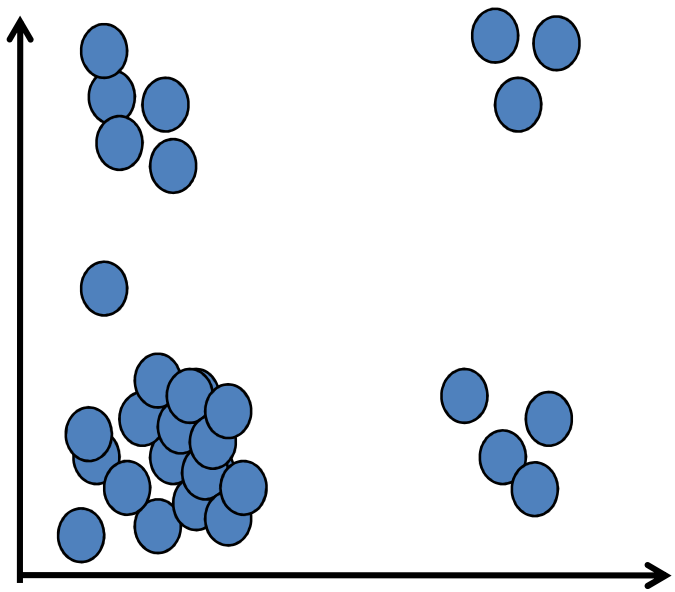


Possible to perform scale selection by looking for window scale where texture description not changing.

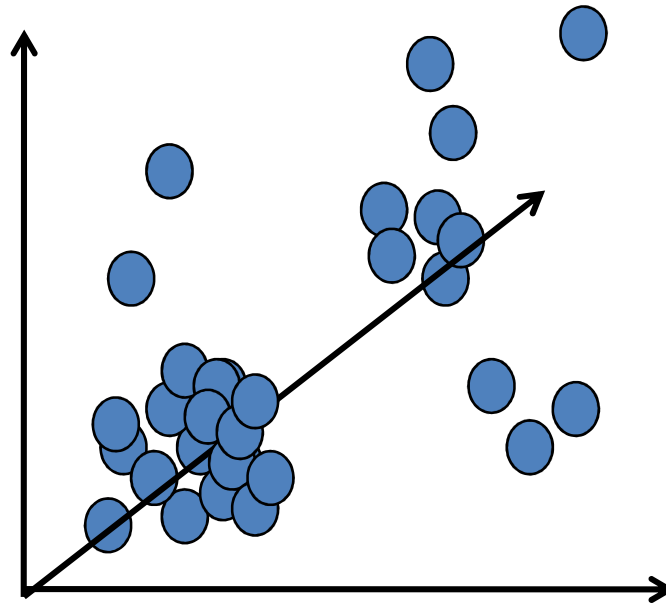
Filter banks

- Our previous example used two filters, and resulted in a 2-dimensional feature vector to describe texture in a window.
 - x and y derivatives revealed something about local structure.
- We can generalize to apply a collection of multiple (d) filters: a “filter bank”
- Then our feature vectors will be d -dimensional.
 - still can think of nearness, farness in feature space

d -dimensional features



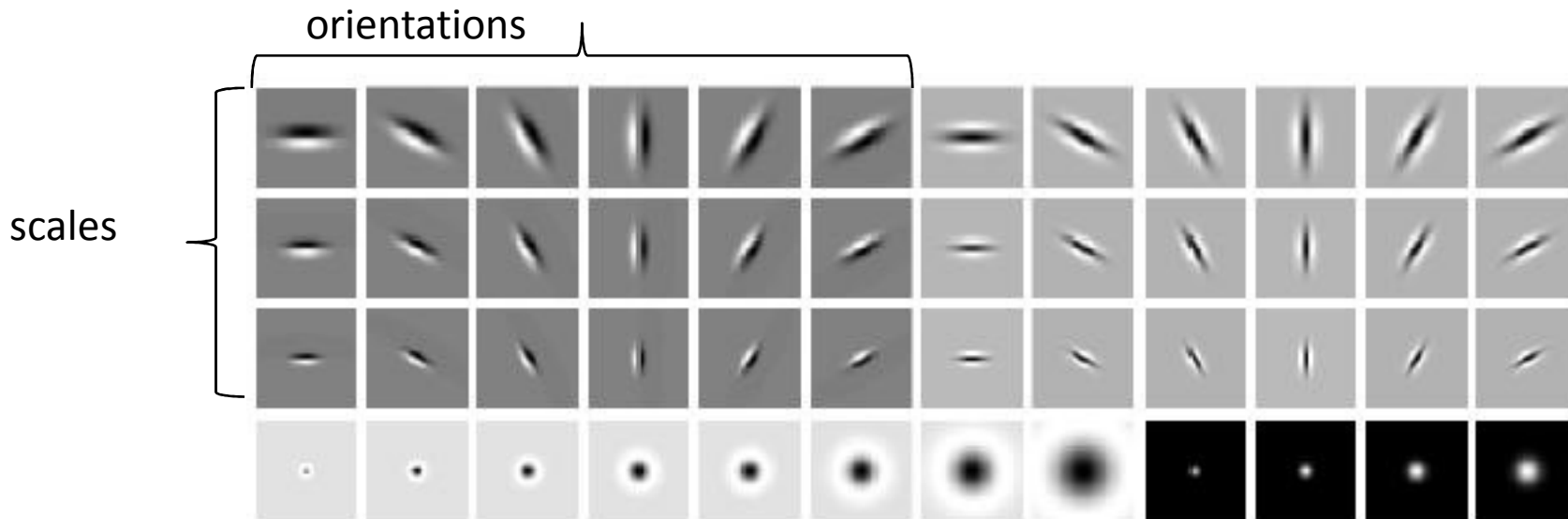
2d



3d

...

Filter banks



- What filters to put in the bank?
 - Typically we want a combination of scales and orientations, different types of patterns.

Matlab code available for these examples:

<http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html>

Influential paper:

Early vision and texture perception

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

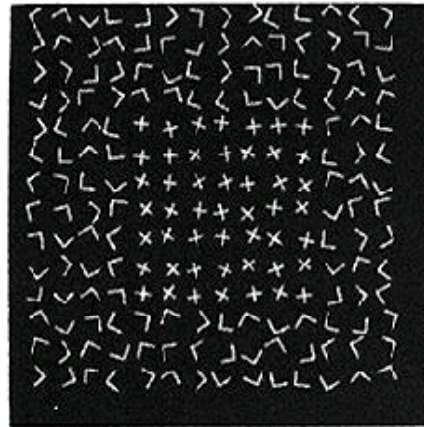
* SRI David Sarnoff Research Center, Princeton,
New Jersey 08540, USA

** Media Lab and Department of Brain and Cognitive Science,
Massachusetts Institute of Technology, Cambridge,
Massachusetts 02139, USA

Learn: use filters.

Bergen and Adelson, Nature 1988

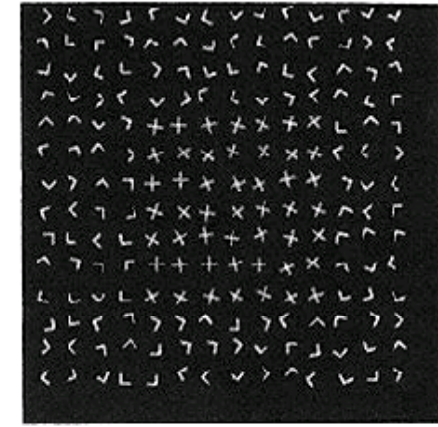
Fig. 1 *Top row*, Textures consisting of Xs within a texture composed of Ls. The micropatterns are placed at random orientations on a randomly perturbed lattice. *a*, The bars of the Xs have the same length as the bars of the Ls. *b*, The bars of the Ls have been lengthened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is enhanced. *c*, The bars of the Ls have been shortened by 25%, and the intensity adjusted for the same mean luminance. Discriminability 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*.



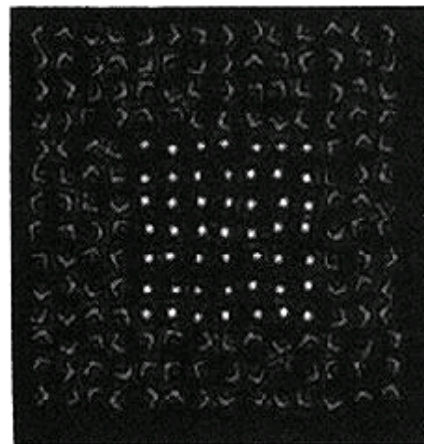
a



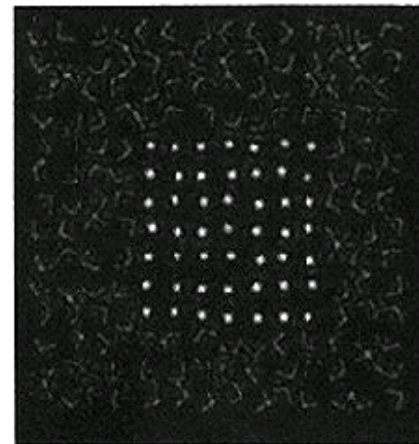
b



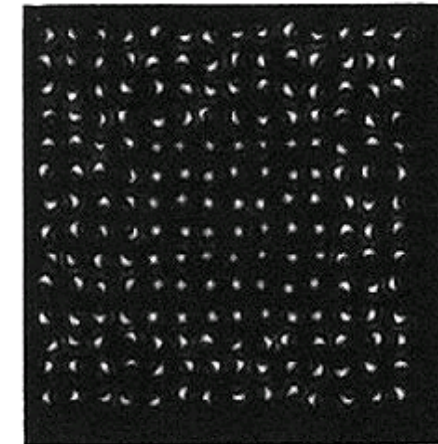
c



d



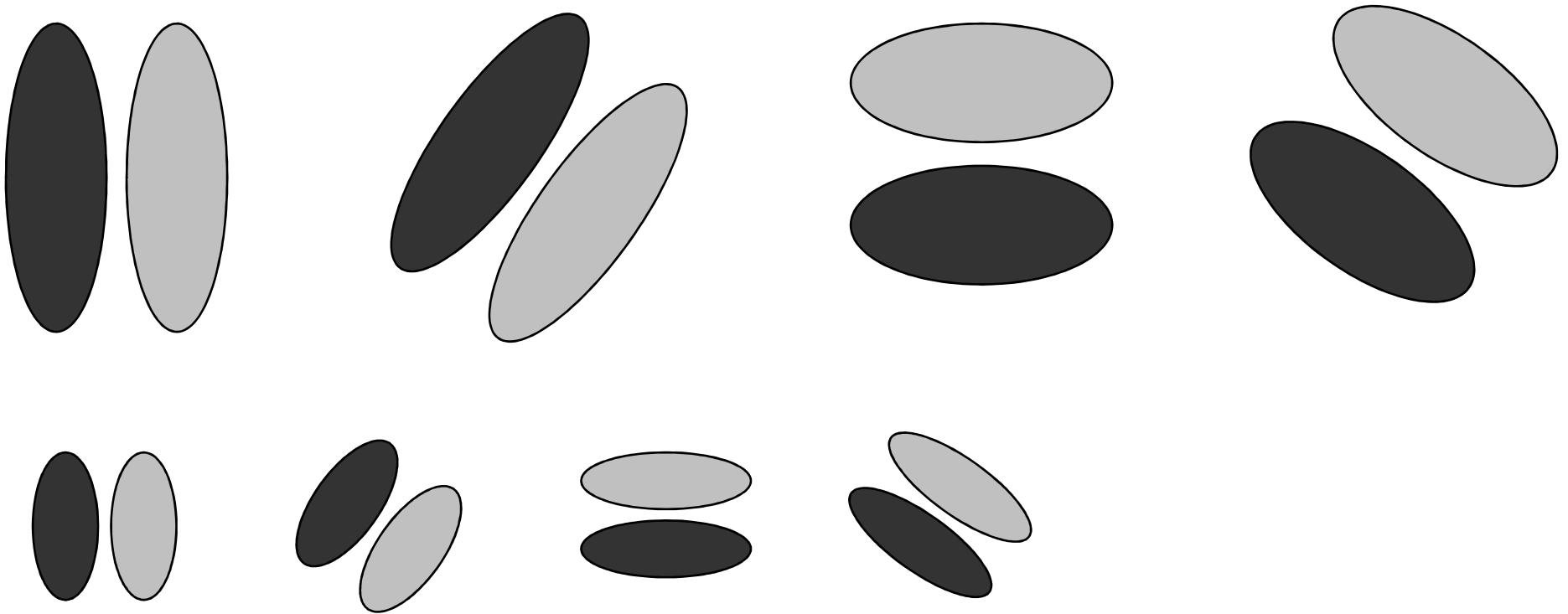
e



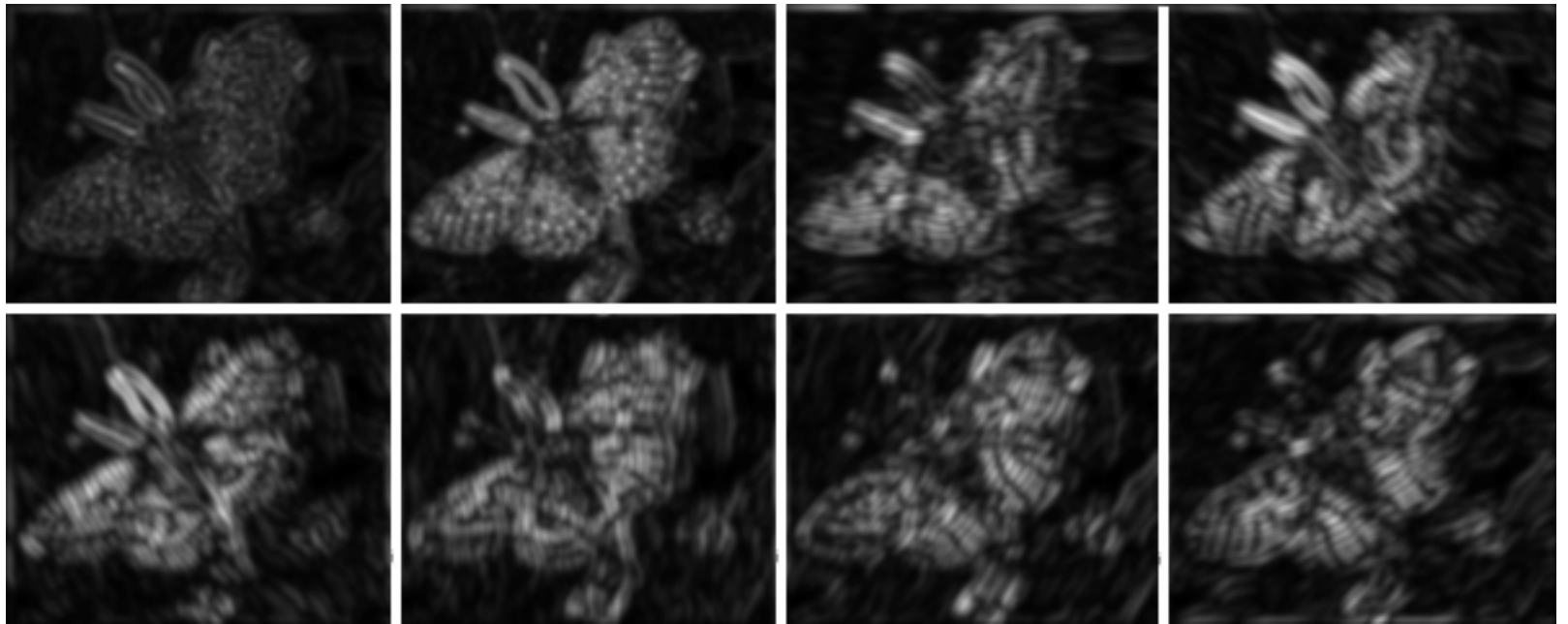
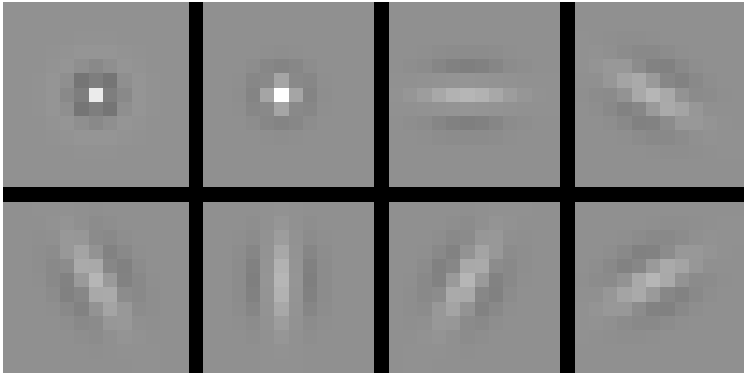
f

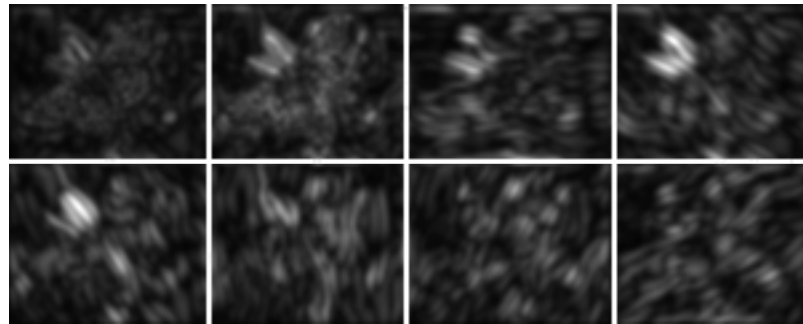
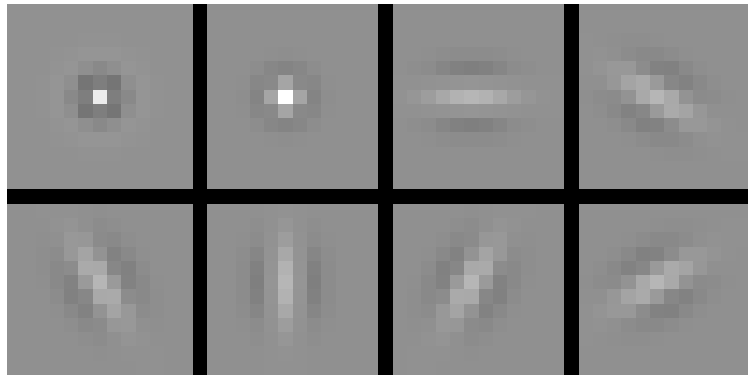
Learn: use lots of filters, multi-ori&scale.

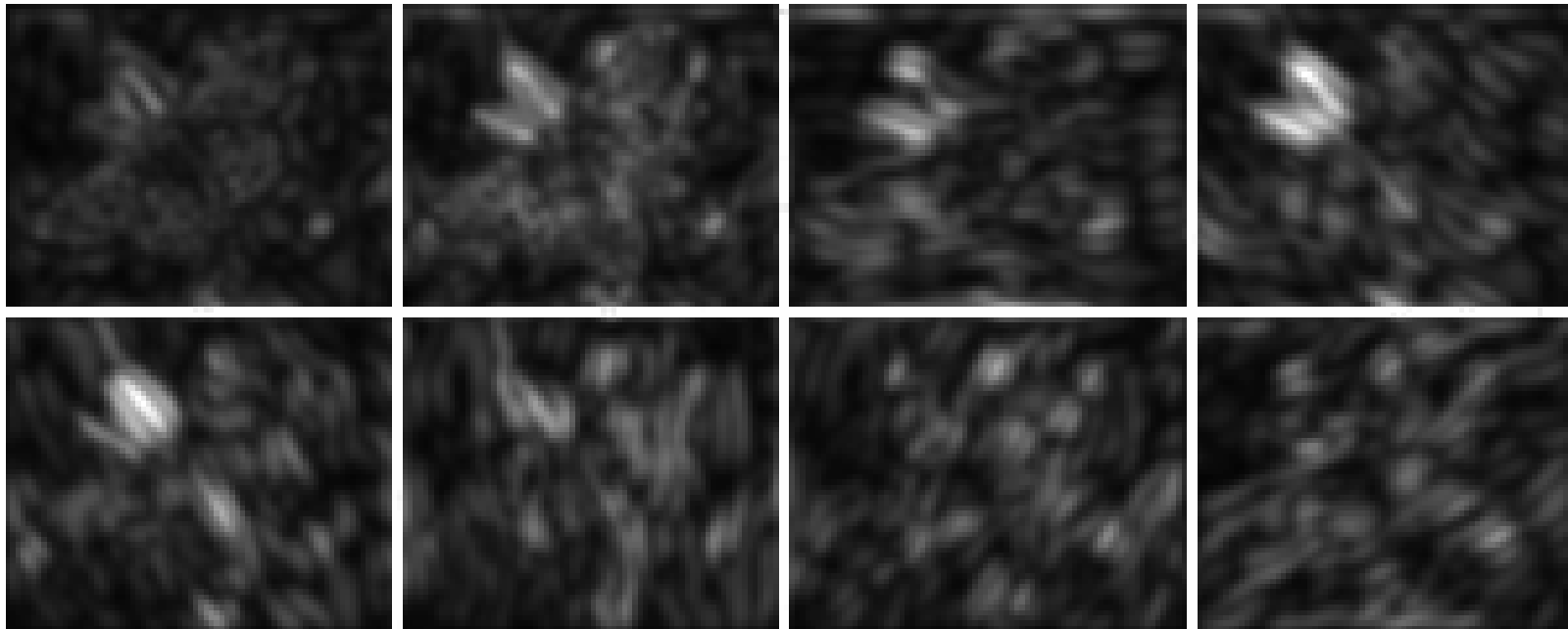
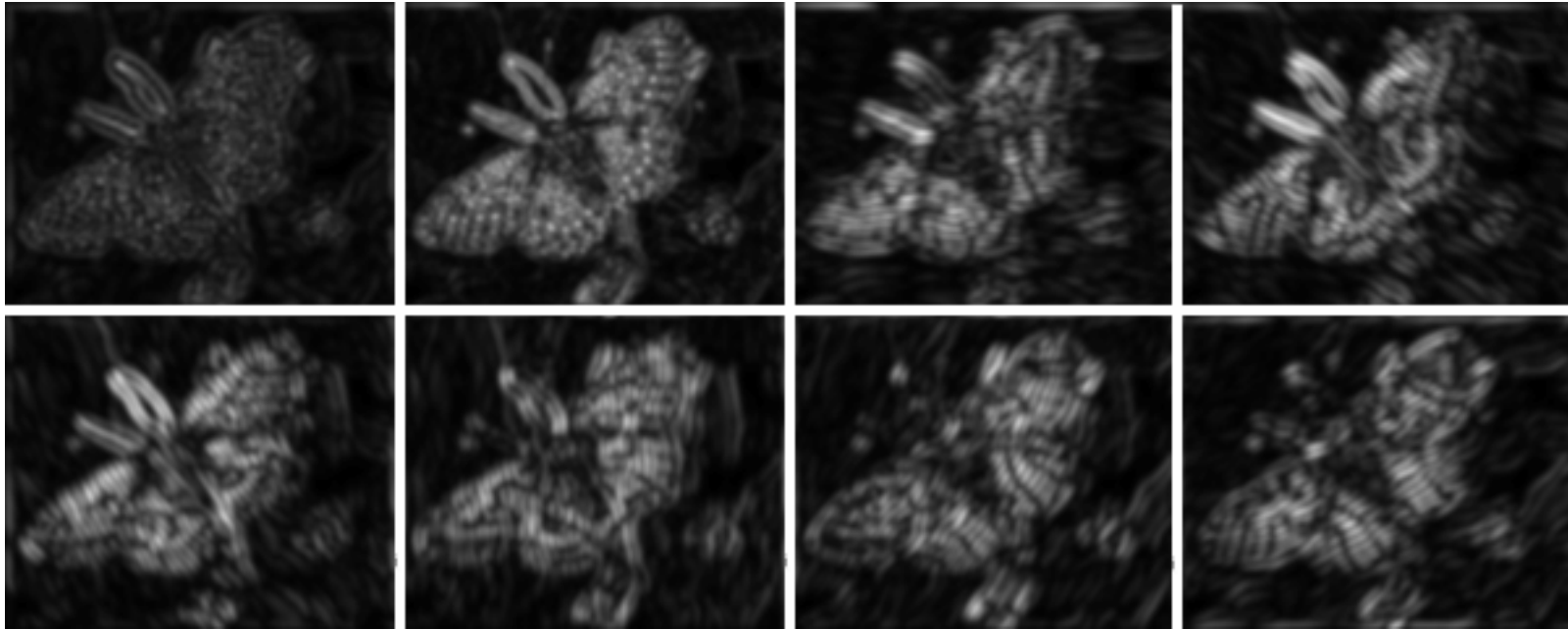
Malik and Perona



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







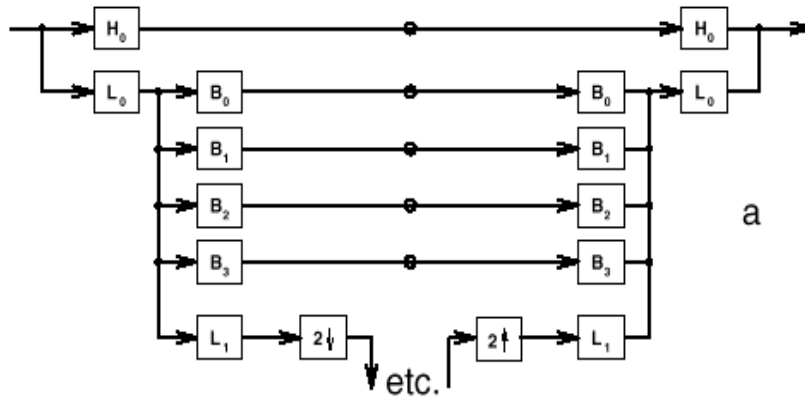
If matching the averaged squared filter values is a good way to match a given texture, then maybe matching the entire marginal distribution (eg, the histogram) of a filter's response would be even better.

Jim Bergen proposed this...

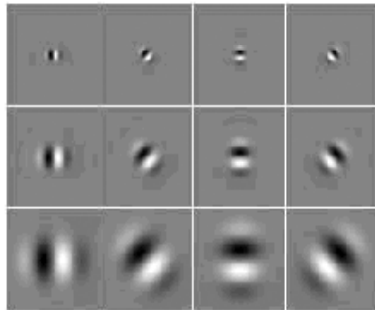
Pyramid-Based Texture Analysis/Synthesis

David J. Heeger^{*}
Stanford University

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



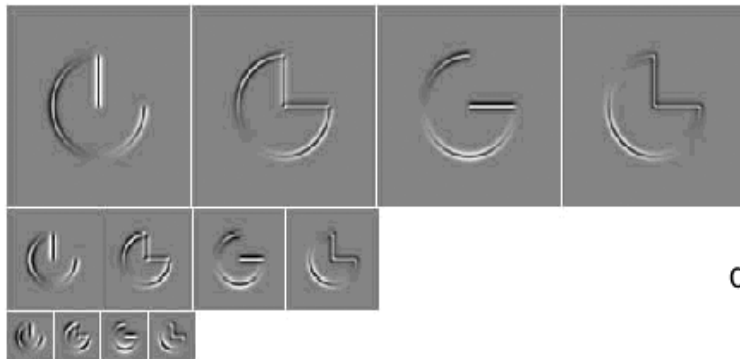
SIGGRAPH 1994



b



c



d

e

Histogram matching algorithm

```
Match-histogram (im1, im2)
  im1-cdf = Make-cdf(im1)
  im2-cdf = Make-cdf(im2)
  inv-im2-cdf = Make-inverse-lookup-table(im2-cdf)
  Loop for each pixel do
    im1[pixel] =
      Lookup(inv-im2-cdf,
            Lookup(im1-cdf, im1[pixel]))
```

“At this im1 pixel value, 10% of the im1 values are lower. What im2 pixel value has 10% of the im2 values below it?”

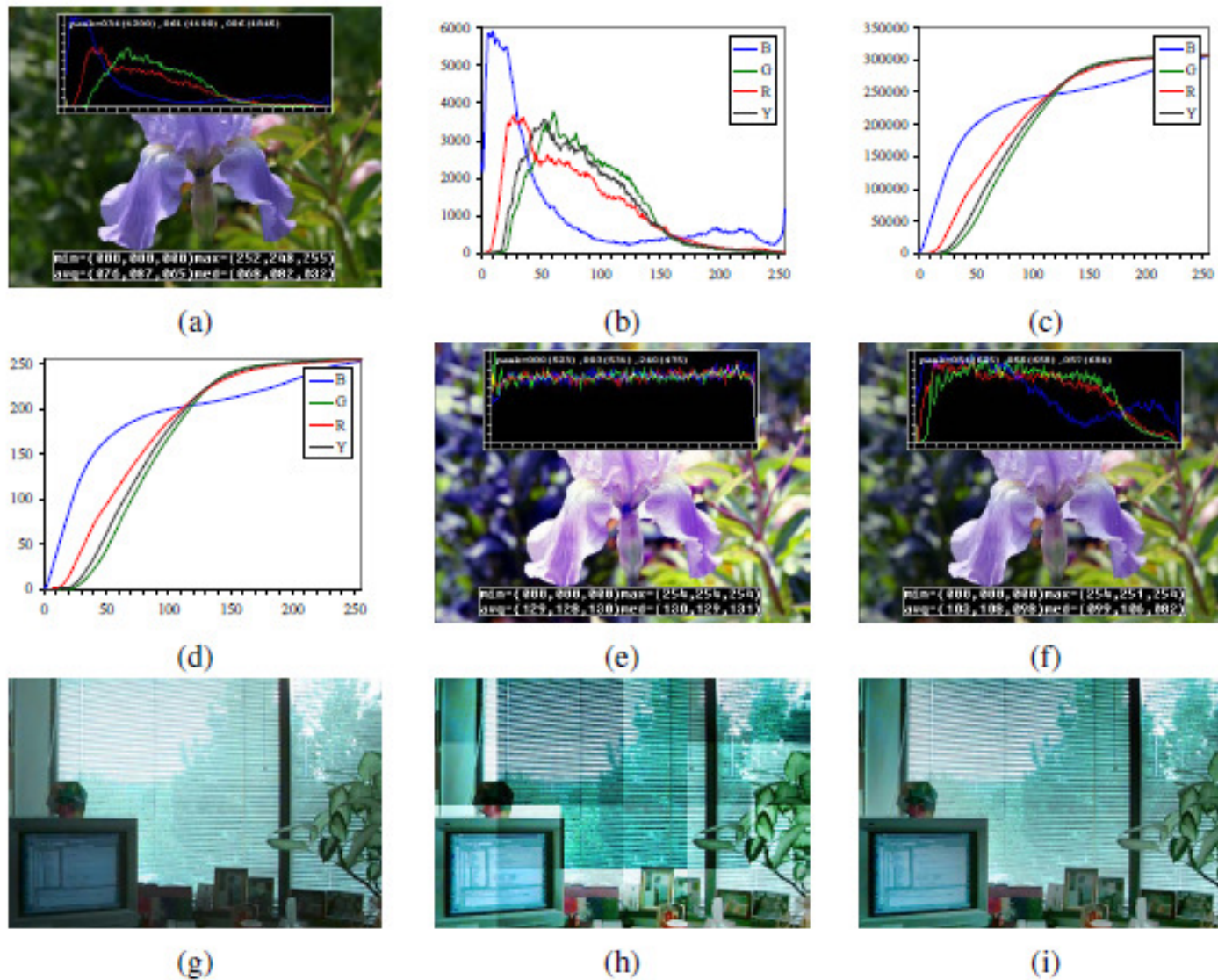


Figure 3.7: Histogram analysis and equalization: (a) original image (b) color channel and intensity (luminance) histograms; (c) cumulative distribution functions; (d) equalization (transfer) functions; (e) full histogram equalization; (f) partial histogram equalization; (g) another sample image; (h) block histogram equalization; (i) locally adaptive histogram equalization.

Heeger-Bergen texture synthesis algorithm

```
Match-texture(noise, texture)
  Match-Histogram (noise, texture)
  analysis-pyr = Make-Pyramid (texture)
  Loop for several iterations do
    synthesis-pyr = Make-Pyramid (noise)
    Loop for a-band in subbands of analysis-pyr
      for s-band in subbands of synthesis-pyr
        do
          Match-Histogram (s-band, a-band)
    noise = Collapse-Pyramid (synthesis-pyr)
  Match-Histogram (noise, texture)
```

Alternate matching the histograms of all the subbands and matching the histograms of the reconstructed images.

Learn: use filter marginal statistics.

Bergen and Heeger

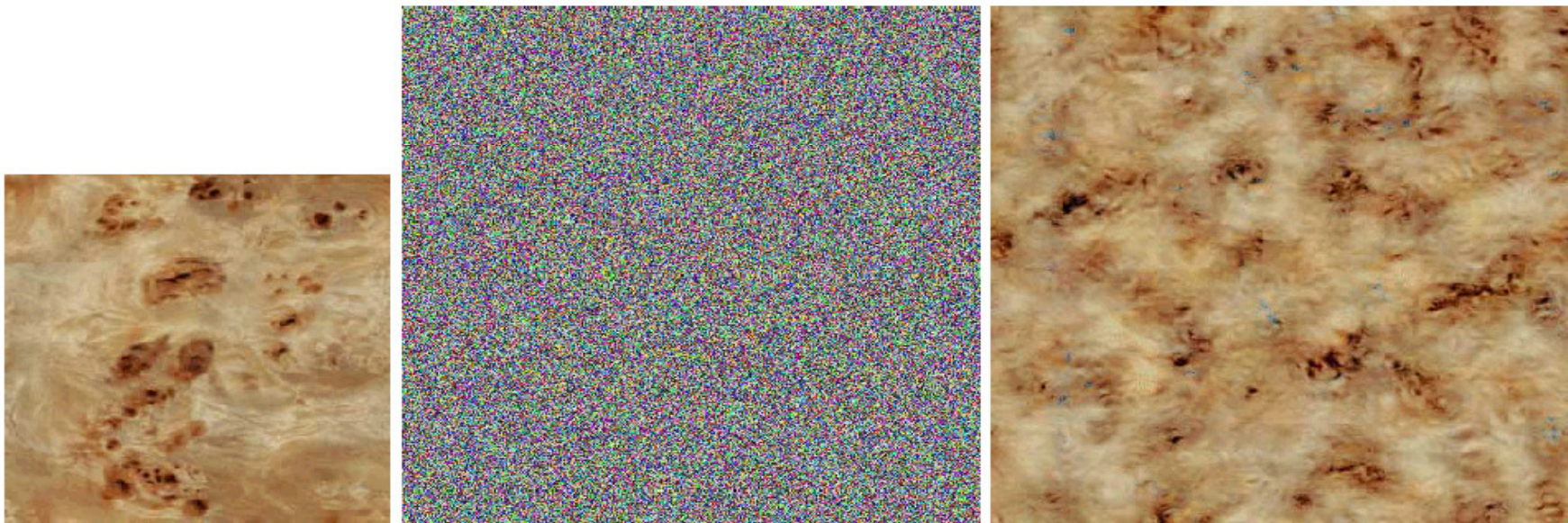


Figure 2: (Left) Input digitized sample texture: burl wood. (Middle) Input noise. (Right) Output synthetic texture that matches the appearance of the digitized sample. Note that the synthesized texture is larger than the digitized sample; our approach allows generation of as much texture as desired. 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

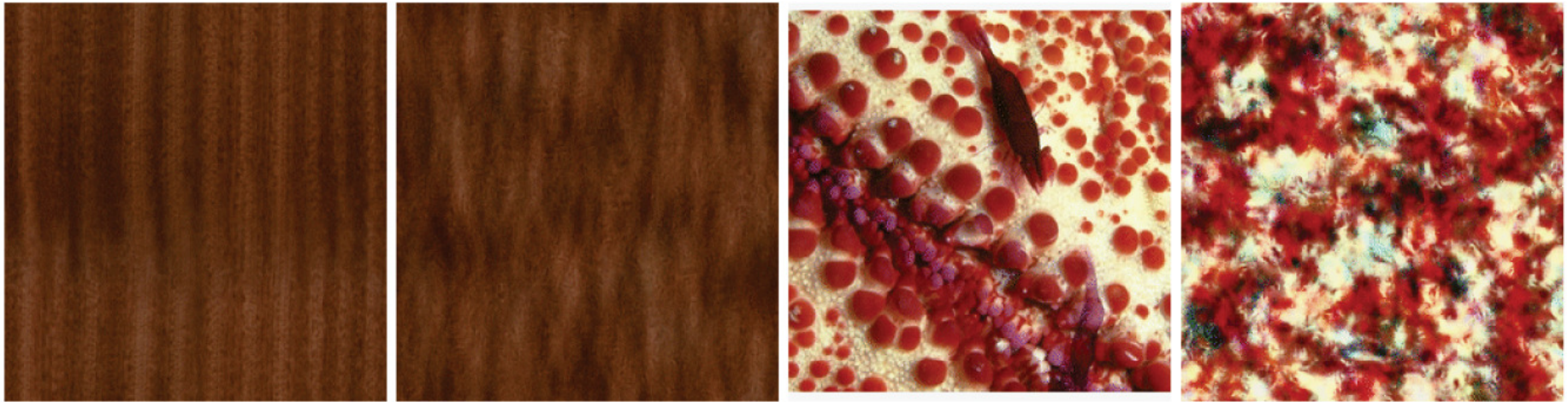


Figure 8: Examples of failures: wood grain and red coral.

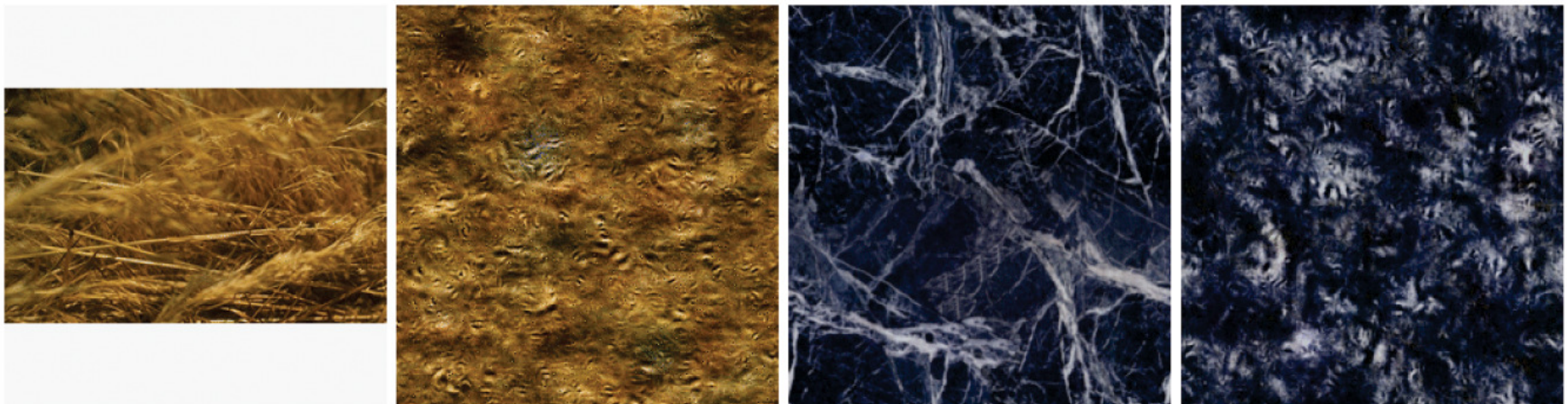


Figure 9: More failures: hay and marble.

De Bonet (and Viola)

SIGGRAPH 1997

Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images

Jeremy S. De Bonet -
Learning & Vision Group
Artificial Intelligence Laboratory
Massachusetts Institute of Technology

EMAIL: jsd@ai.mit.edu
HOMEPAGE: <http://www.ai.mit.edu/~jsd>

Learn: use filter conditional statistics across scale. DeBonet

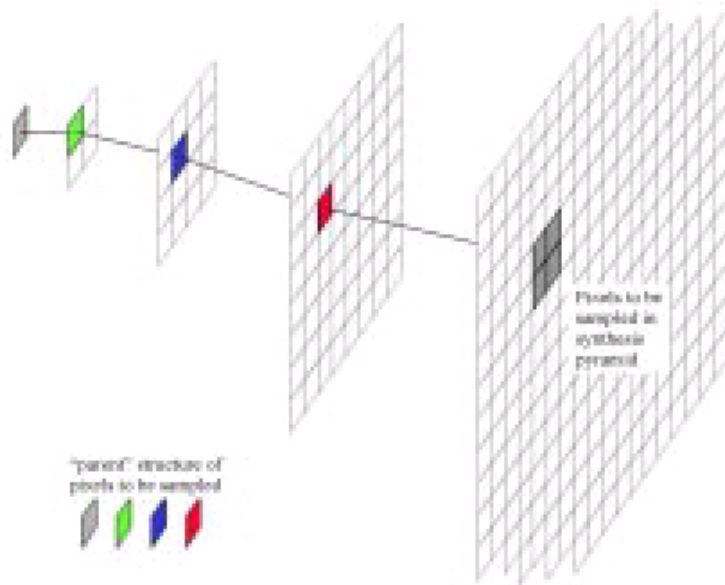


Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the “parent” structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

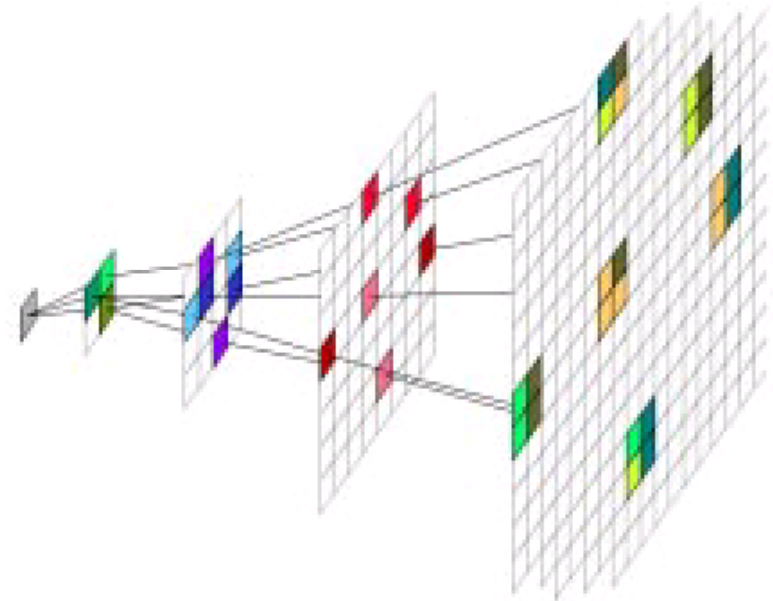
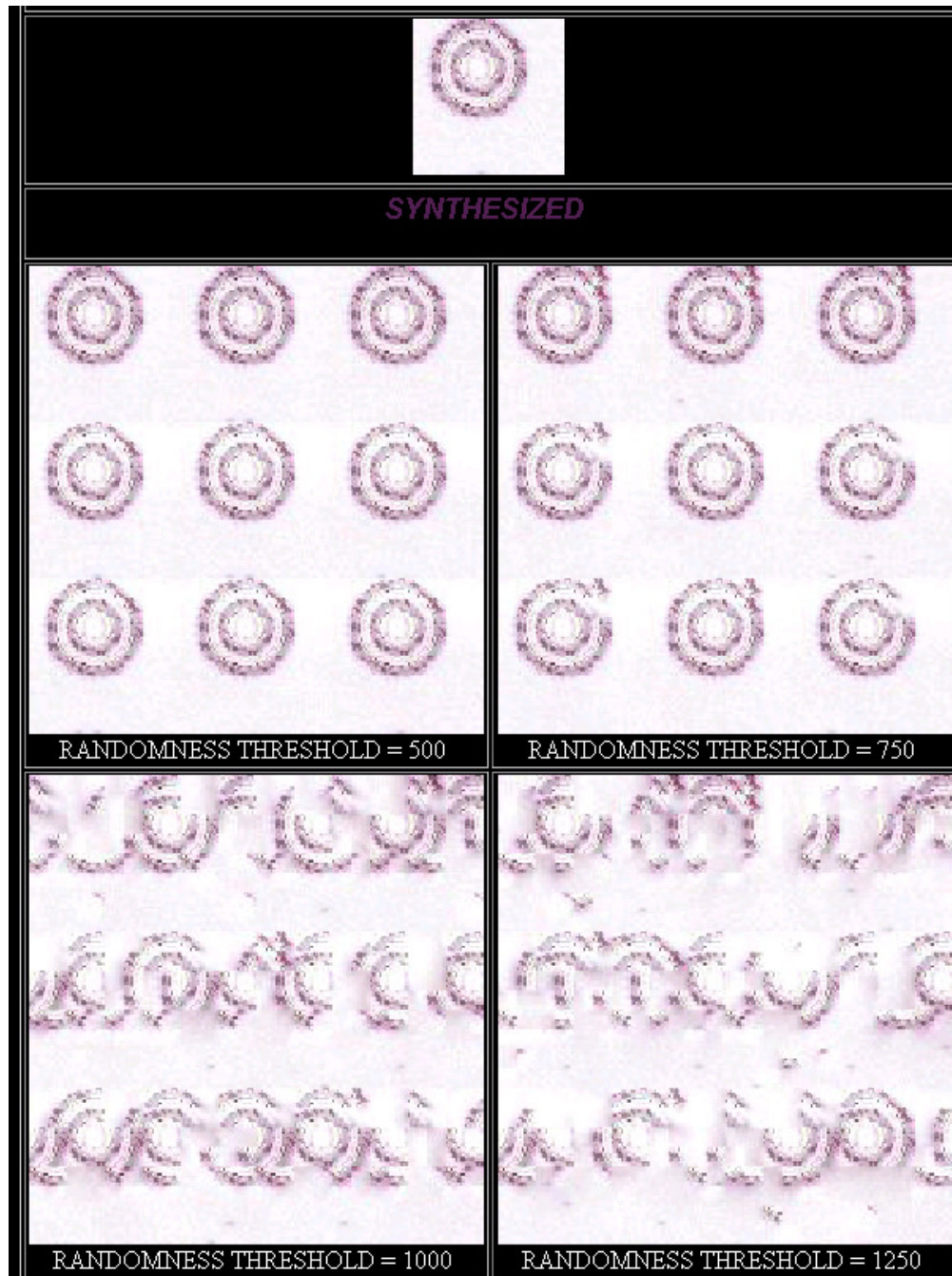
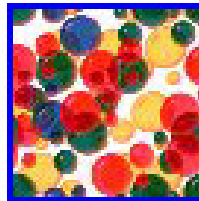
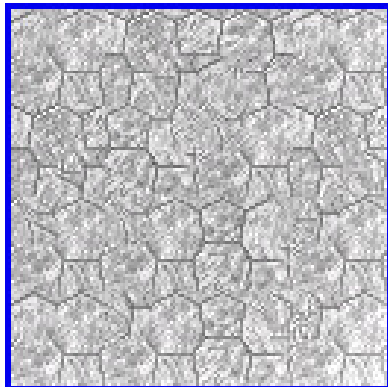
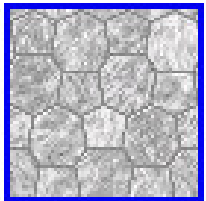
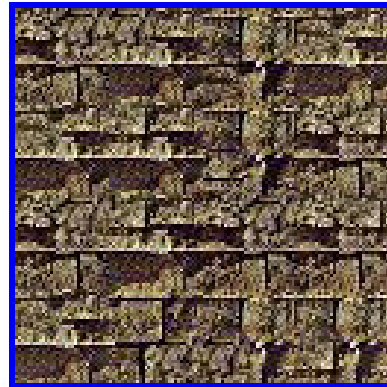
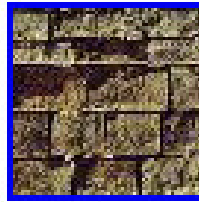
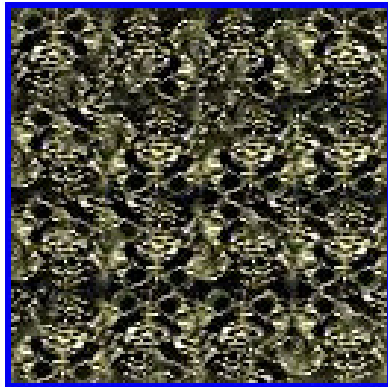
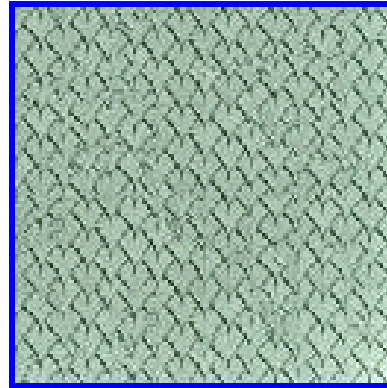
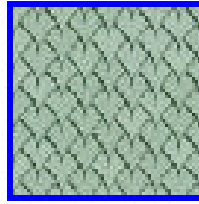
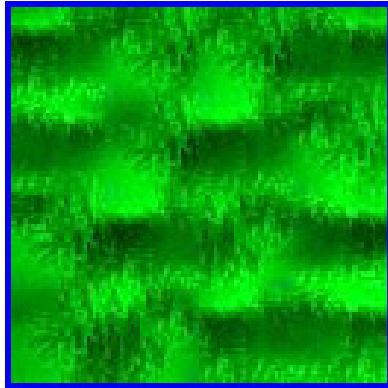
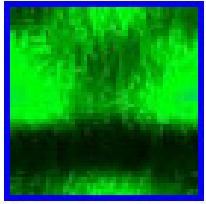


Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.

DeBonet



DeBonet



What we've learned from the previous texture synthesis methods

From Adelson and Bergen:

examine filter outputs

From Perona and Malik:

use multi-scale, multi-orientation filters.

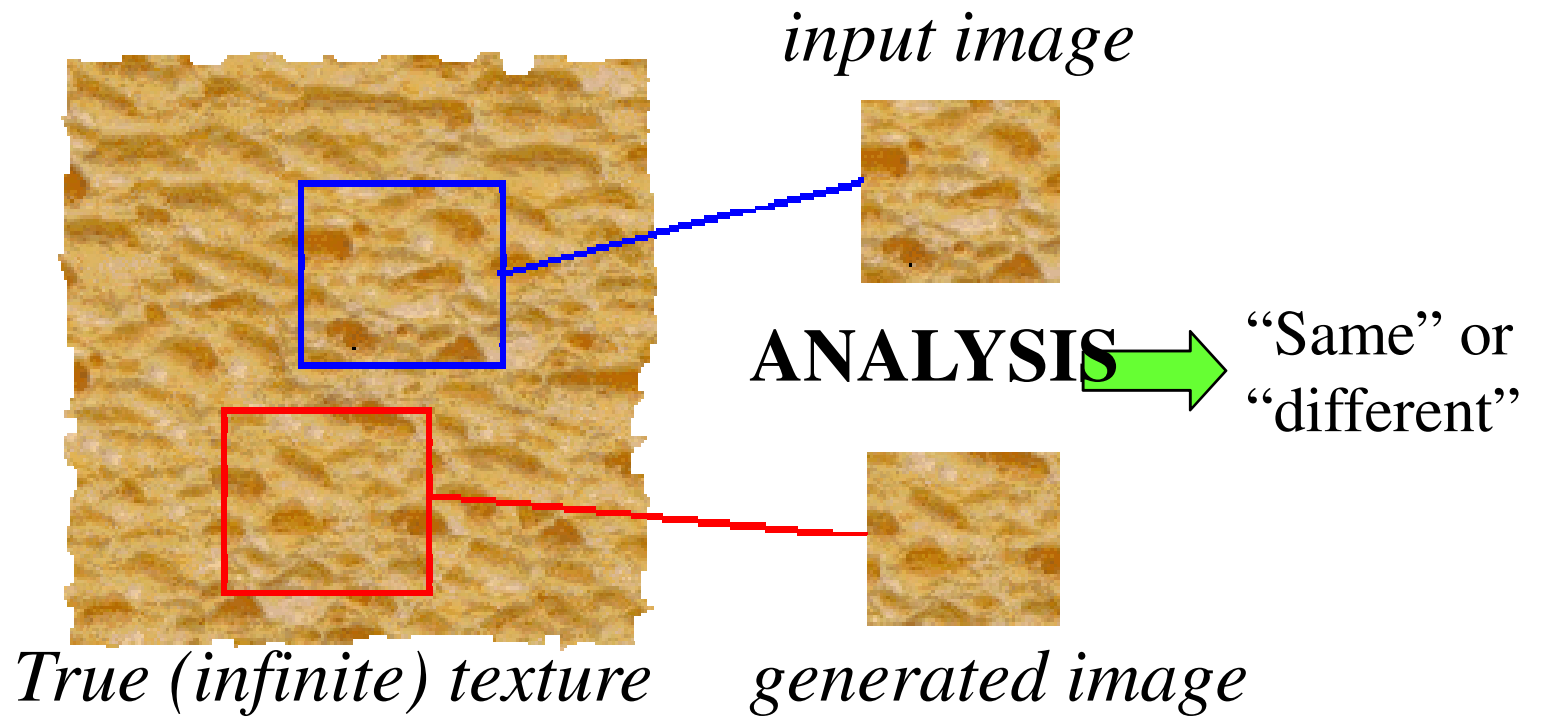
From Heeger and Bergen:

use marginal statistics (histograms) of filter responses.

From DeBonet:

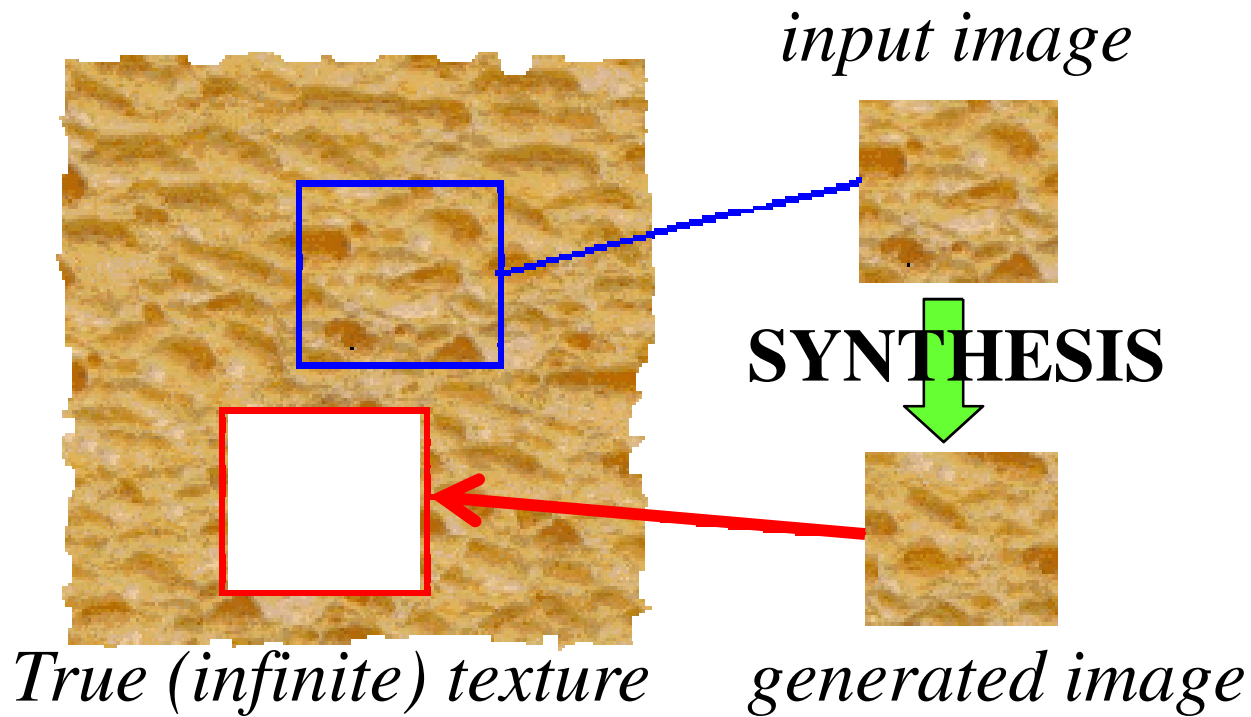
use conditional filter responses across scale.

The Goal of Texture Analysis



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

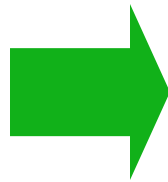
The Goal of Texture Synthesis



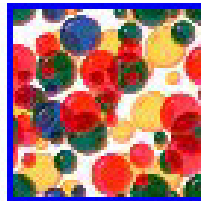
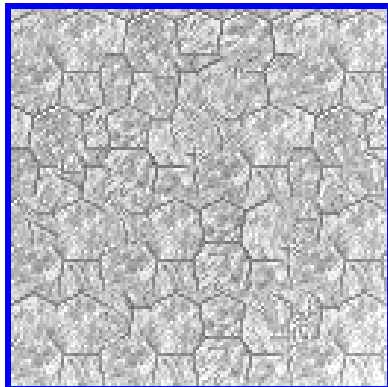
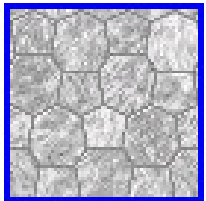
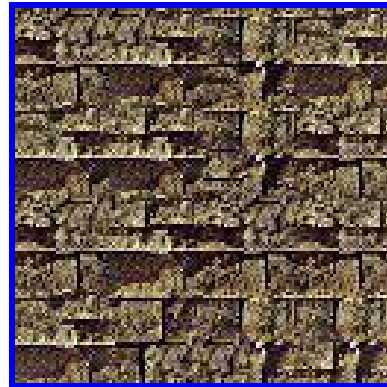
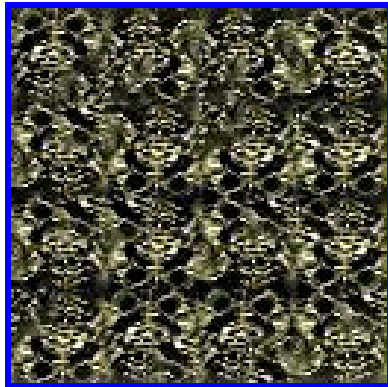
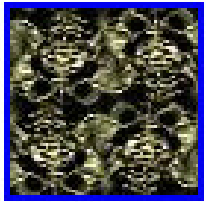
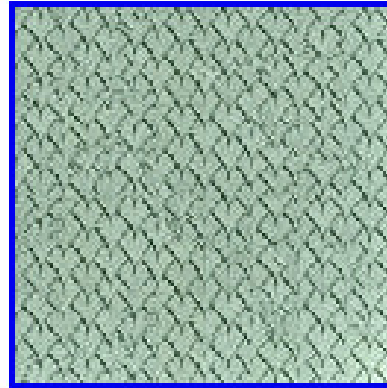
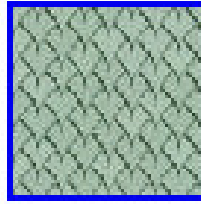
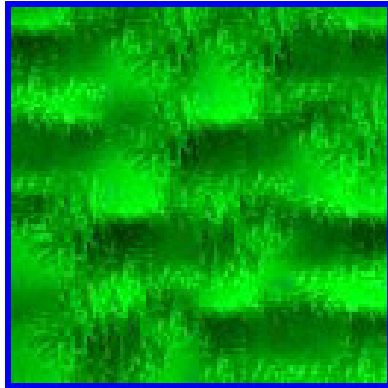
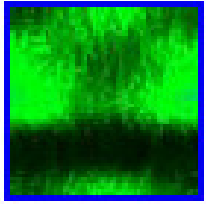
- Given a finite sample of some texture, the goal is to synthesize other samples from that same texture
 - The sample needs to be "large enough"

Texture synthesis

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



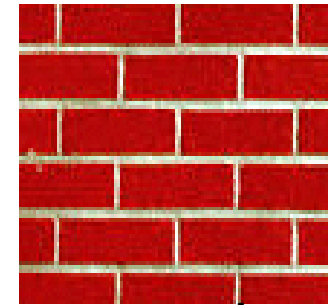
DeBonet,
again...



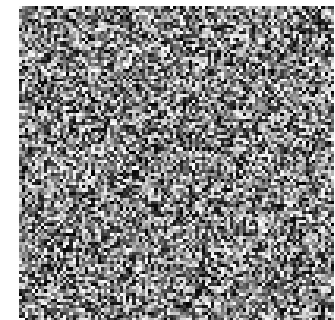
The Challenge

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

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



repeated



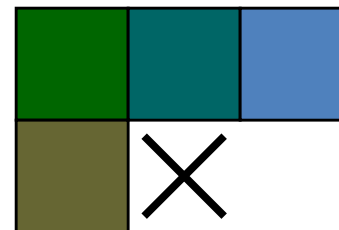
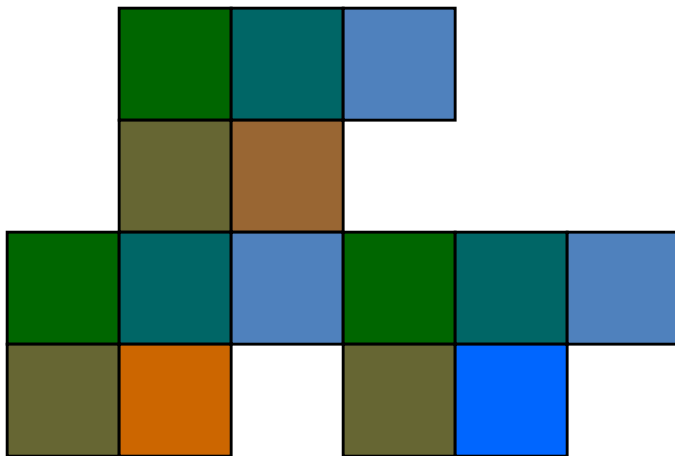
stochastic



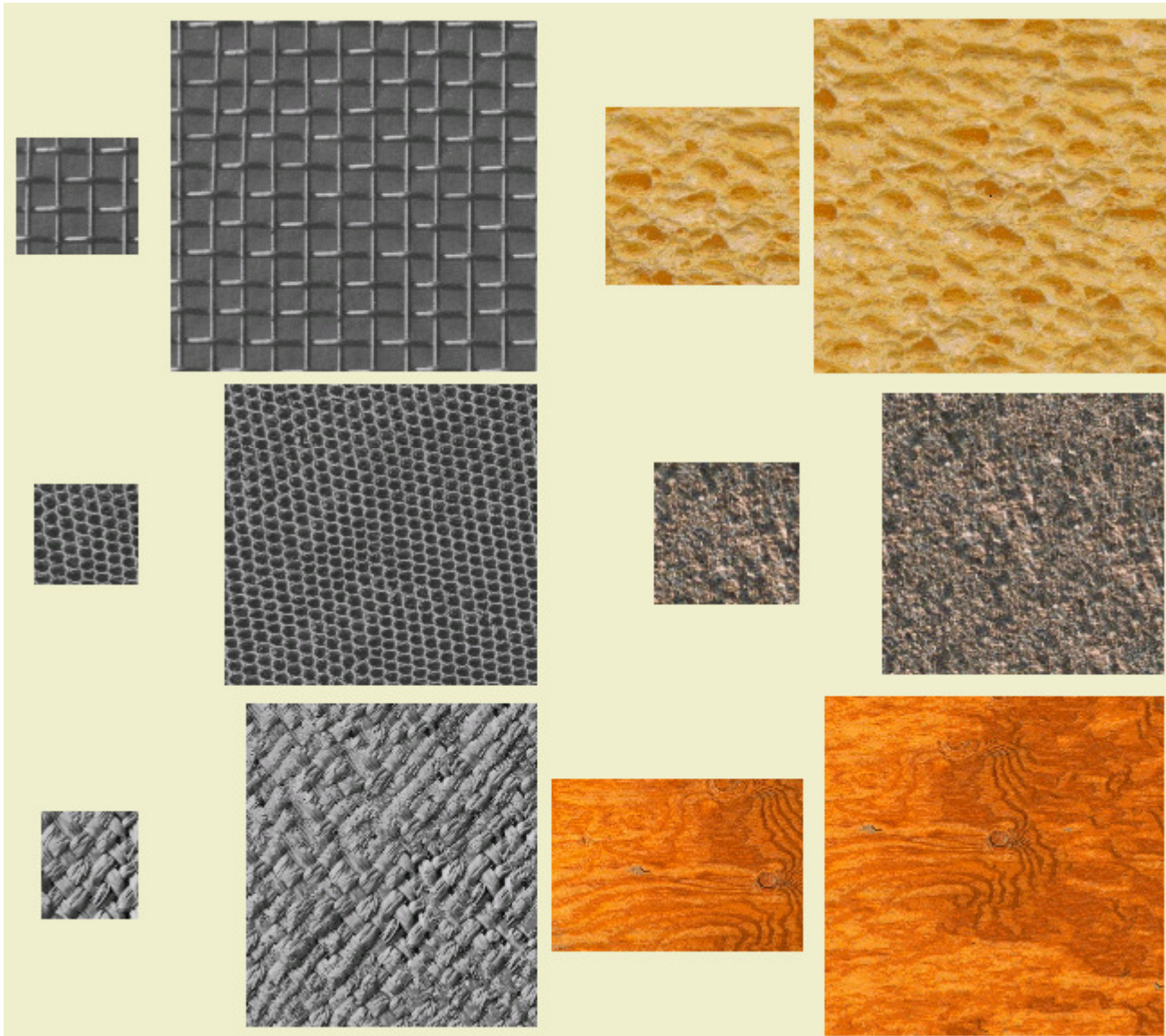
Both?

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



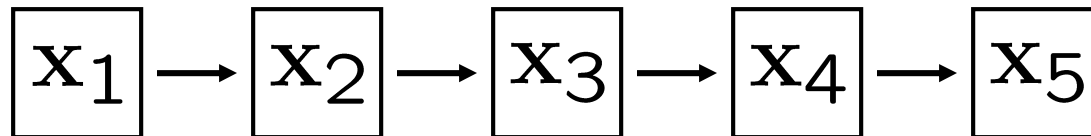
Efros and Leung



Markov Chains

Markov Chain

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



Markov Chain Example: Text

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

\mathbf{x}_{t-1}

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					
a	dog	is	man’s	best	friend	it’s	eat	world	out	there	.

\mathbf{x}_t

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

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

WE NEED TO EAT CAKE

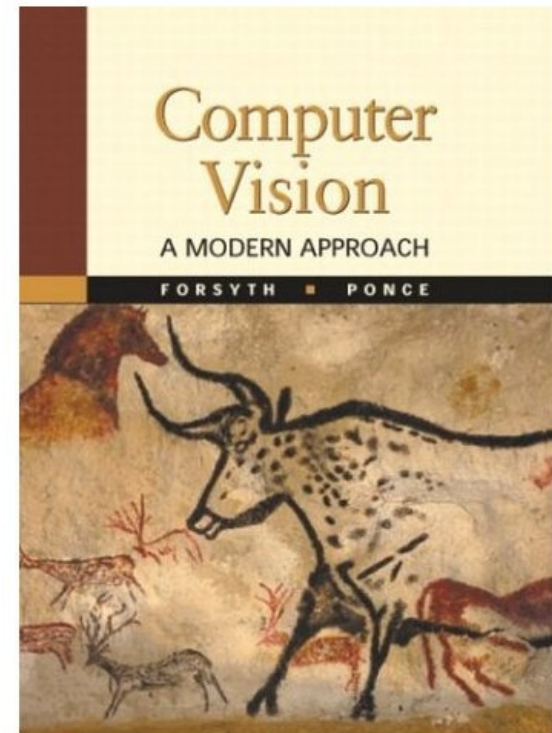
Text synthesis

- Results:
 - *“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.

Synthesizing Computer Vision text

- What do we get if we extract the probabilities from the F&P 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>

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).

Markov Random Field

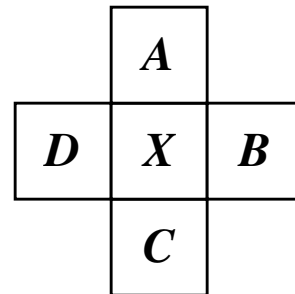
A Markov random field (MRF)

- generalization of Markov chains to two or more dimensions.

First-order MRF:

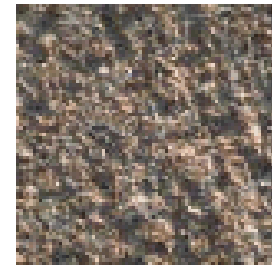
- probability that pixel X takes a certain value given the values of neighbors A , B , C , and D :

$$P(X|A, B, C, D)$$



Texture Synthesis [\[Efros & Leung, ICCV 99\]](#)

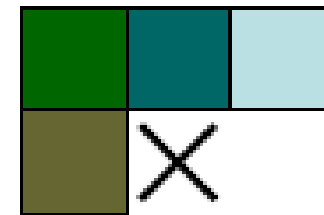
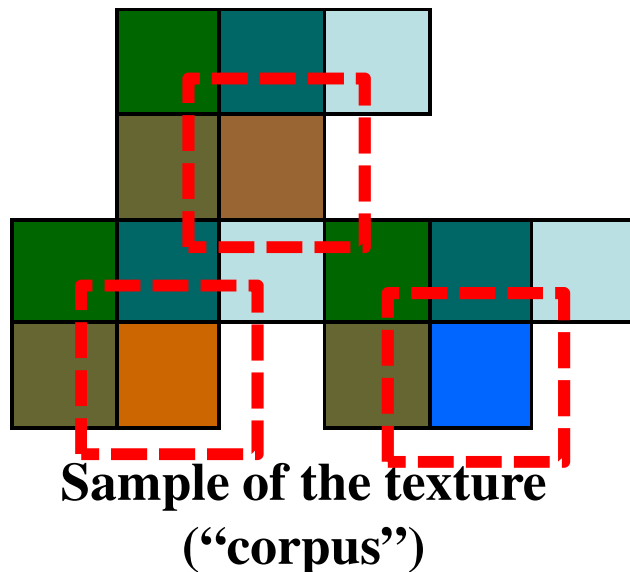
Can apply 2D version of text synthesis



Texture synthesis: intuition

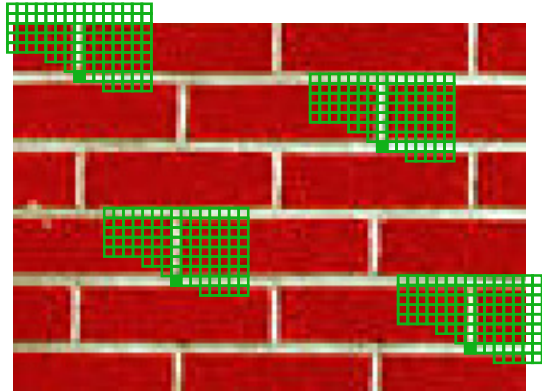
Before, we inserted the next word based on existing nearby words...

Now we want to insert **pixel intensities** based on existing nearby pixel values.

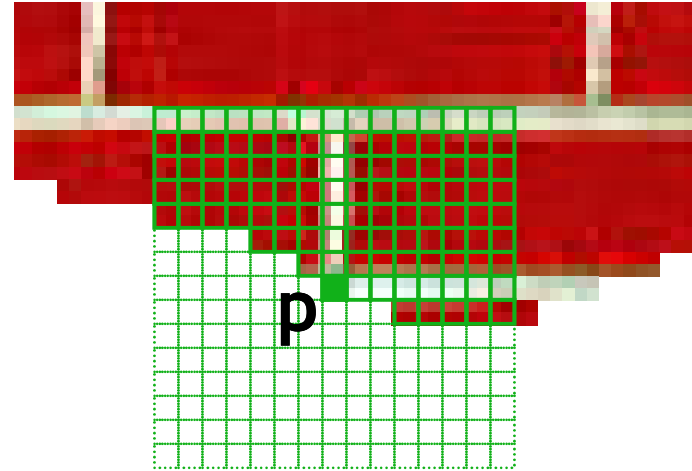


Distribution of a value of a pixel is conditioned on its neighbors alone.

Synthesizing One Pixel



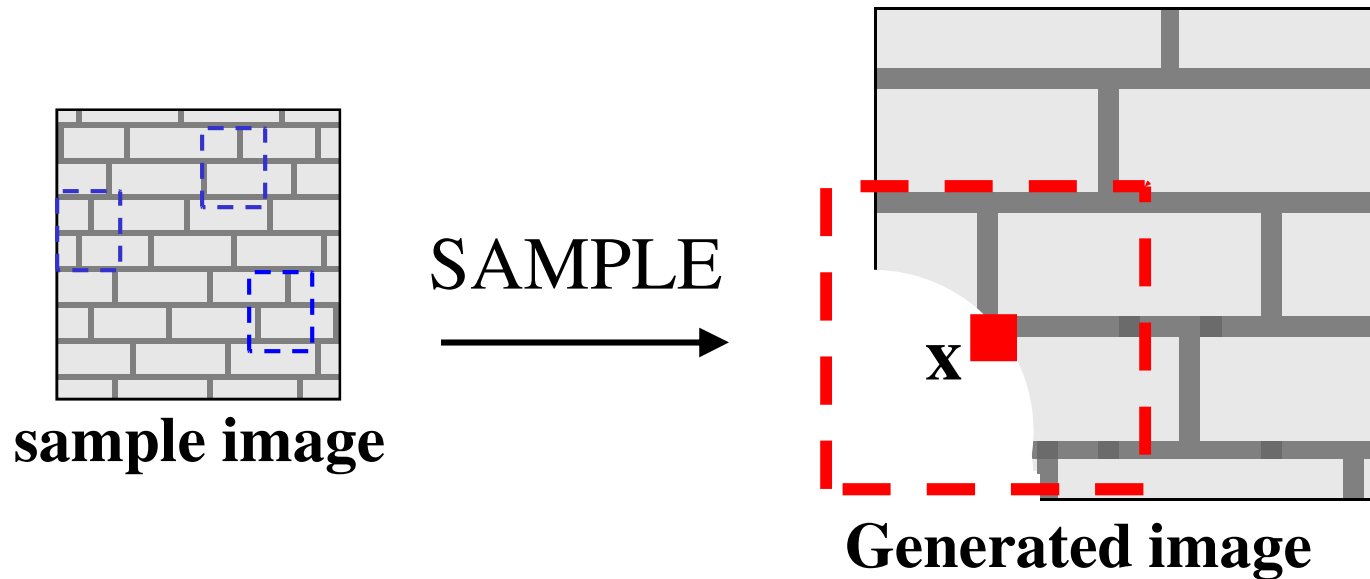
input image



synthesized image

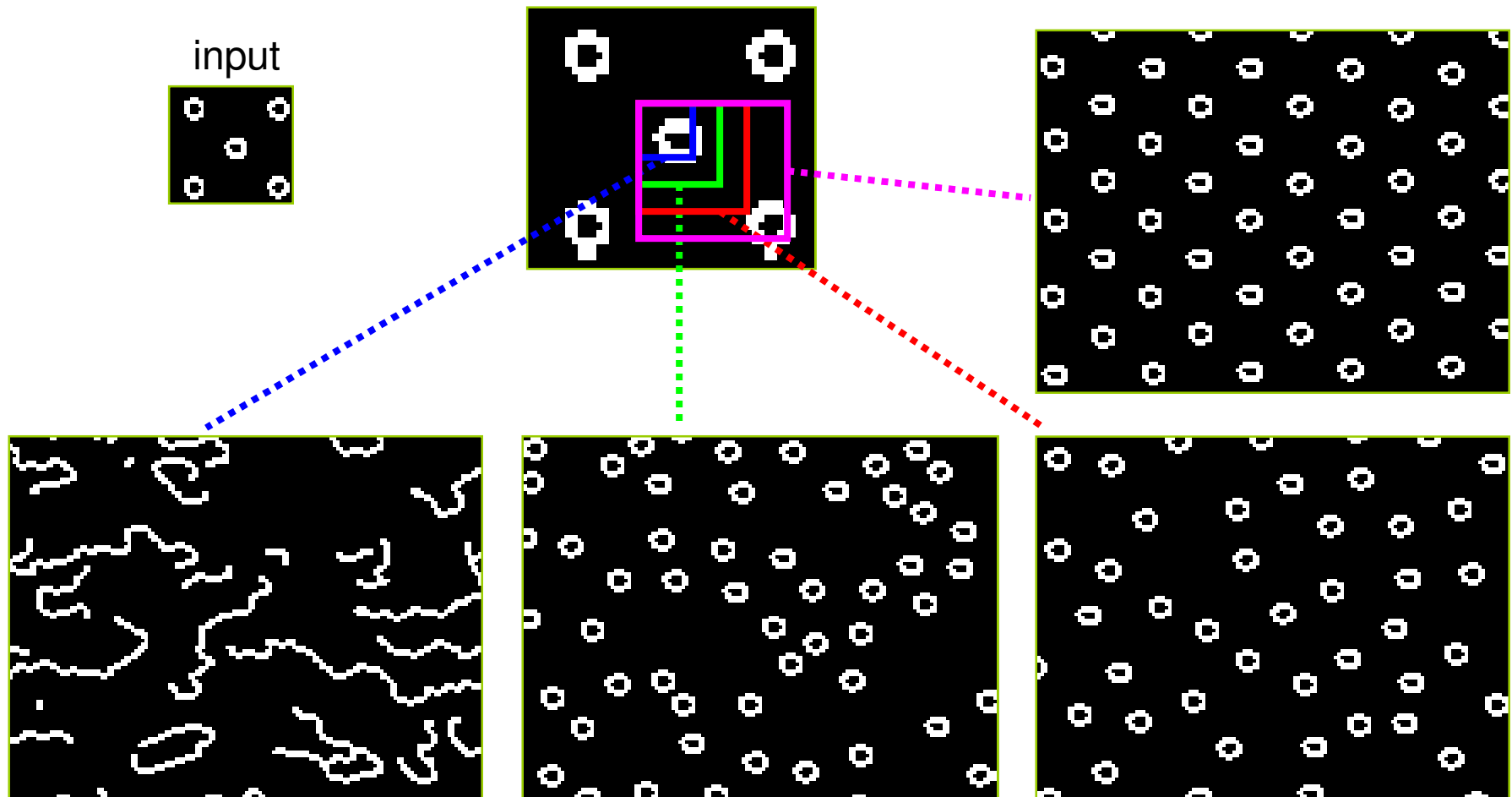
- What is $P(\mathbf{x}|\text{neighborhood of pixels around } \mathbf{x})$?
- Find all the windows in the image that match the neighborhood
 - consider only pixels in the neighborhood that are already filled in
- To synthesize \mathbf{x}
 - pick one matching window at random
 - assign \mathbf{x} to be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

Really Synthesizing One Pixel

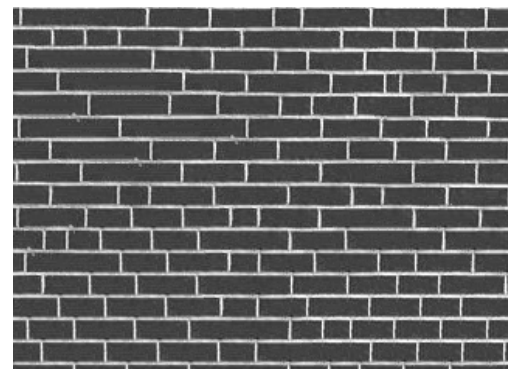
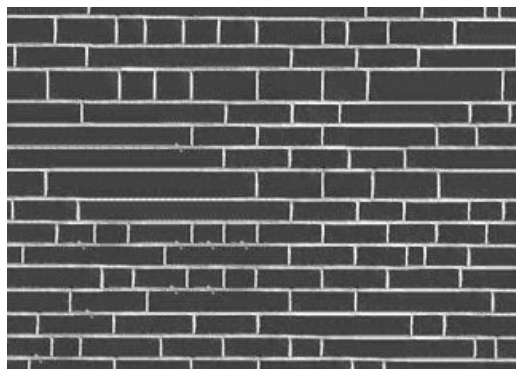
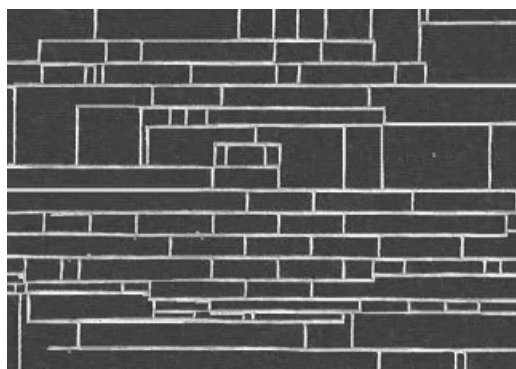
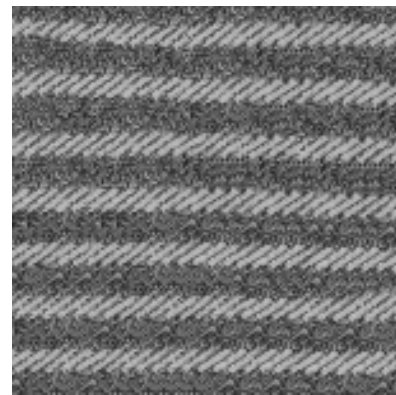
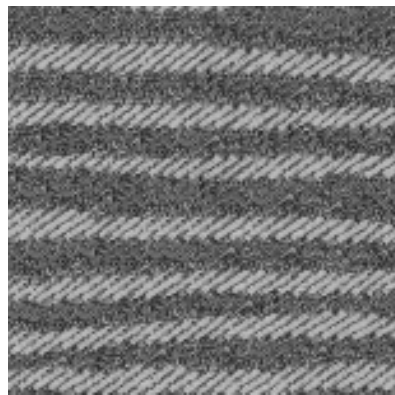
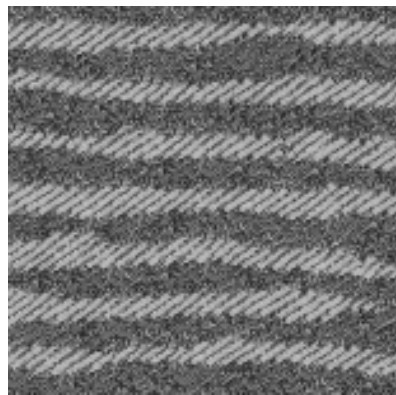
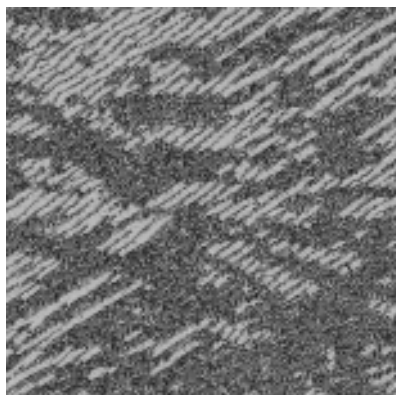


- An exact neighbourhood match might not be present
- So we find the **best** matches using SSD error and randomly choose between them, preferring better matches with higher probability

Neighborhood Window



Varying Window Size

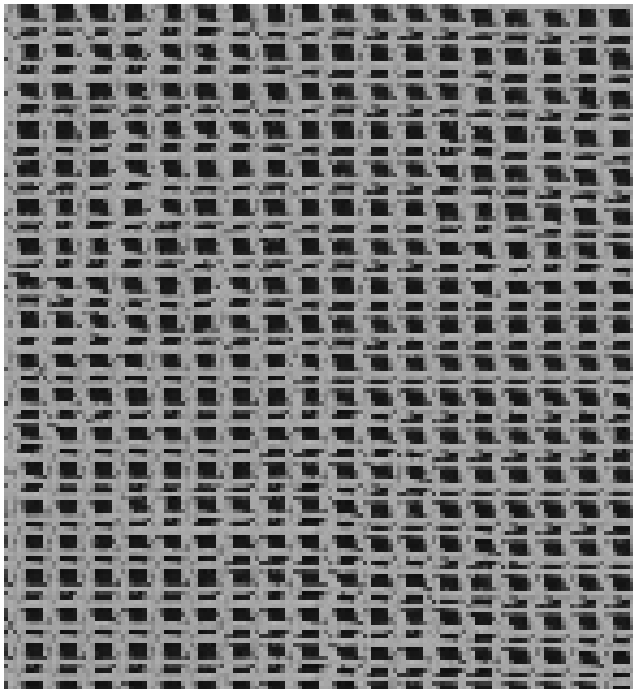
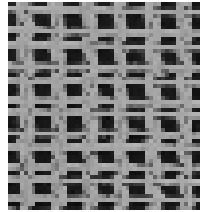


Increasing window size

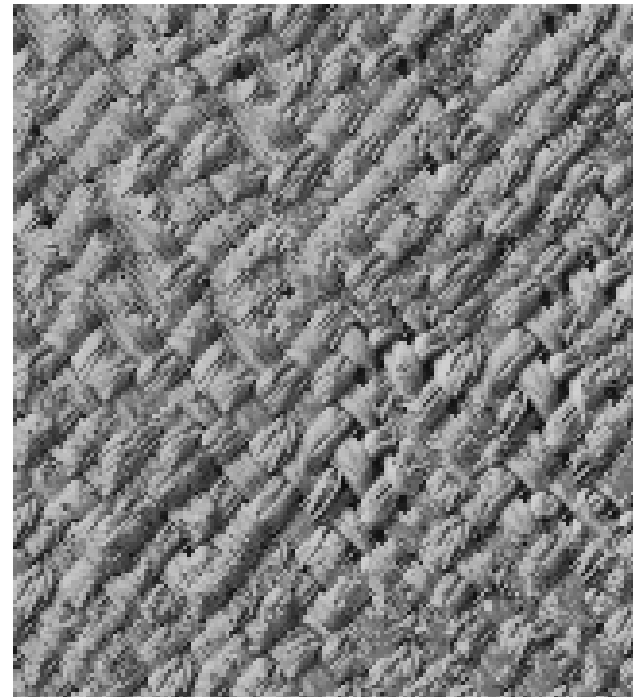
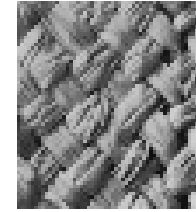


Synthesis results

french canvas



rafia weave

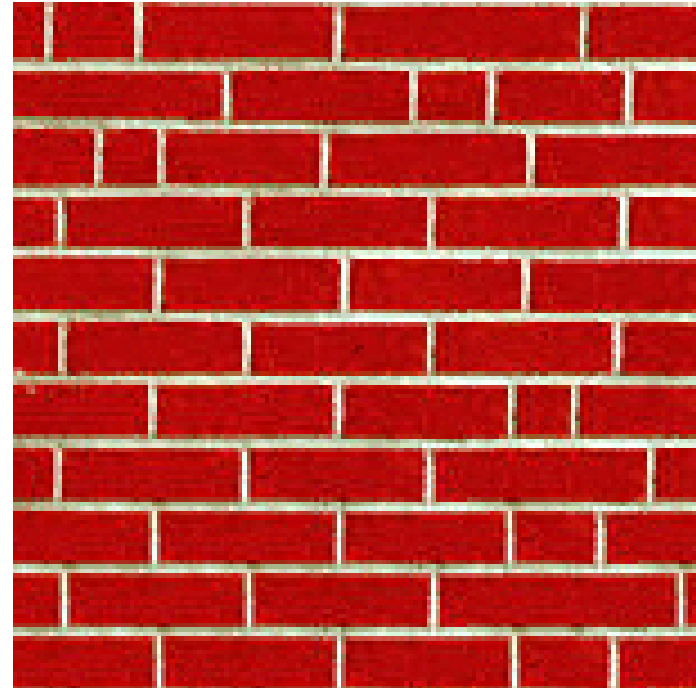
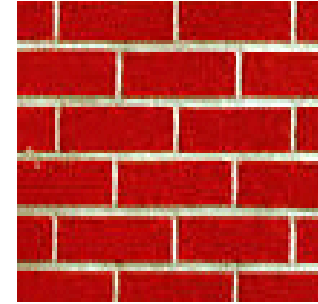


Synthesis results

white bread



brick wall

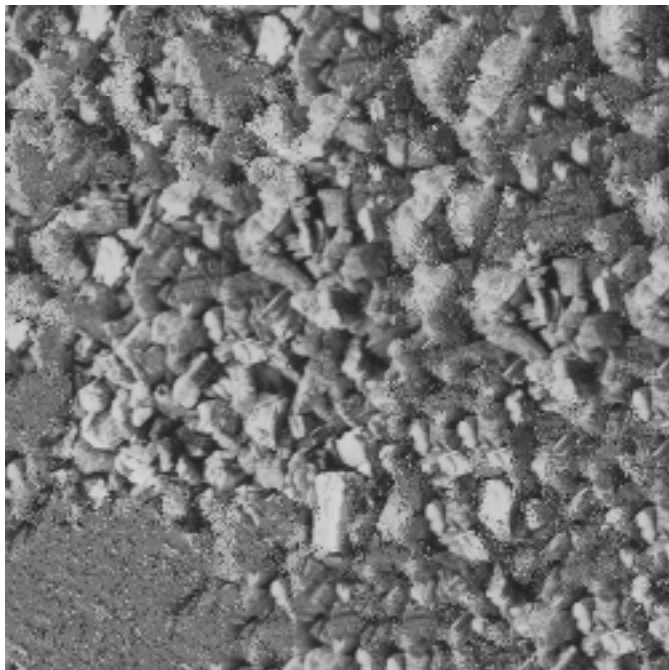
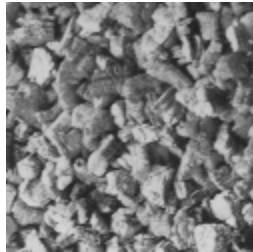


Synthesis results

...ing in the unsensadon
r Dick Gephardt was fai
rful riff on the looming
nly asked, "What's your
tions?" A heartfelt sigh
story about the emergen
es against Clinton. "Boy
g people about continuin
ardt began, patiently obs
s, that the legal system h
g with this latest tanger

ithaim. them . "Whnephartfe lartifelintomimen.
lel ck Clirtioout omaim thartfelins.f out s anetc
the ry onst wartfe lck Gephtoomimeationl sigab
Chiooufit Clinut Clil riff on. hat's yodn, parut tly:
ons yontonsteht waked, paim t sahe loo riff on l
nskoneploourtfeas leil A nst Clit, "Wheontongal s
k Cirtioouirtfepe.ong pme abegal fartfenstemem
tiensteneltorydt telemephminsverdt was agemer
ff ons artientont Cling peme asartfe atih, "Boui s
nal s fartfelt sig pedrtldt ske abounutie aboutioo
tfaonewas yow abowonthardt thatins fain, ped, '
ains. them, pabout wasy arfint coutly d, l n A h
ble emthringbooreme agas fa bontinsyst Clinut
ory about continst Clipeopinst Cloke agatiff out C
stome zinemen fly ardt beorabou n, thenly as t C
cons faimeme Diontont wat coutlyohgans as fan
ien, phrtfaul, "Wbaut cout congagal comininga:
mifmst Cliiy abon al coountha.emungaint tf ouu
The loocrysta loontieph. intly on, theoplegatick C
ul fatieeontly atie Diontiomt wal s f tbegeae ener
nthahgat's enenhhmas fan. "intchthorv abons w

Failure Cases

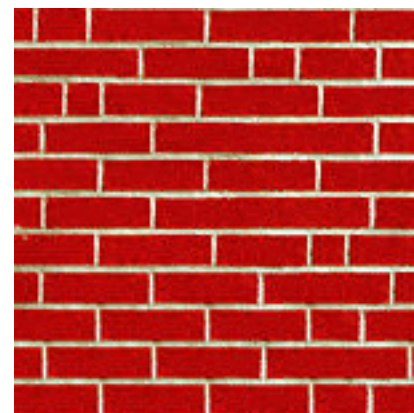


Growing garbage

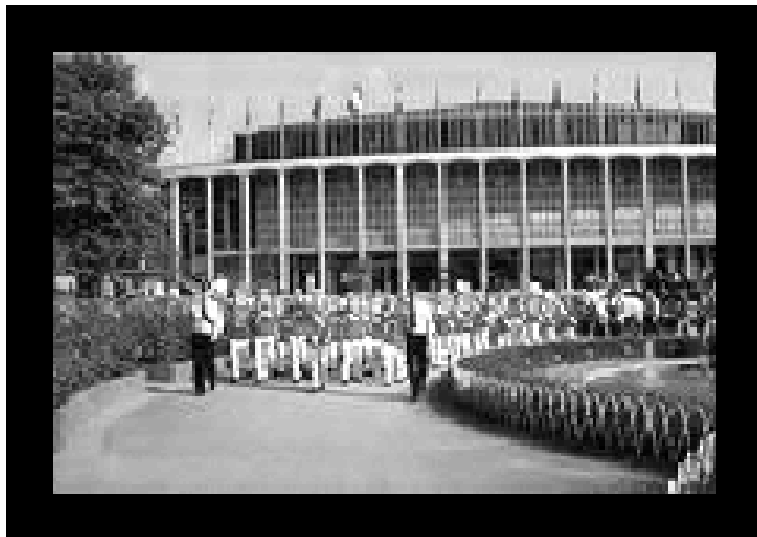
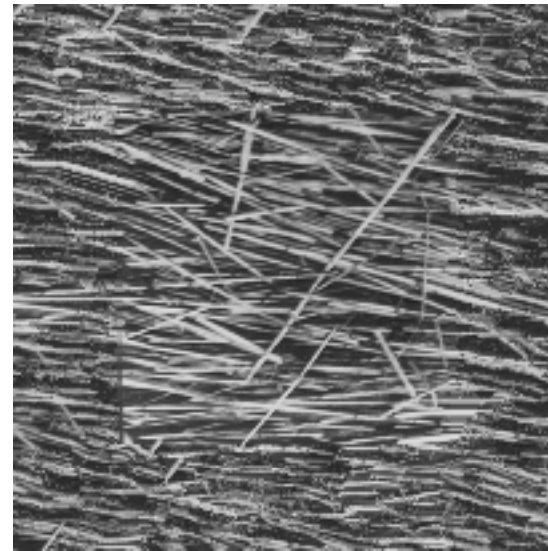
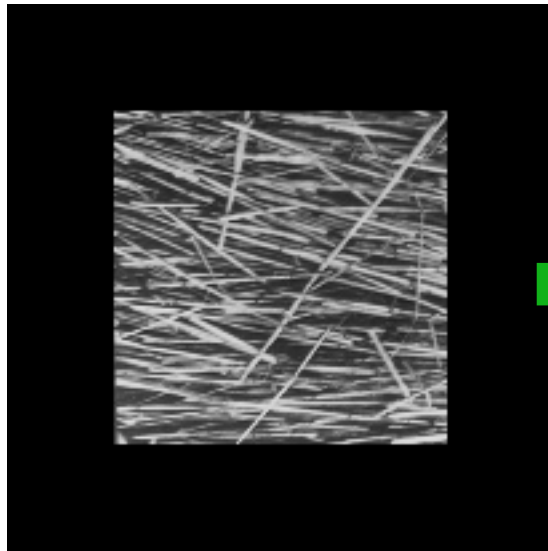


Verbatim copying

Hole Filling



Extrapolation



Slide from Alyosha Efros, ICCV 1999

Recall: What we had learned from the first four texture synthesis methods

From Adelson and Bergen:

examine filter outputs

From Perona and Malik:

use multi-scale, multi-orientation filters.

From Heeger and Bergen:

use marginal statistics (histograms) of filter responses.

From DeBonet:

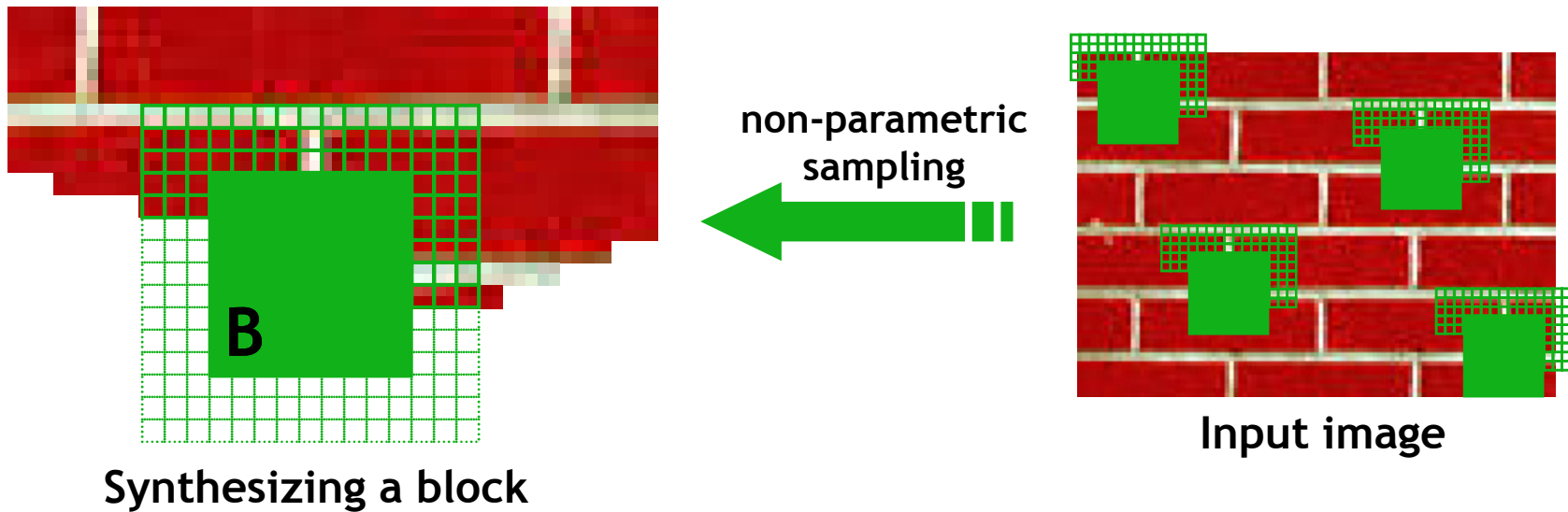
use conditional filter responses across scale.

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.

- The Efros & Leung algorithm
 - Simple
 - Surprisingly good results
 - Synthesis is easier than analysis!
 - ...but very slow

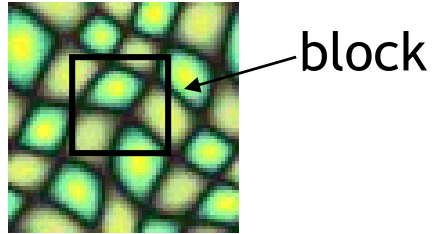
Image Quilting [Efros & Freeman 2001]



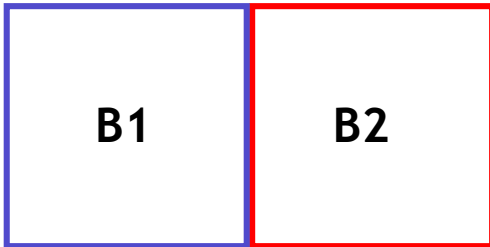
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

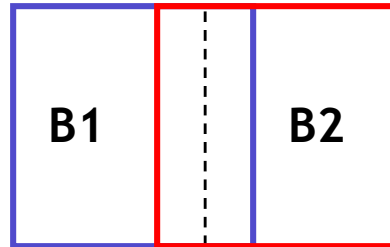
- Exactly the same but now we want $P(B|N(B))$
- Much faster: synthesize all pixels in a block at once



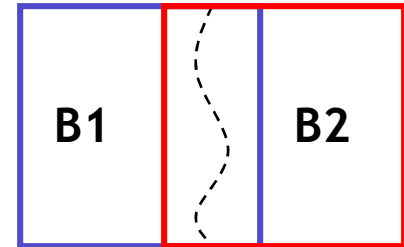
Input texture



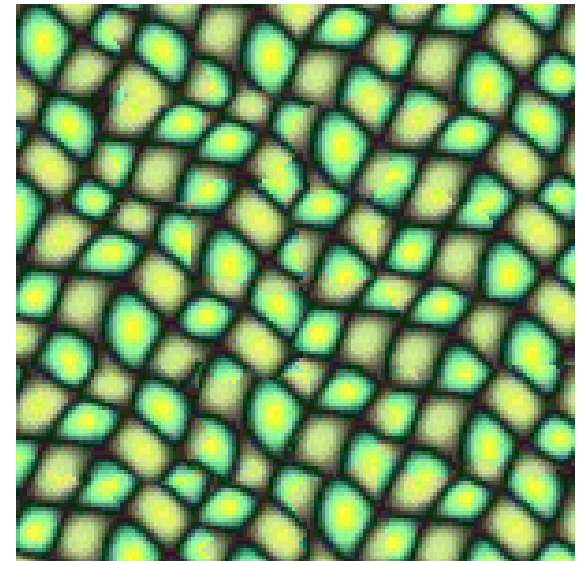
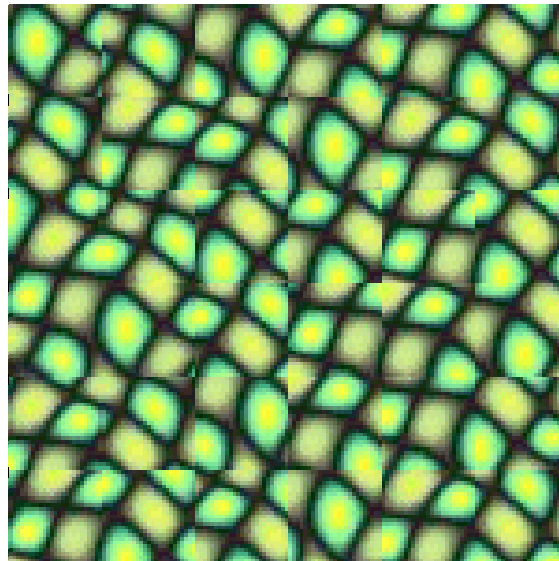
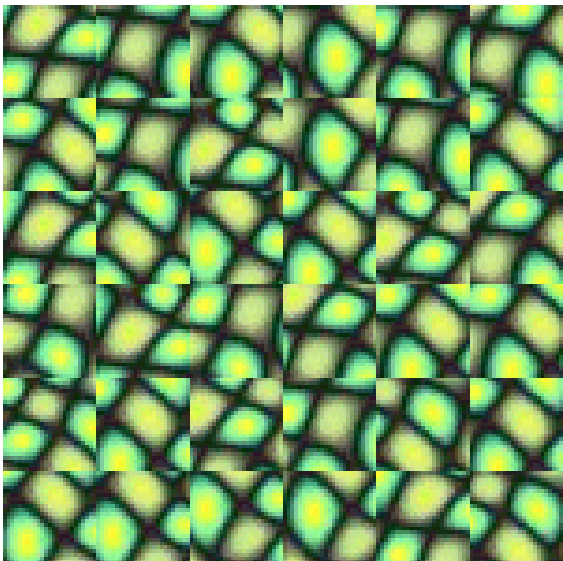
Random placement of blocks



Neighboring blocks constrained by overlap

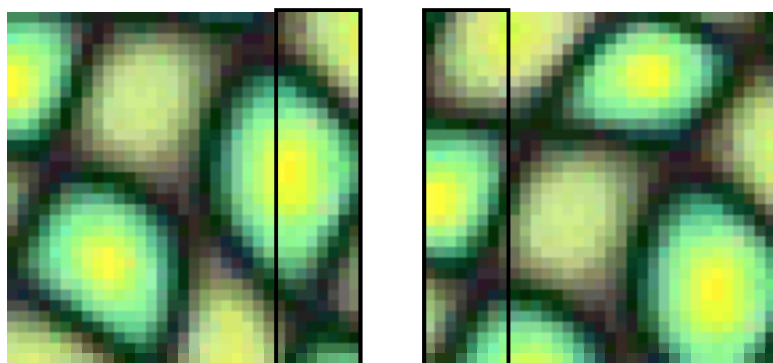


Minimal error boundary cut

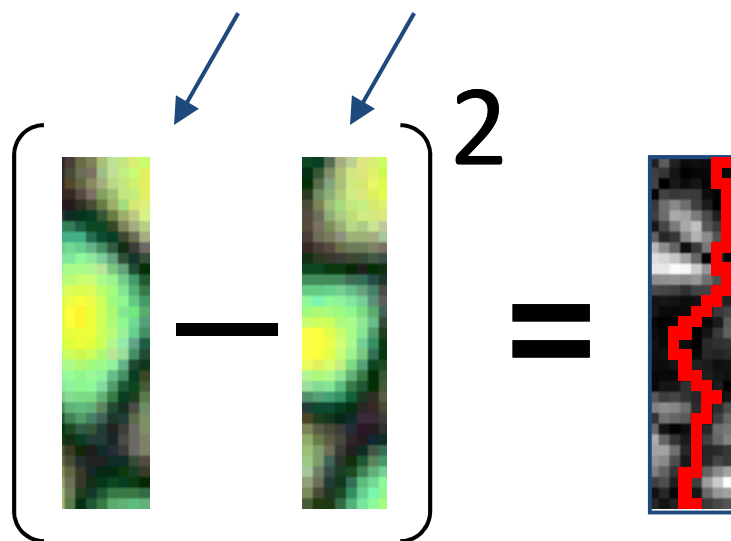
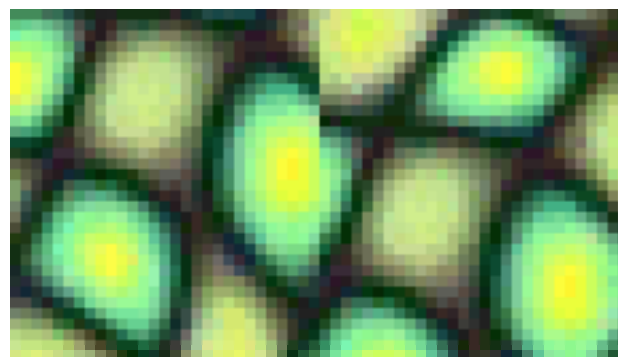


Minimal error boundary

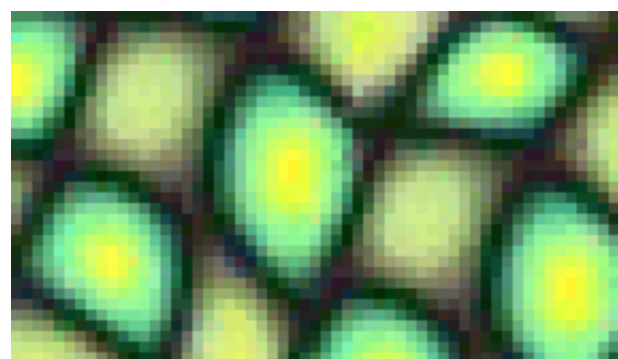
overlapping blocks



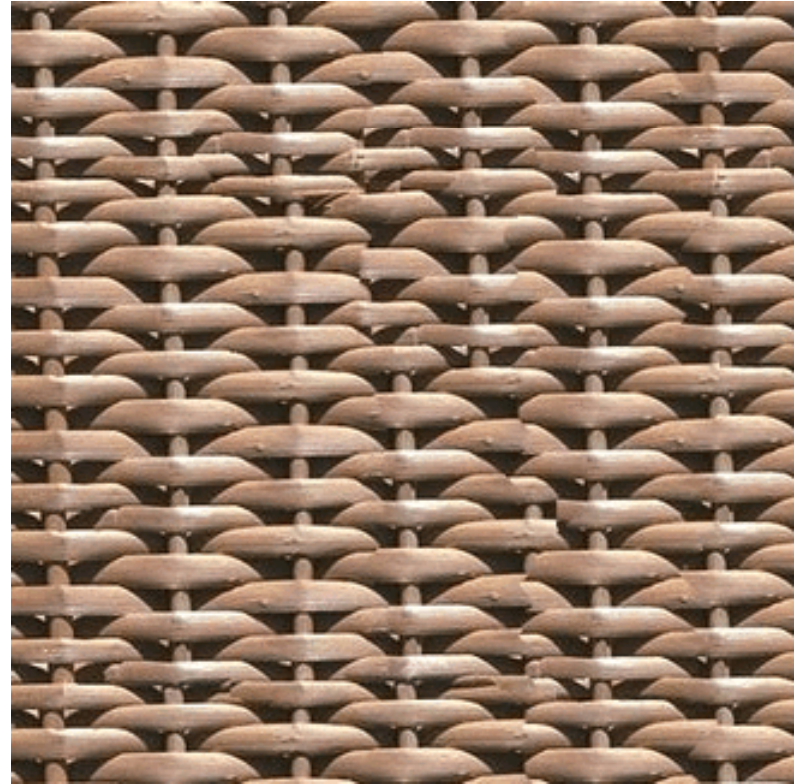
vertical boundary

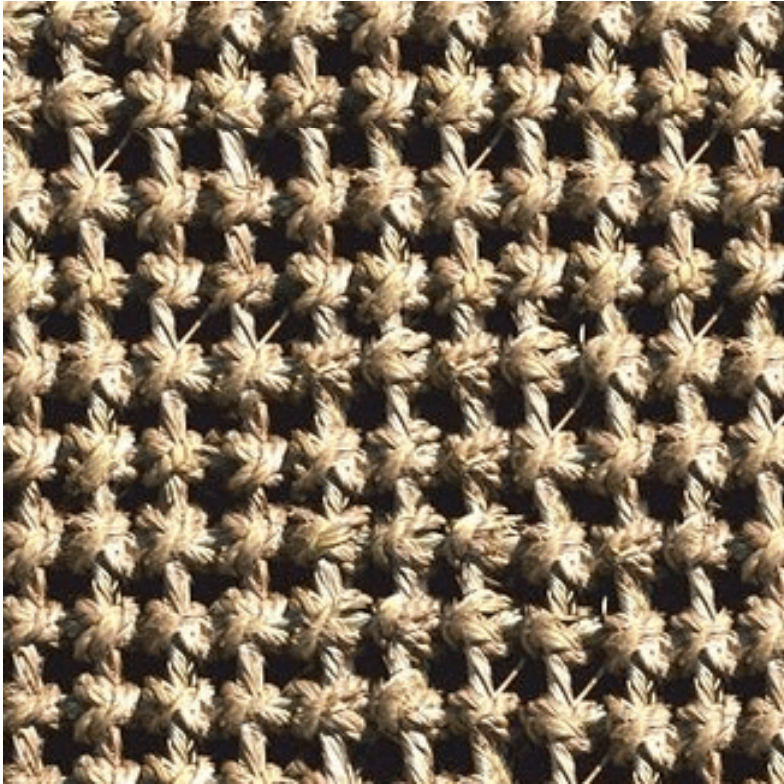
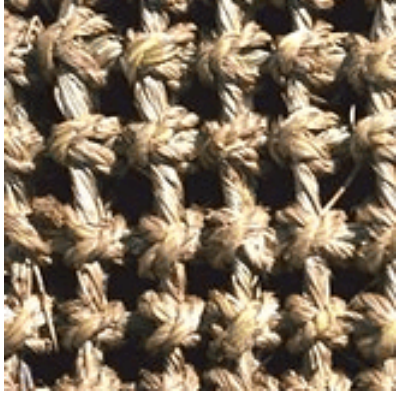


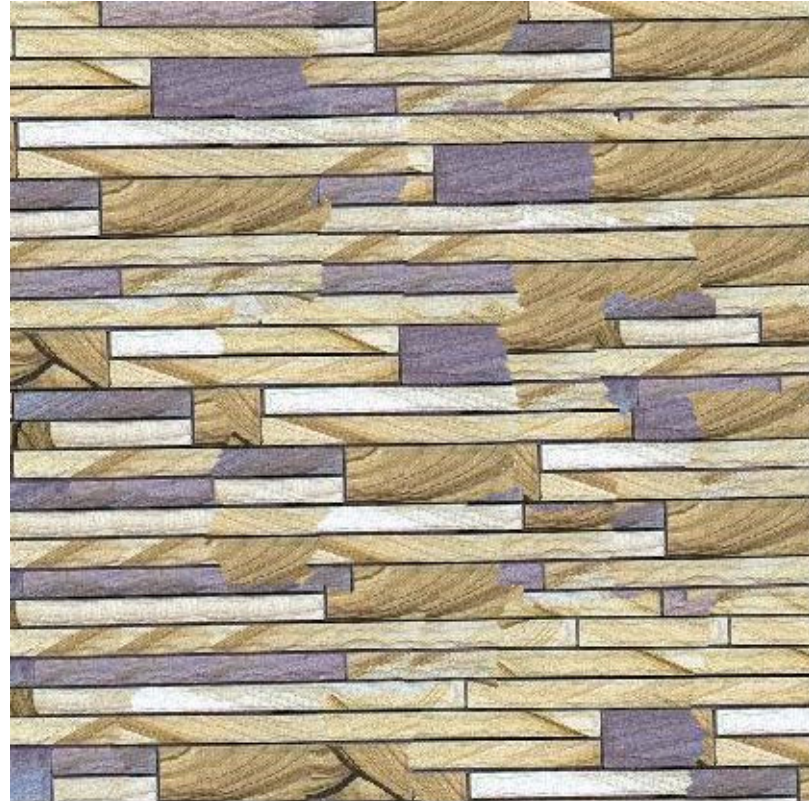
overlap error

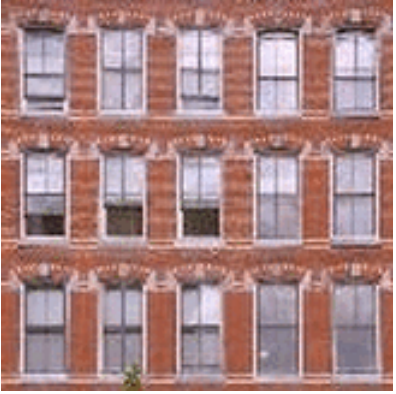


min. error boundary

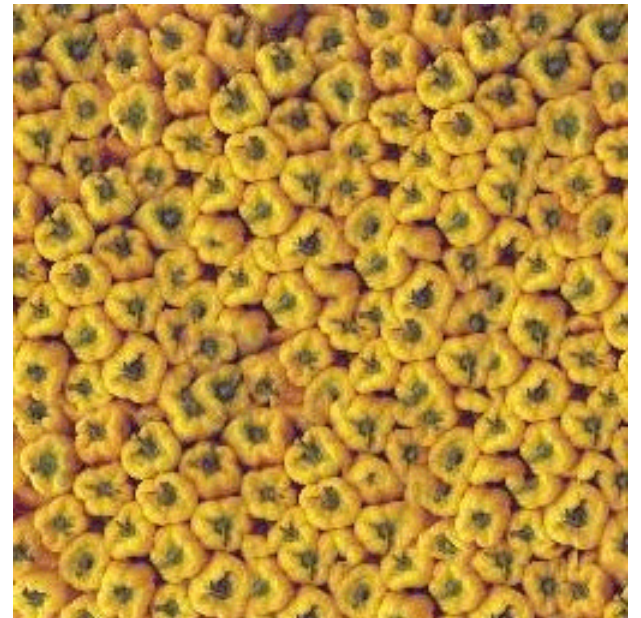
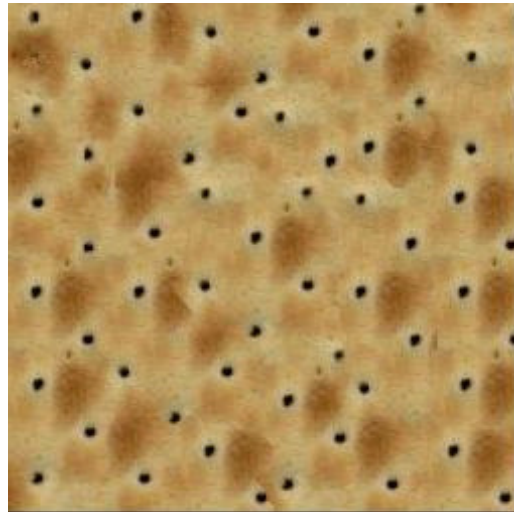
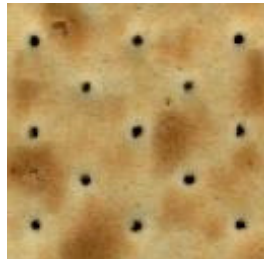








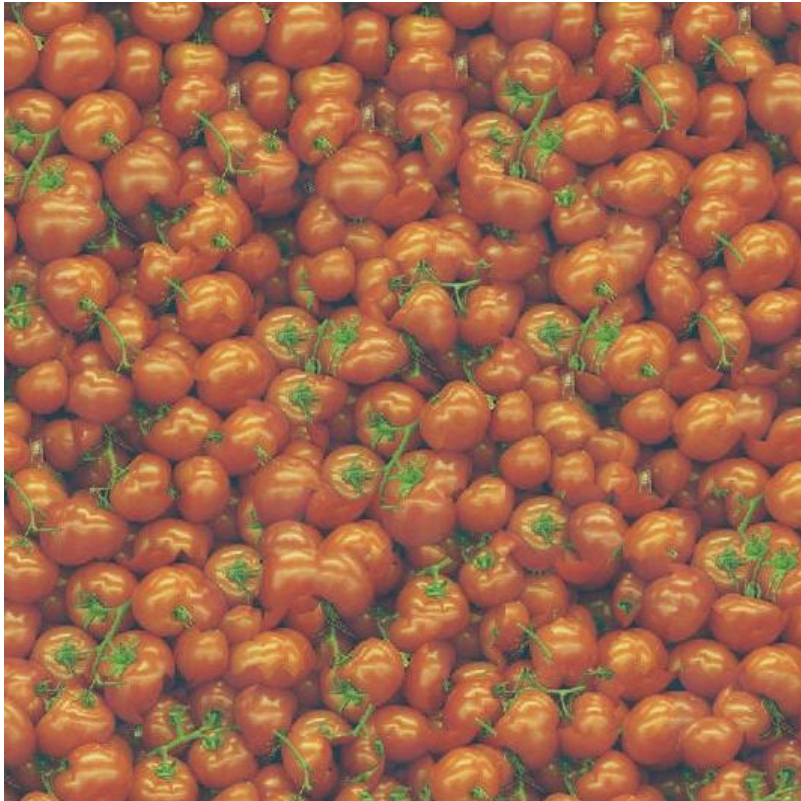
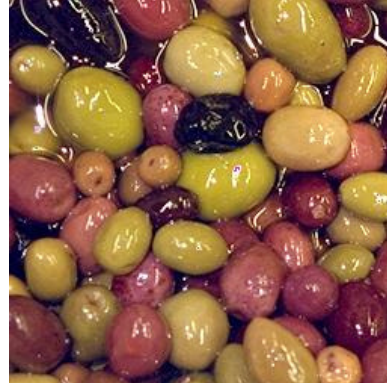






Failures

(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

+



=



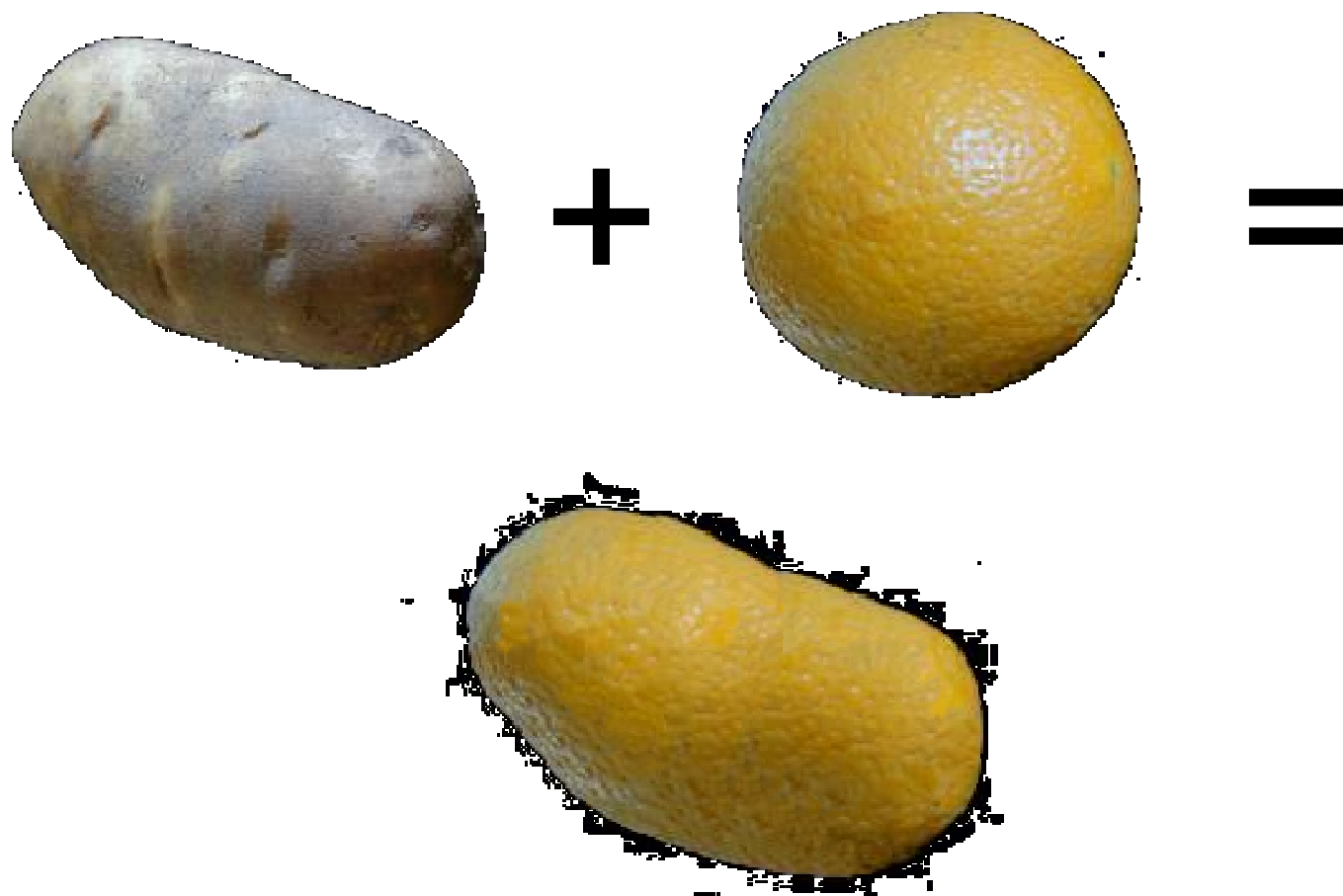
rice

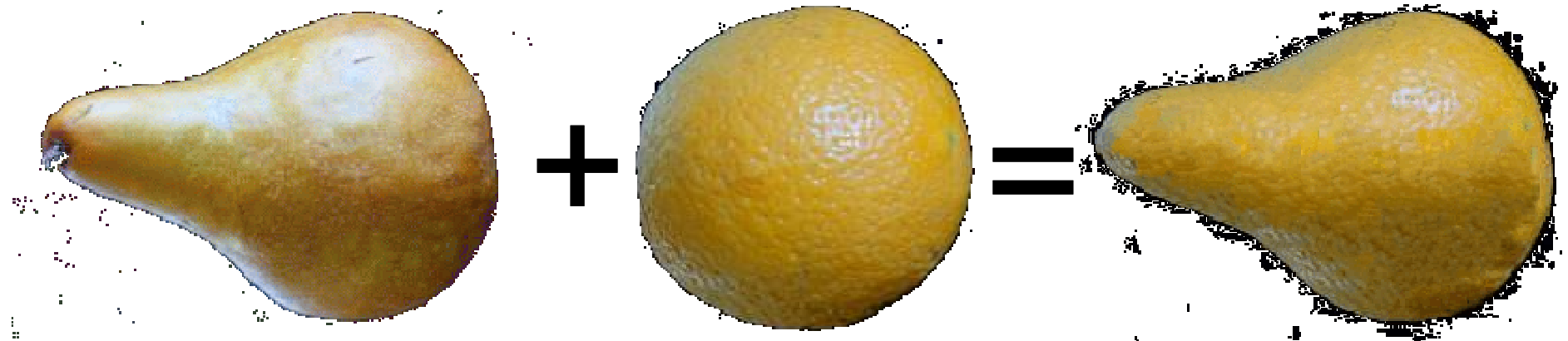
+



=





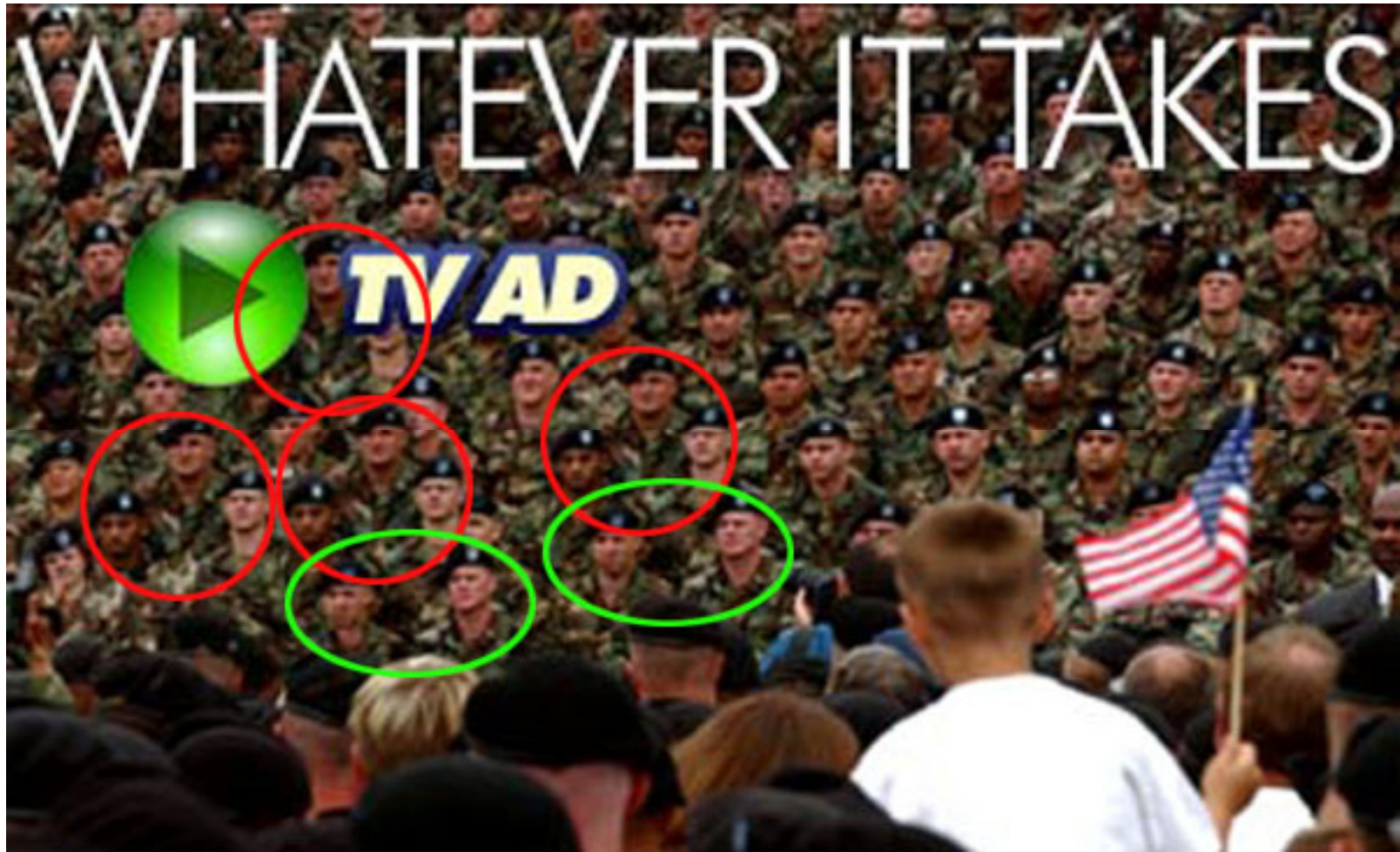


(Manual) texture synthesis in the media

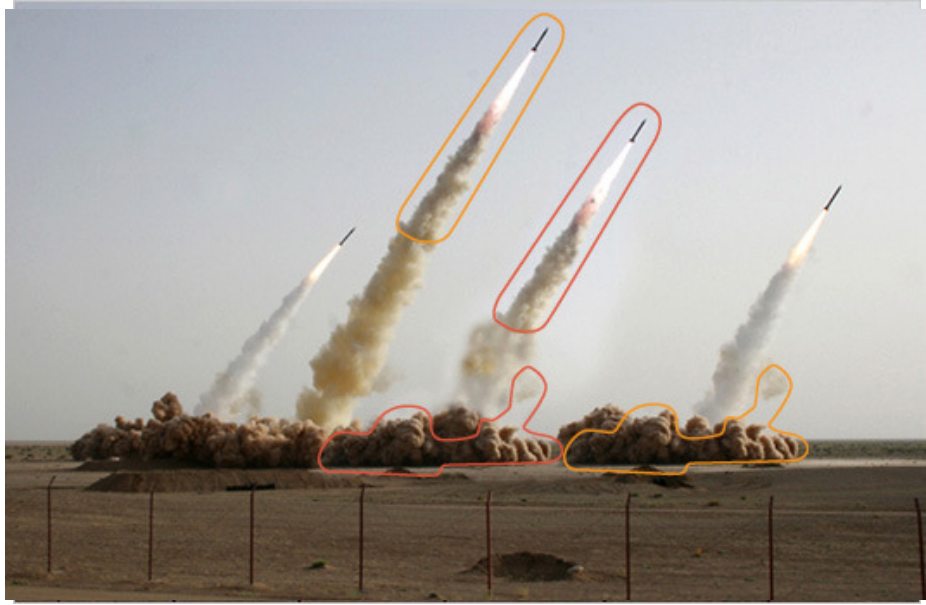


(Manual) texture synthesis in the media





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



<http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/>

Summary

- Texture is a useful property that is often indicative of materials, appearance cues
- **Texture representations** attempt to summarize repeating patterns of local structure
- **Filter banks** useful to measure redundant variety of structures in local neighborhood
 - Feature spaces can be multi-dimensional
- Neighborhood statistics can be exploited to “sample” or **synthesize** new texture regions
 - Example-based technique particularly good for synthesis

Significant overlap in recent literature on texture and object recognition.... More later in course...

Slide Credits

- Steve Seitz
- Bill Freeman
- Kristen Grauman
- Alyosha Efros

Next time: Image Stitching



(a)



(b)



(c)