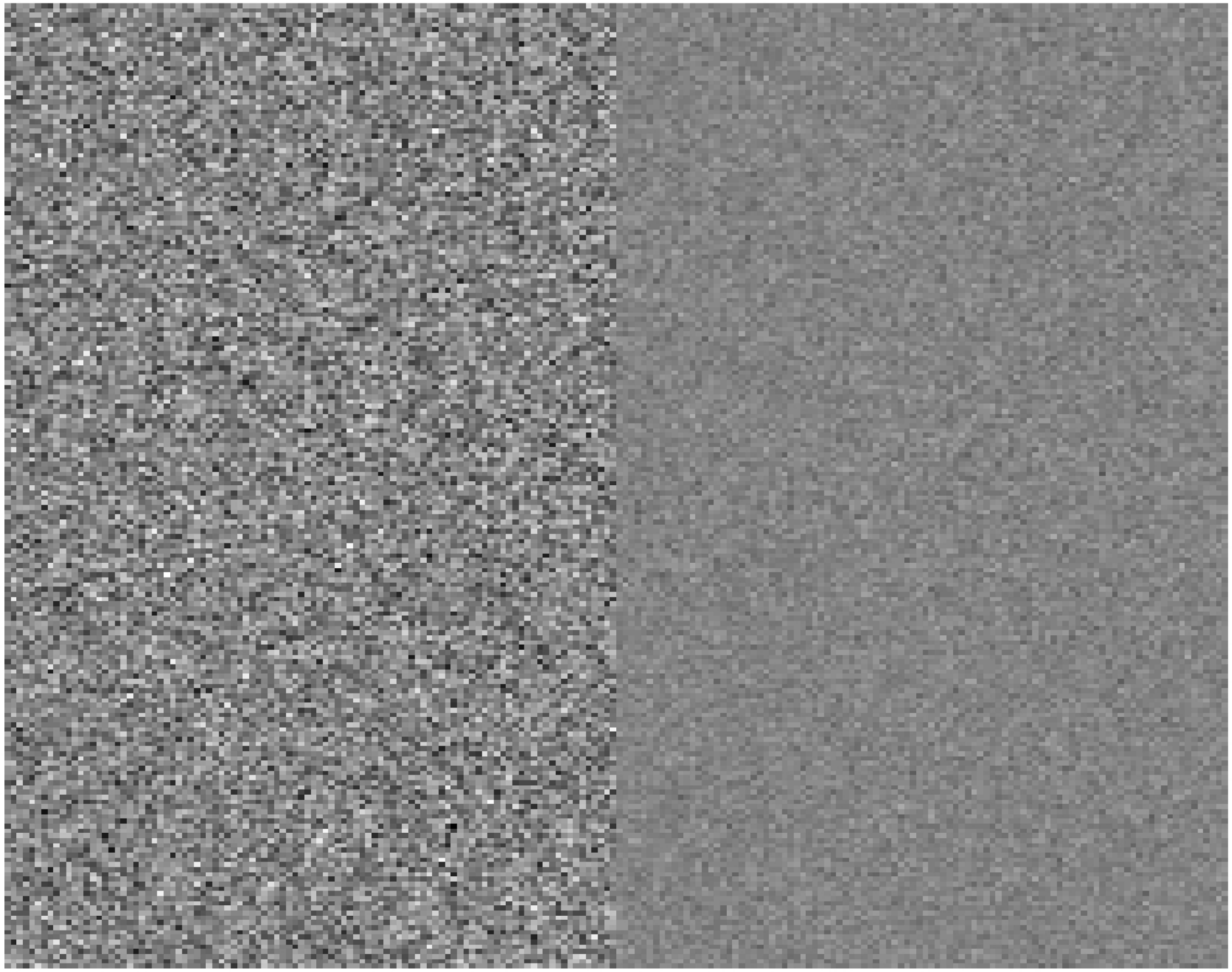
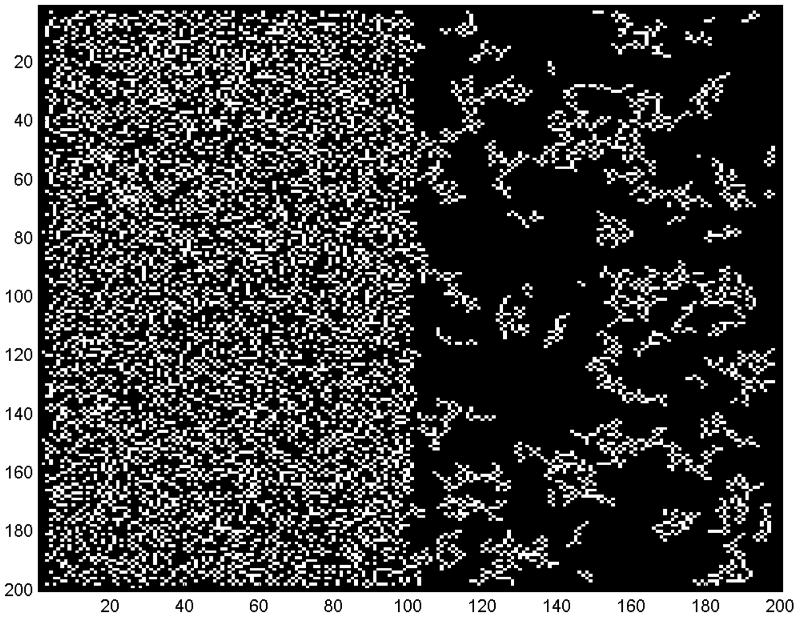
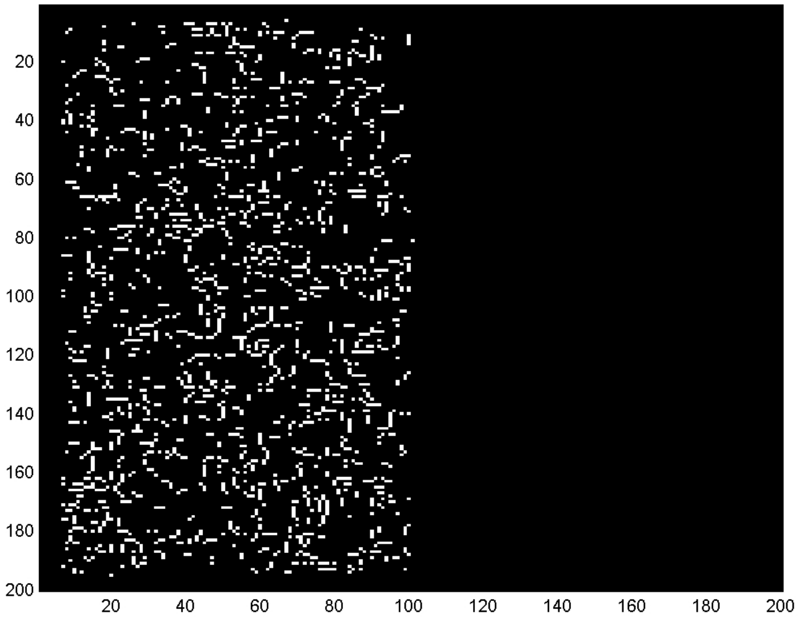
Texture

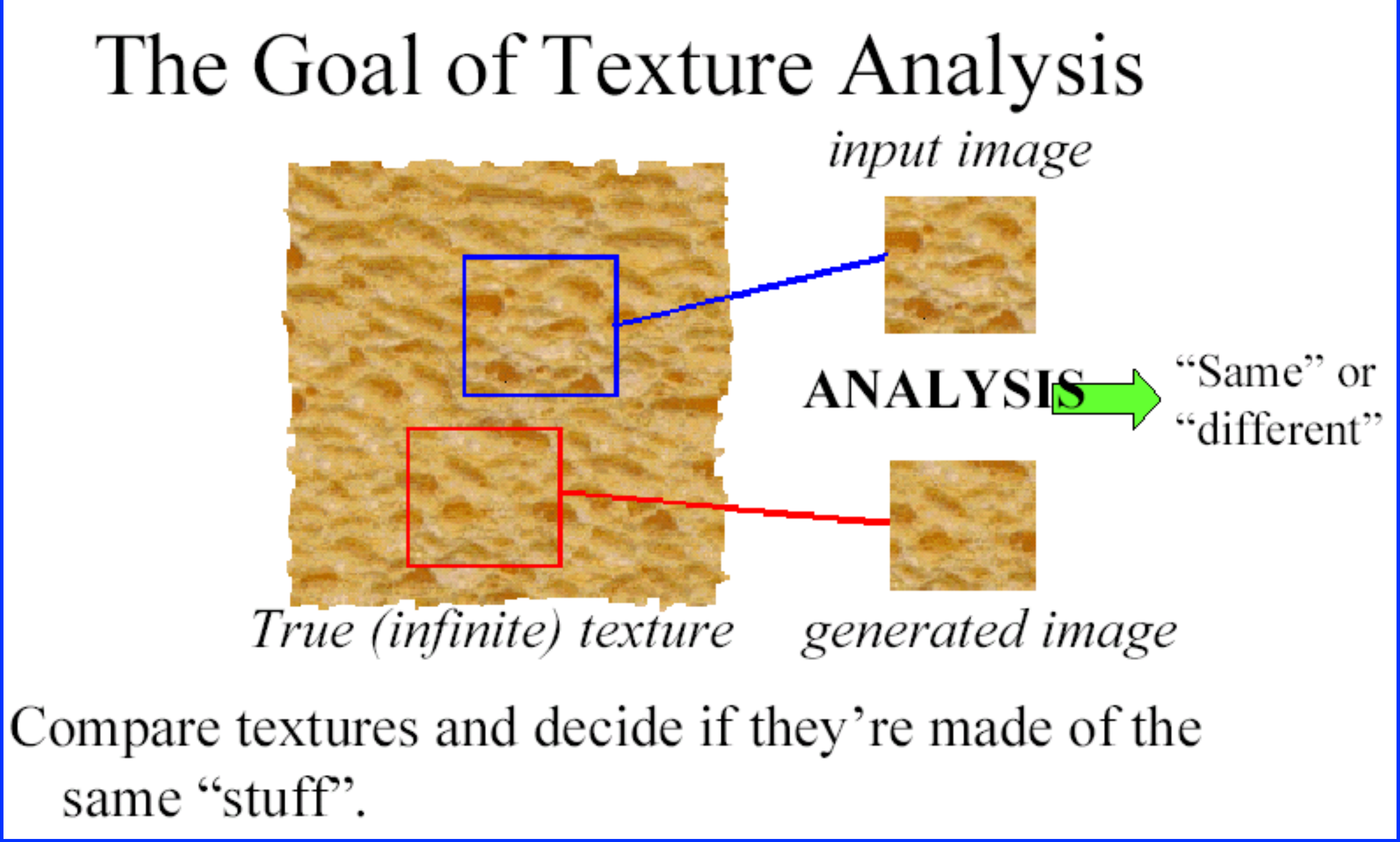
- Edge detectors find differences in overall intensity.
- Average intensity is only simplest difference.




20
40
60
80
100
120
140
160
180
200
20
40
60
80
100
120
140
160
180
200


20
40
60
80
100
120
140
160
180
200
20
40
60
80
100
120
140
160
180
200

Issues: 1) Discrimination/Analysis

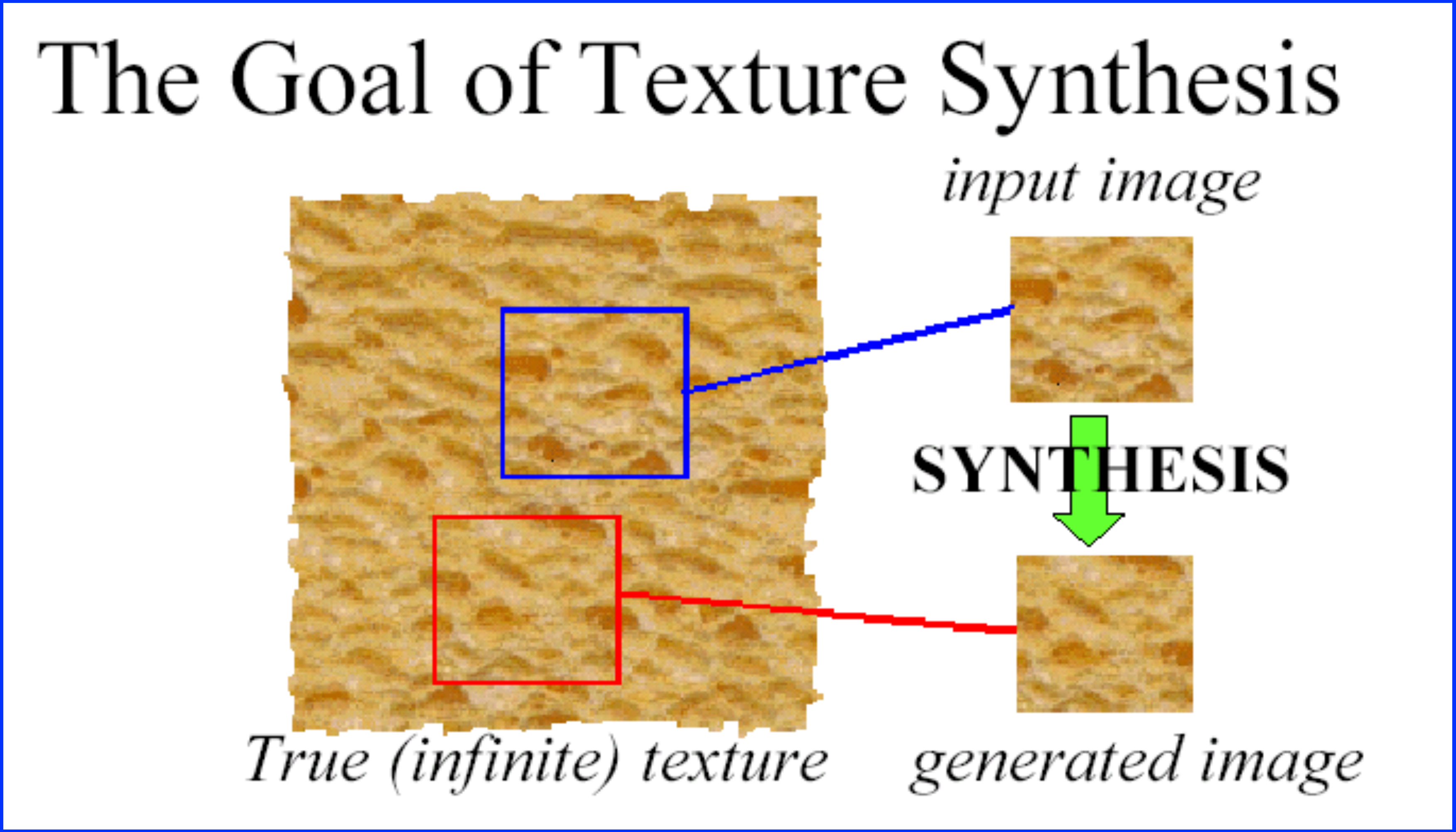


The Goal of Texture Analysis

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

(Freeman)

2) Synthesis



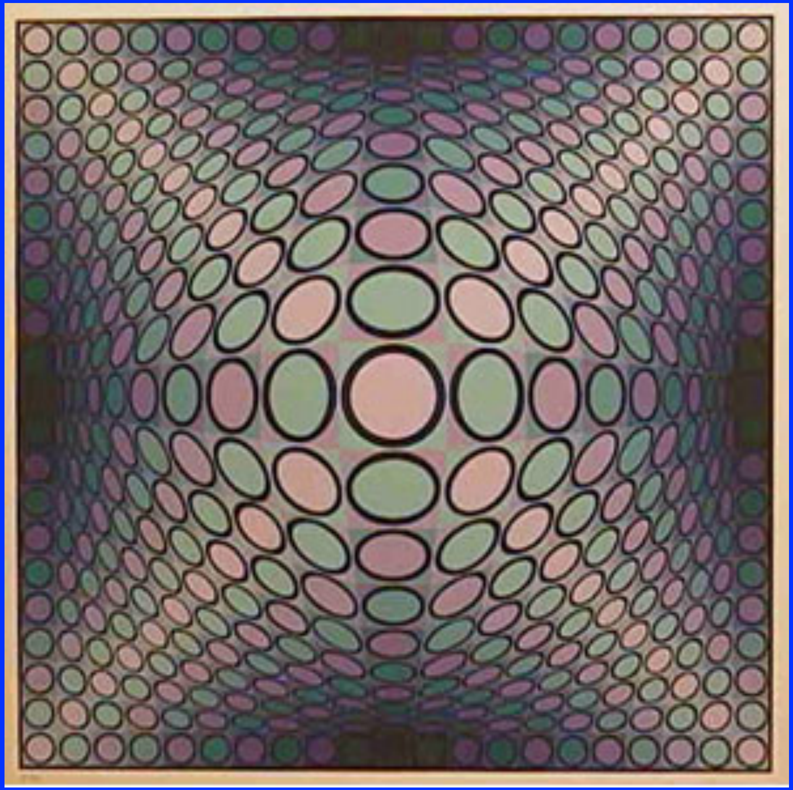
The Goal of Texture Synthesis

Many more issues

3. Texture boundary detection.

4. Shape from texture.

We’ll focus on 1 and 2.



(www.cmap.polytechnique.fr/
~maureen/vasarely3.jpg)

What is texture?

- Something that repeats with variation.
- Must separate what repeats and what stays the same.
- Model as repeated trials of a random process
  - The probability distribution stays the same.
  - But each trial is different.
  - This may be true (eg., pile of objects)
  - Or not really (tile floor).

Simplest Texture

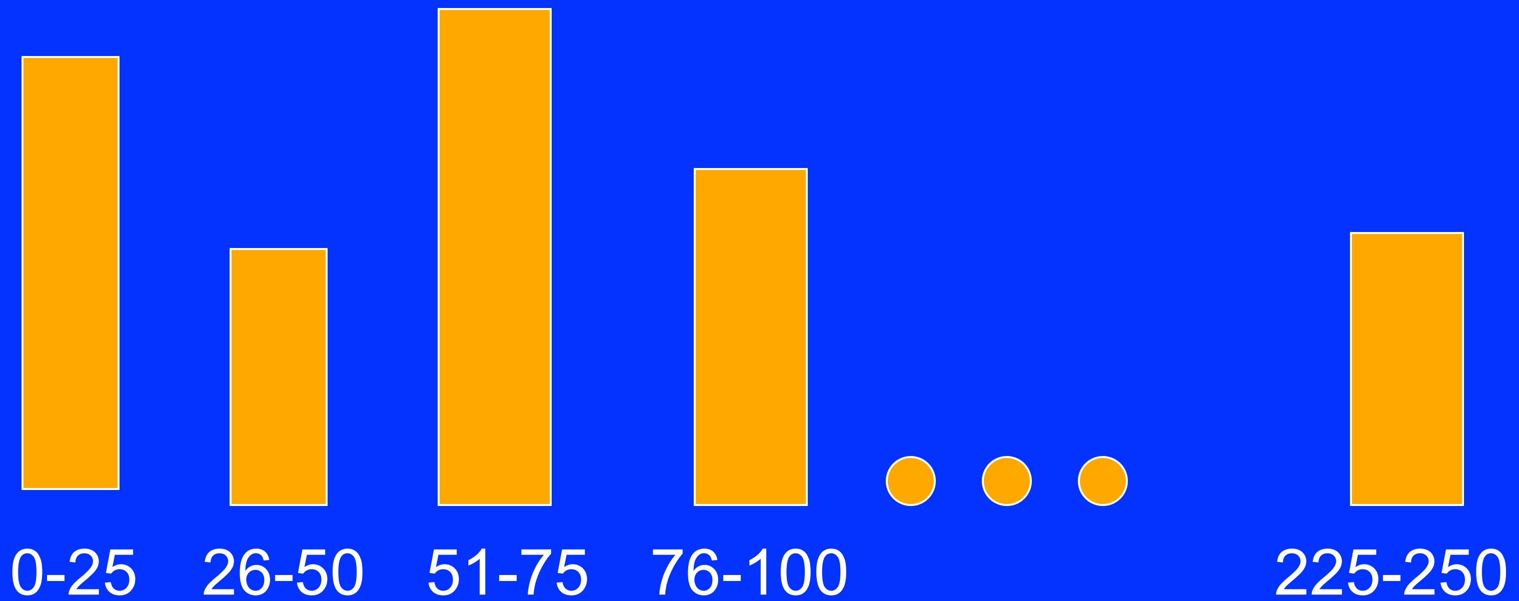
- Each pixel independent, identically distributed (iid).
- Examples:
  - Region of constant intensity.
  - Gaussian noise pattern.
  - Speckled pattern

Texture Discrimination is then Statistics

- Two sets of samples.
- Do they come from the same random process?

Simplest Texture Discrimination

- Compare sample distributions (histograms).
  - – Divide intensities into discrete ranges.
  - – Count how many pixels in each range.



How/why to compare

- Simplest comparison is SSD, many others.
- Can view probabilistically.
  - Histogram is a set of samples from a probability distribution.
  - With many samples it approximates distribution.
  - Test probability samples drawn from same distribution. Ie., is difference greater than expected when two samples come from same distribution?

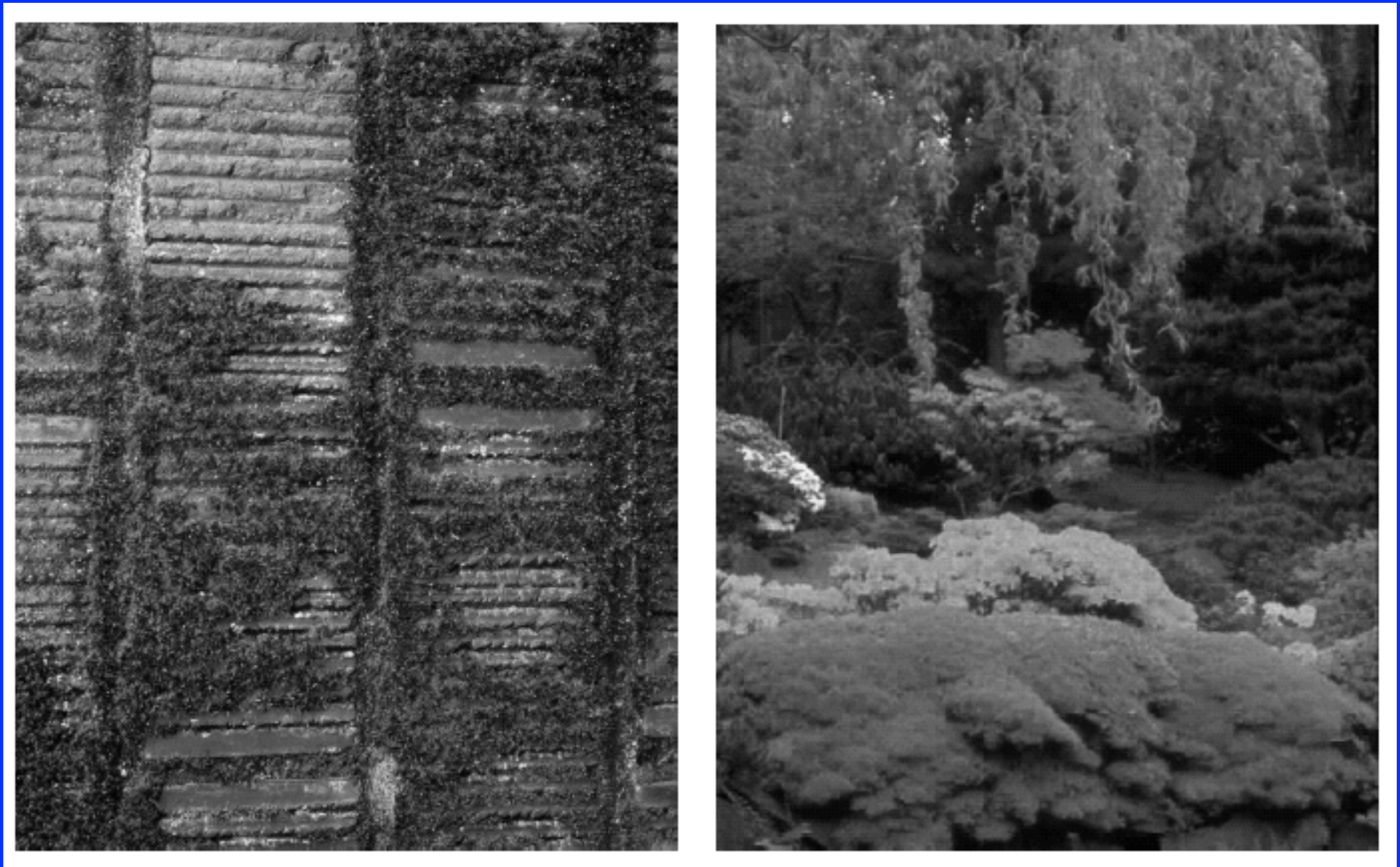
More Complex Discrimination

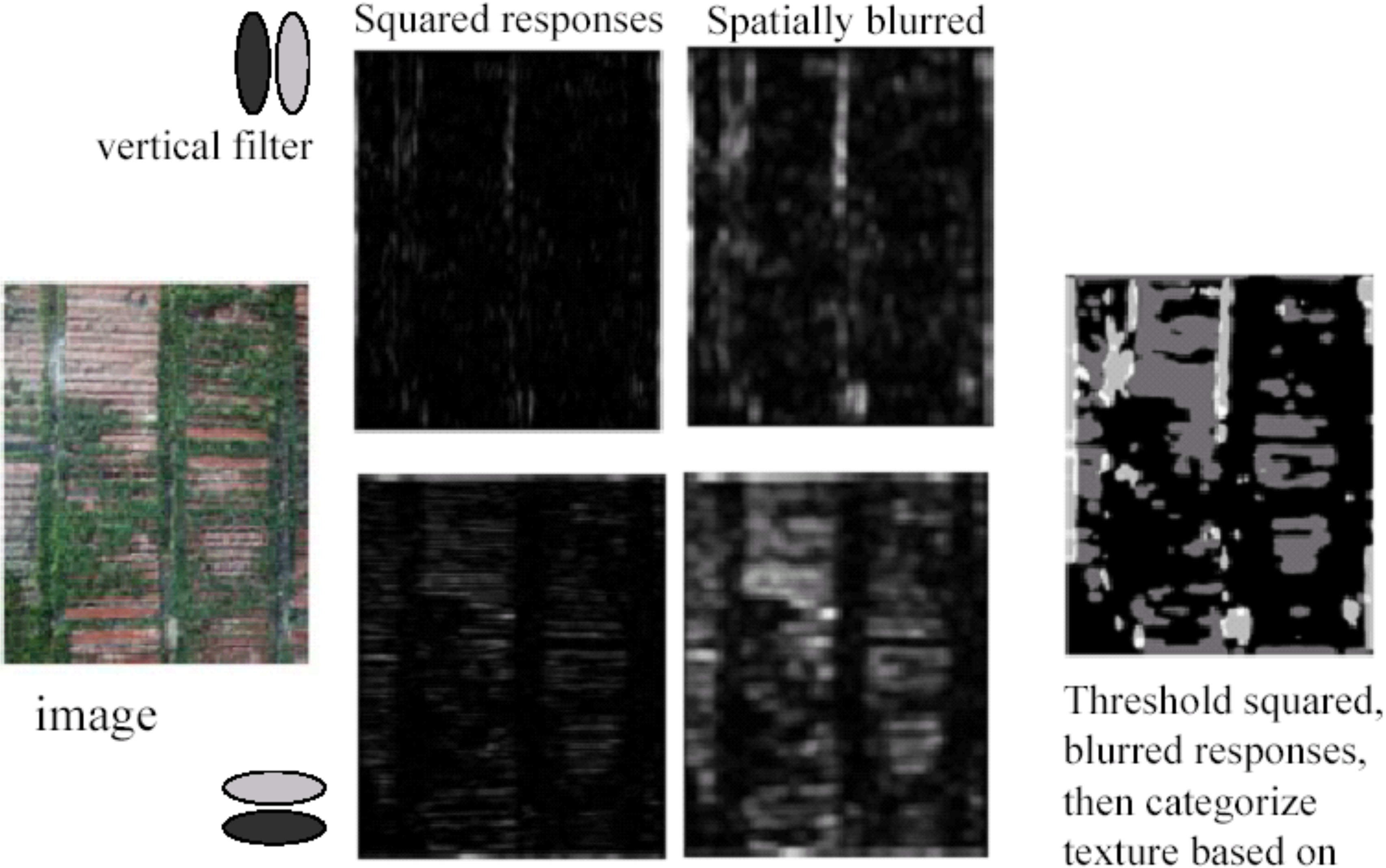
- Histogram comparison is very limiting
  - Every pixel is independent.
  - Everything happens at a tiny scale.

Wavelet representations

- Wavelet coefficients are less dependent than pixels
  - Neighboring pixels are very dependent.
  - This is why used for compression (JPEG2000).
- Less local, seem to capture more info.

Example (Forsyth & Ponce)





vertical filter

Squared responses

Spatially blurred

image

horizontal filter

Threshold squared, blurred responses, then categorize texture based on those two bits

What are Right Filters?

- Multi-scale is good, since we don’t know right scale a priori.
- Easiest to compare assuming independence:

Filter image one: $(F1, F2, \ldots)$

Filter image two: $(G1, G2, \ldots)$

S means image one and two have same texture.

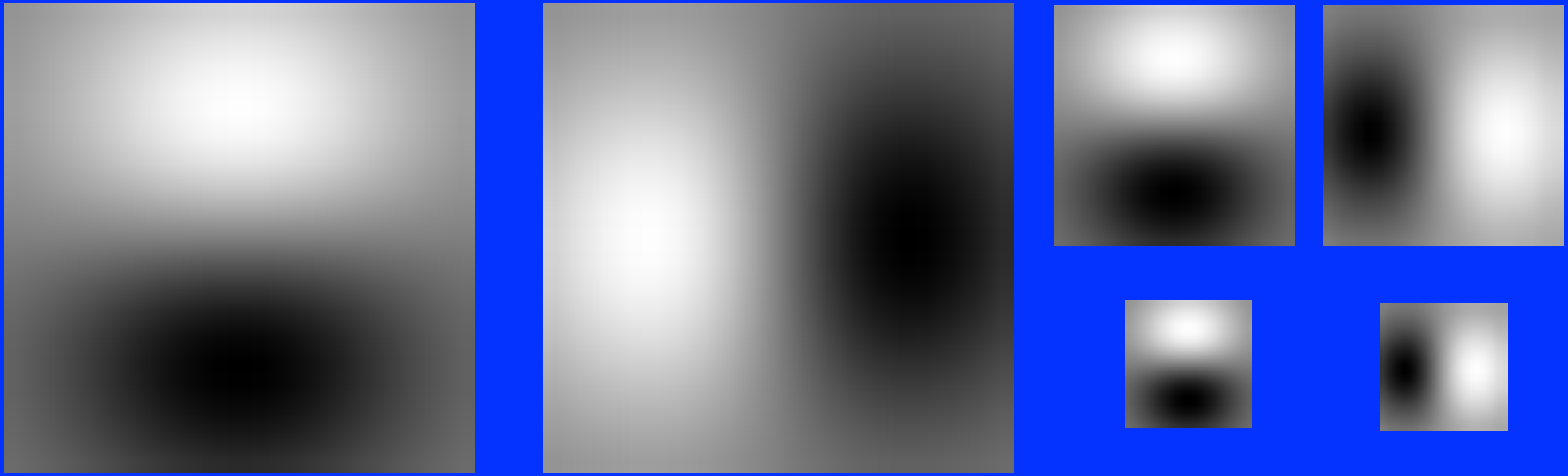
Approximate: $P(F1,G1,F2,G2, \ldots| S)$

By $\mathrm{P}(F1,G1|S)*\mathrm{P}(F2,G2|S)*\ldots$

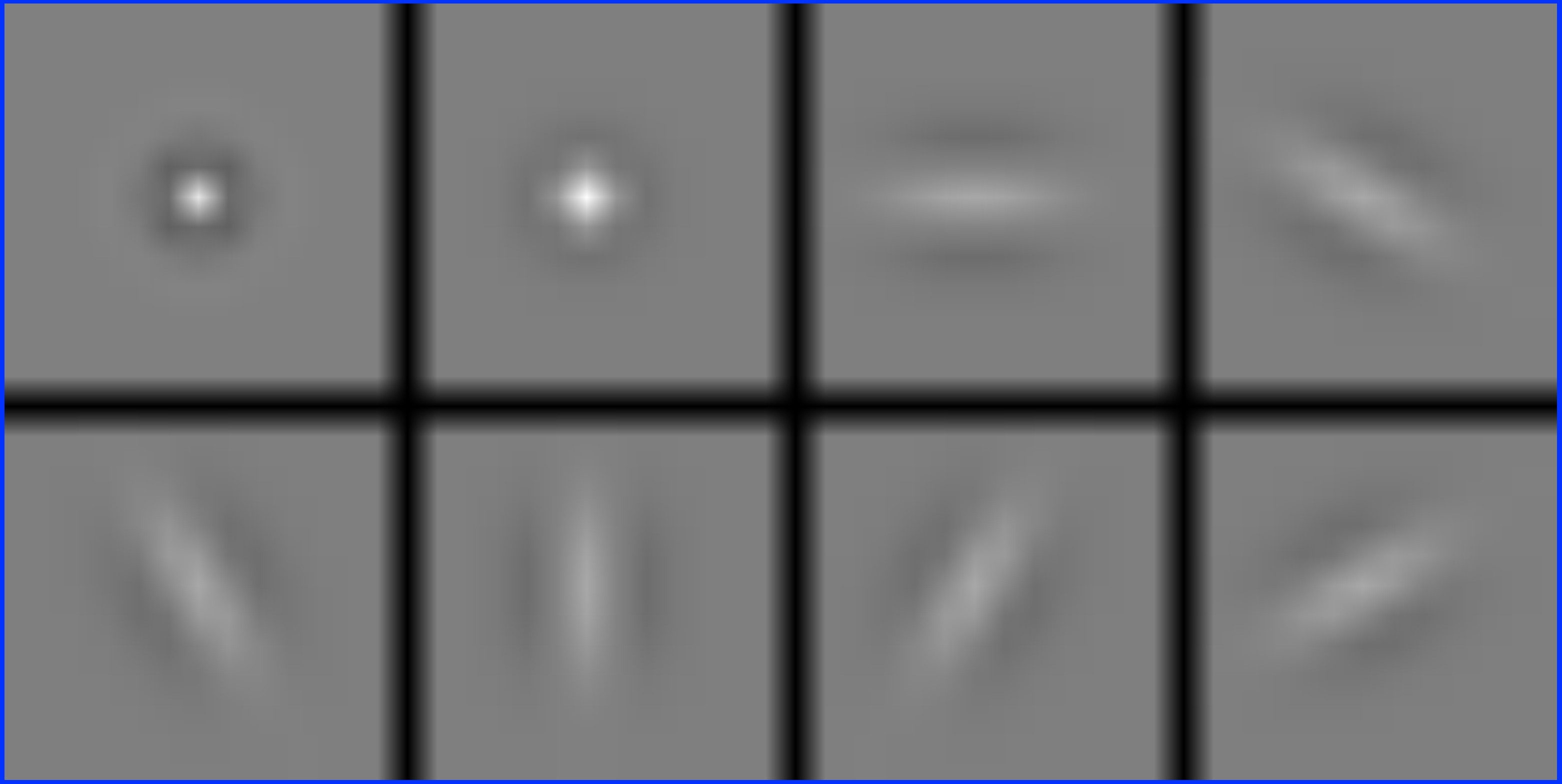
What are Right Filters?

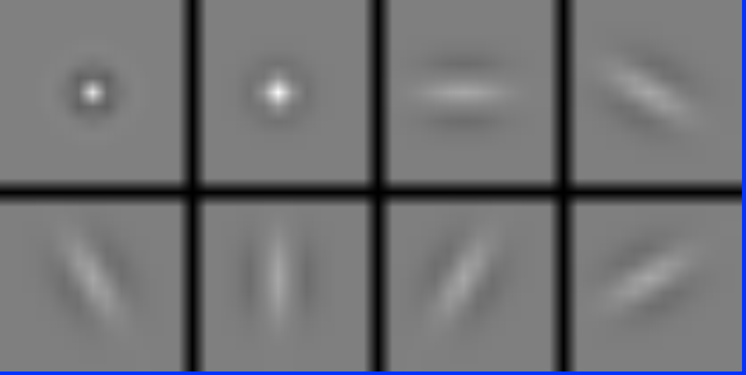
- The more independent the better.
  - In an image, output of one filter should be independent of others.
  - Because our comparison assumes independence.
  - Wavelets seem to be best.

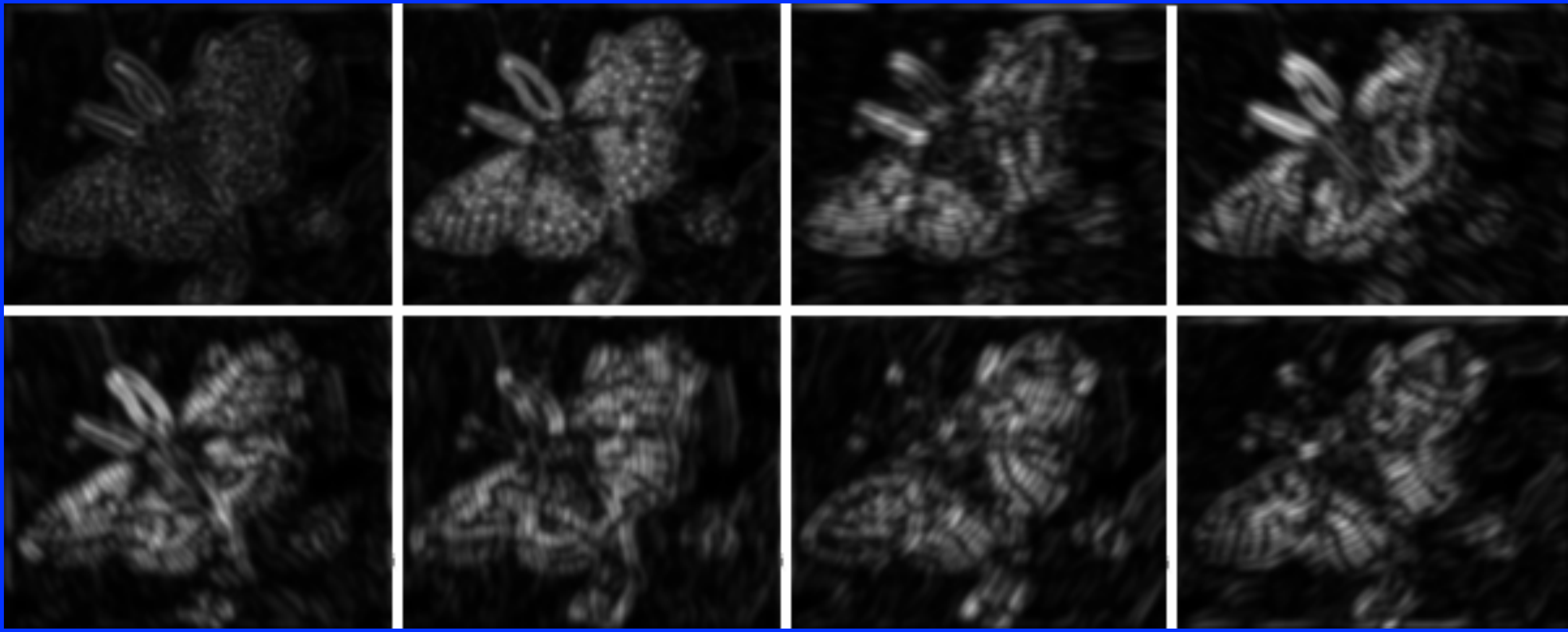
Difference of Gaussian Filters

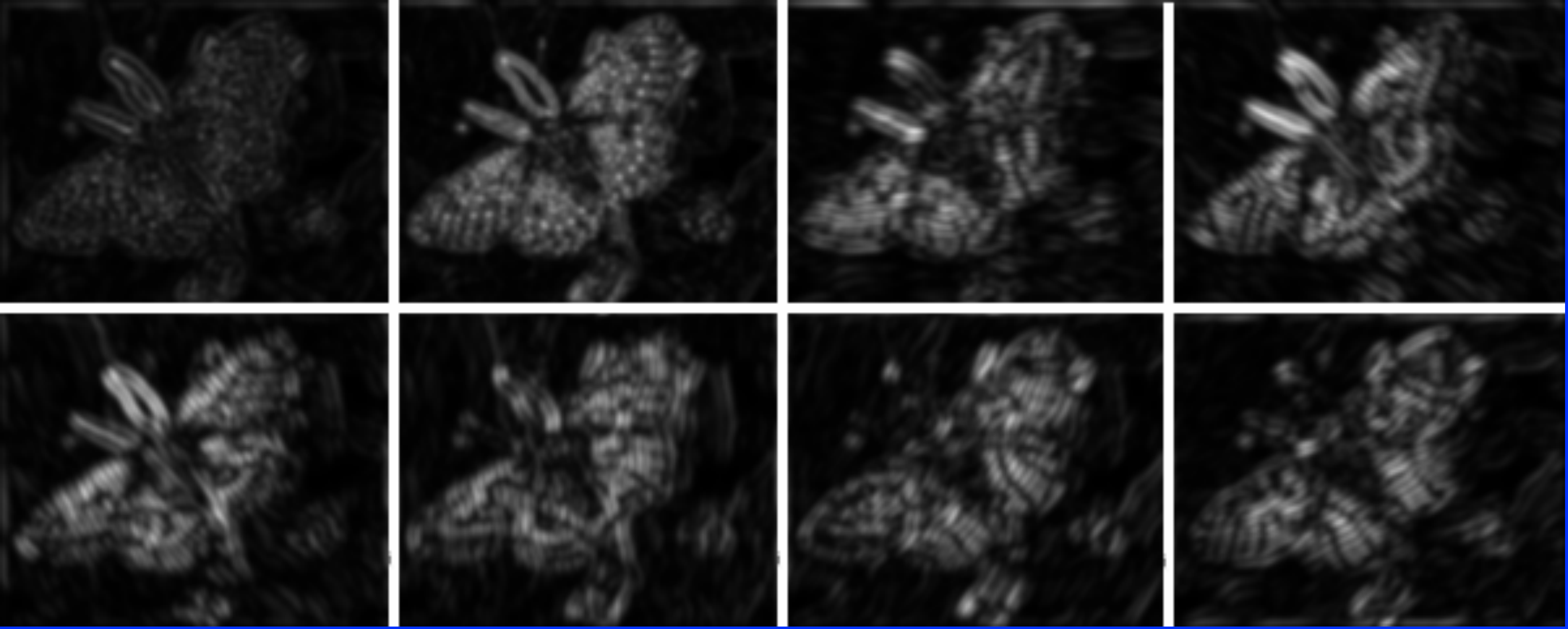


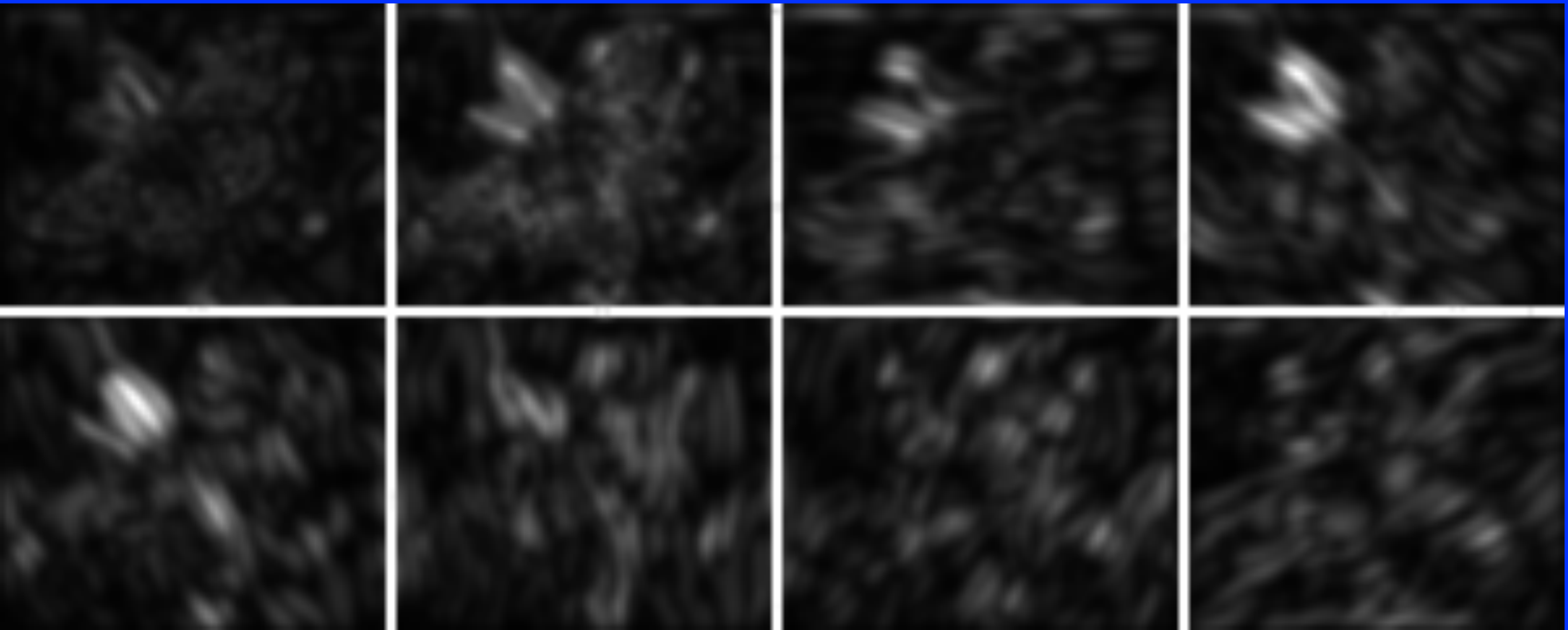
Spots and Oriented Bars
(Malik and Perona)



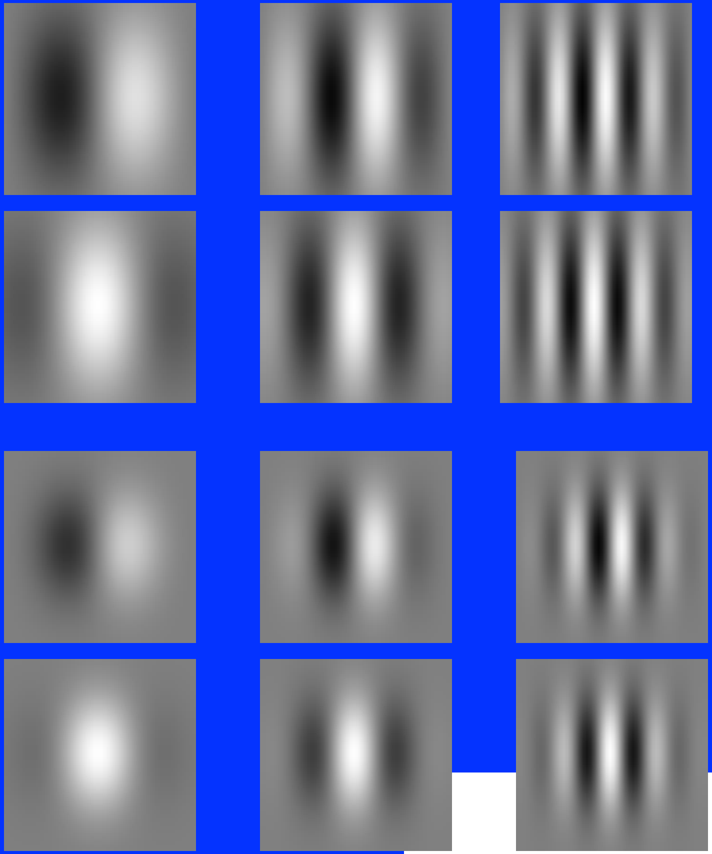








Gabor Filters



Gabor filters at different scales and spatial frequencies

top row shows anti-symmetric (or odd) filters, bottom row the symmetric (or even) filters.

$$\cos(k_x x + k_y y)\exp - \left\{\frac{x^2 + y^2}{2\sigma^2}\right\}$$

Gabor filters are examples of Wavelets

- We know two bases for images:
  - Pixels are localized in space.
  - Fourier are localized in frequency.
- Wavelets are a little of both.
- Good for measuring frequency locally.

Synthesis with this Representation (Bergen and Heeger)

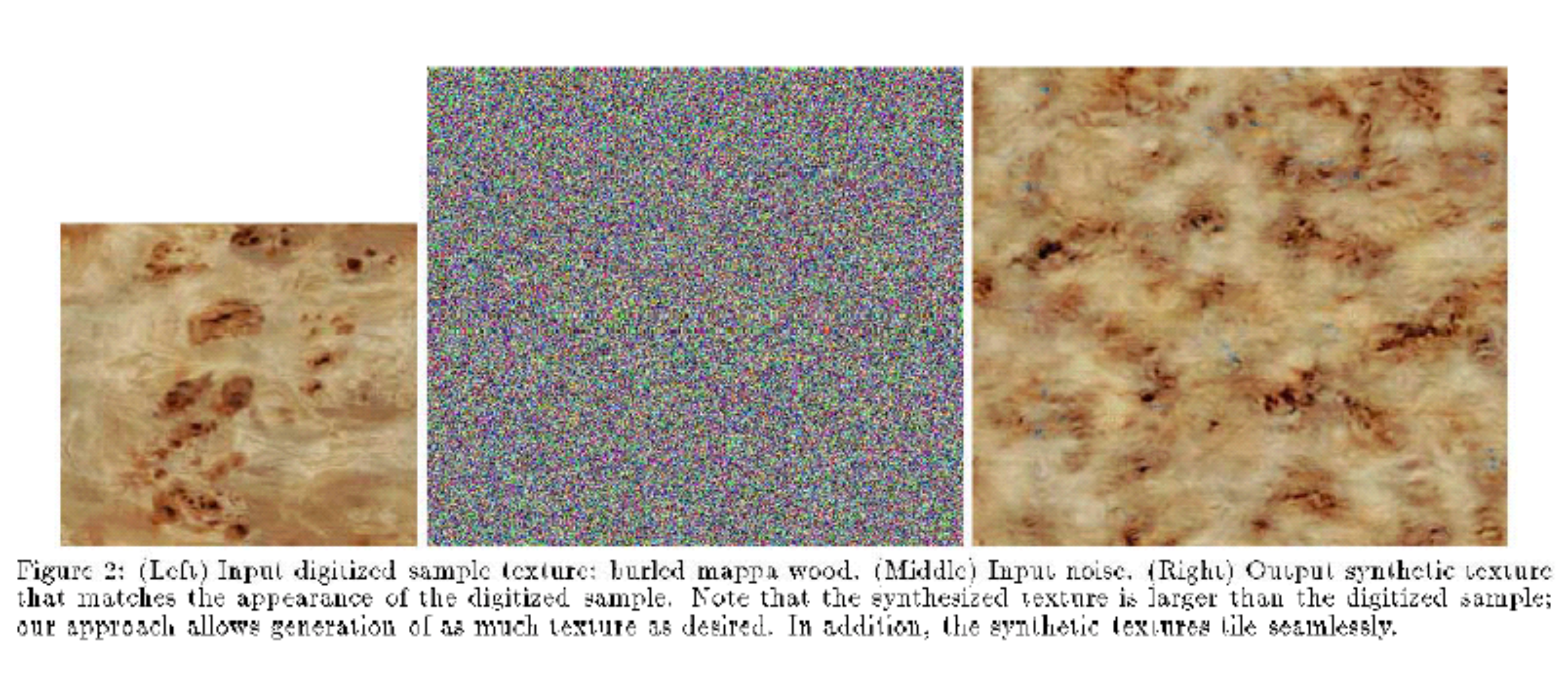


Figure 2: (Left) Input digitized sample texture: burled mappa wood. (Middle) Input noise. (Right) Output synthetic texture that matches the appearance of the digitized sample. Note that the synthesized texture is larger than the digitized sample; our approach allows generation of as much texture as desired.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

Bergen and Heeger results

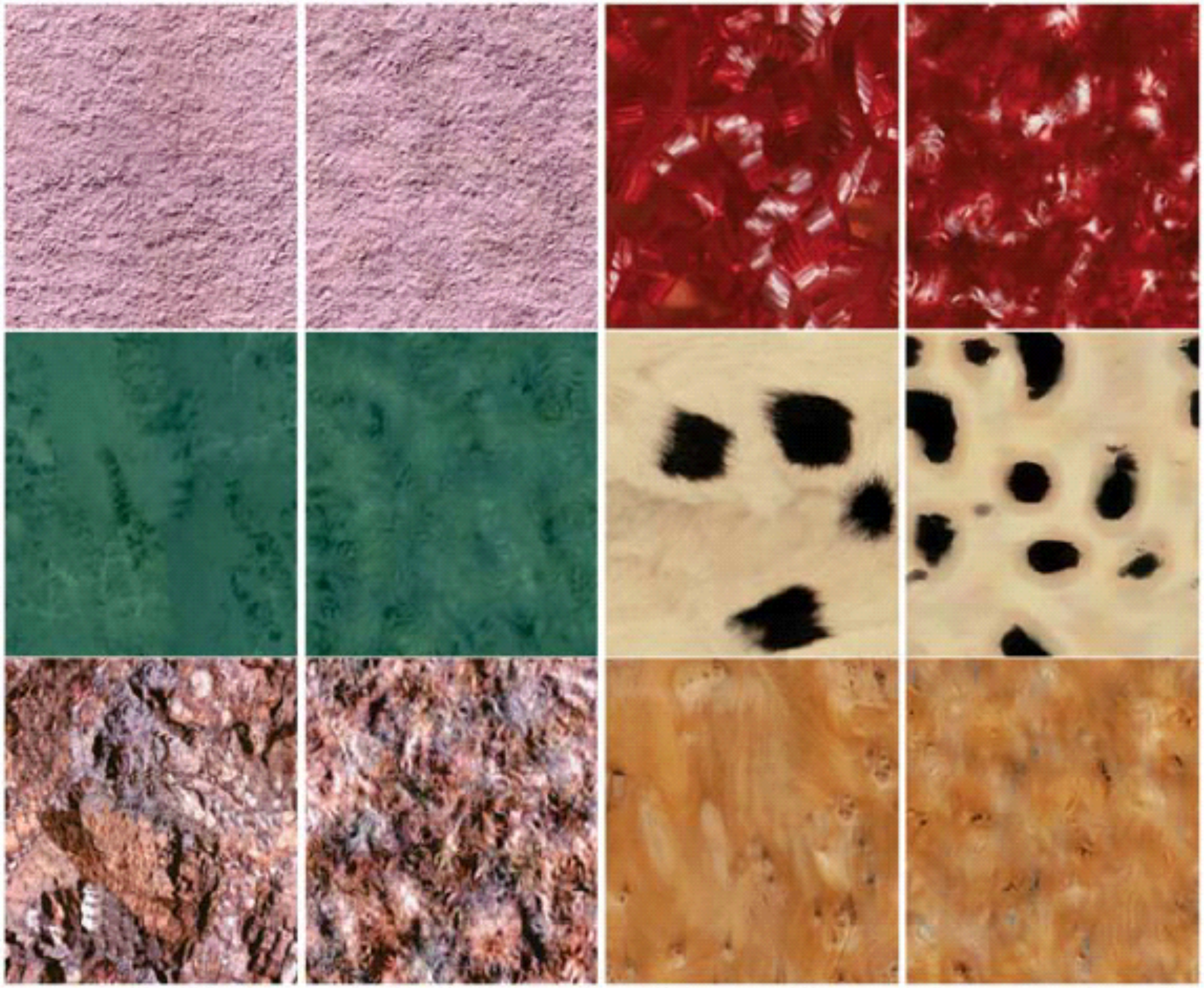


Figure 3: In each pair left image is original and right image is synthetic: stucco, iridescent ribbon, green marble, panda fur, slag stone, figured yew wood.

Bergen and Heeger failures

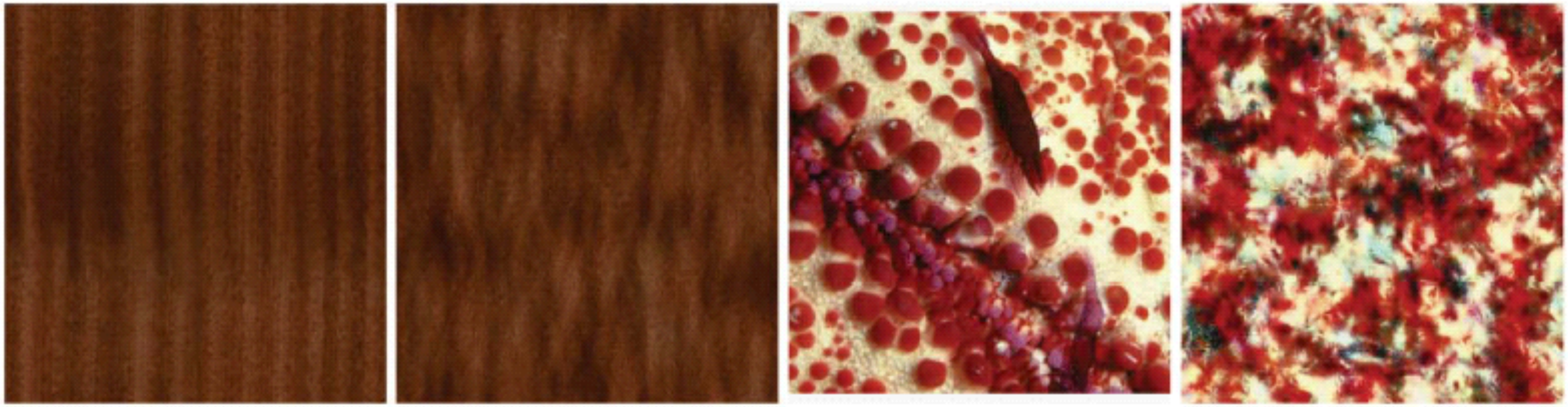


Figure 8: Examples of failures: wood grain and red coral.

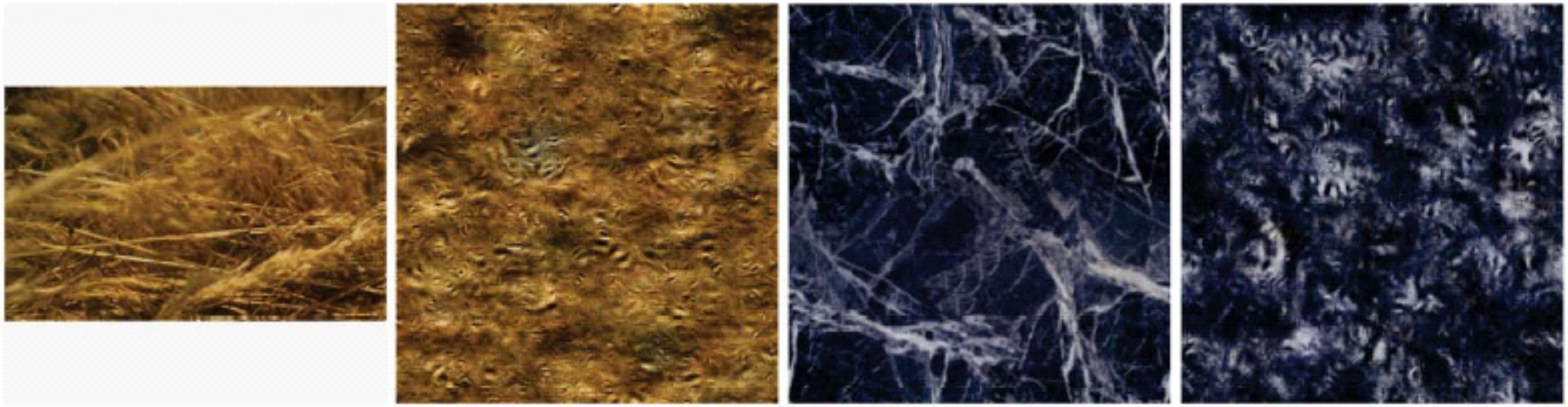


Figure 9: More failures: hay and marble.

Modeling Dependencies

- Pairwise dependencies
  - Co-occurrence of intensities at different distance/angles.
  - Covariance matrix of pixel and all nearby pixels.

Gabor vectors (aka Gabor Jet)

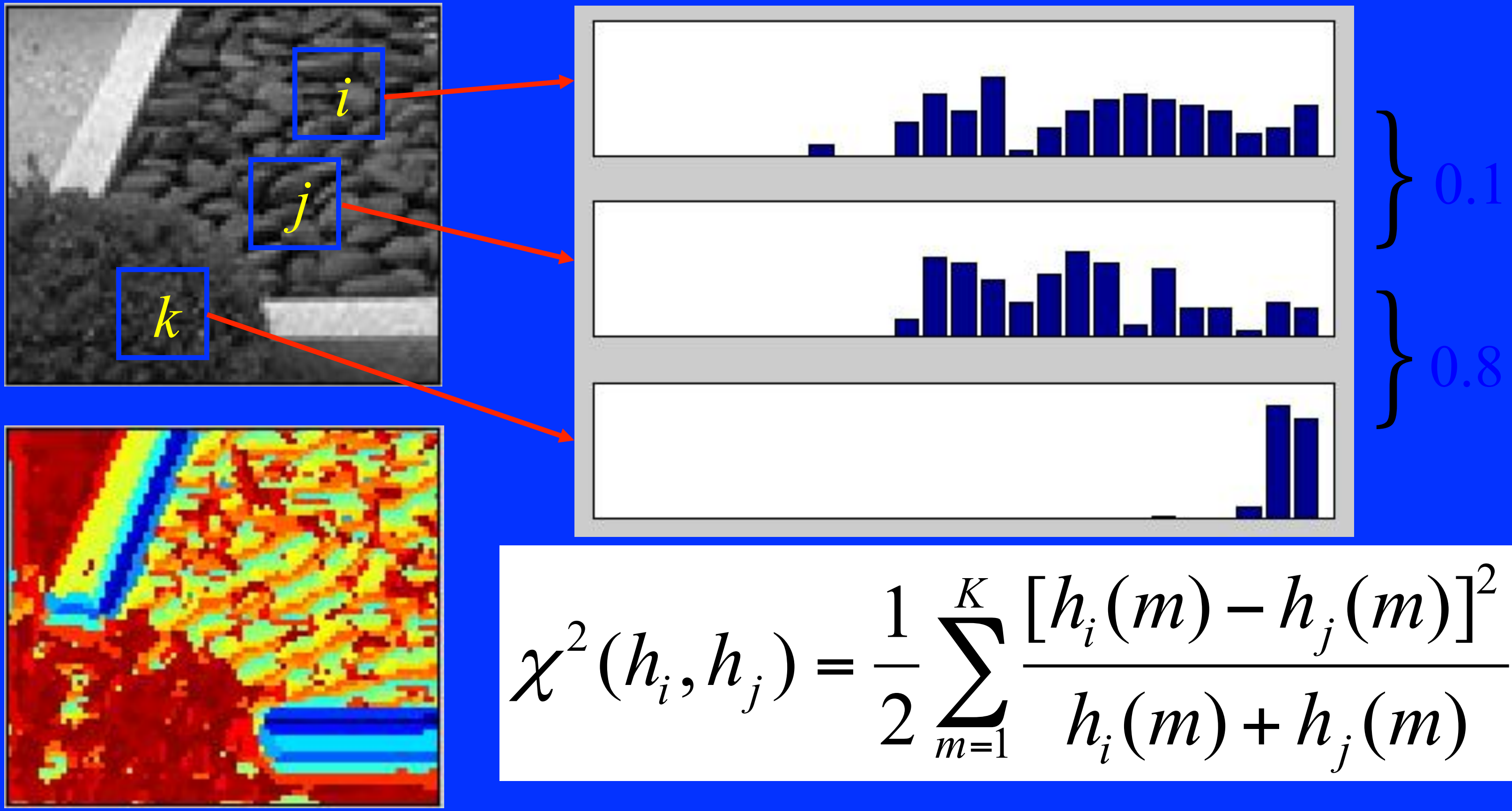
- Compute Gabors at (8) different orientations and (5) scales.
- Each image point -> a point in an 80 dimensional space (each Gabor output is complex).
- Normalize vector length to make Gabor Jet
- Compare histograms in 80D
  - This is hard part.
  - Dividing space into regular buckets doesn’t work.
  - Use Vector Quantization

Vector Quantization for Histogram Comparison

- Given two images, compute Gabor Jet at each pixel.
- Cluster all pixels (using k-means) to get ~100 clusters.
- For each image, count #pixels in each cluster, producing histogram with ~100 buckets.
- Compare histograms (eg., using SSD, Chi-square distance, …)

Chi square distance between texton histograms

Chi-square



$$\chi^2(h_i,h_j) = \frac{1}{2}\sum_{m=1}^{K}\frac{[h_i(m)-h_j(m)]^2}{h_i(m)+h_j(m)}$$

(Malik)

Markov Model

- Captures local dependencies.
  - Each pixel depends on neighborhood.
- Example, 1D first order model

$P(p1, p2, \ldots pn) = P(p1)*P(p2|p1)*P(p3|p2,p1)*\ldots$

$= P(p1)*P(p2|p1)*P(p3|p2)*P(p4|p3)*\ldots$

Markov model of Printed English

- From Shannon: "A mathematical theory of communication."
- Think of text as a 1D texture
- Choose next letter at random, based on previous letters.

- Zero’ th order:

XFOML RXKHJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZIHJQD

- Zero’ th order:

XFOML RXKHJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZIHJQD

- First order:

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRI

- First order:

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRI

- Second order

ON IE ANTSOUTINYS ARE T INCTORE T BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

- Second order

ON IE ANTSOUTINYS ARE T INCTORE T BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third order:

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

- Zero’th order: XFOML RXKHJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZIHJQD
- First order: OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRI
- Second order ON IE ANTSOUTINYS ARE T INCTORE T BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE
- Third order: IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

Markov models of words

- First order:

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

- Second order:

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

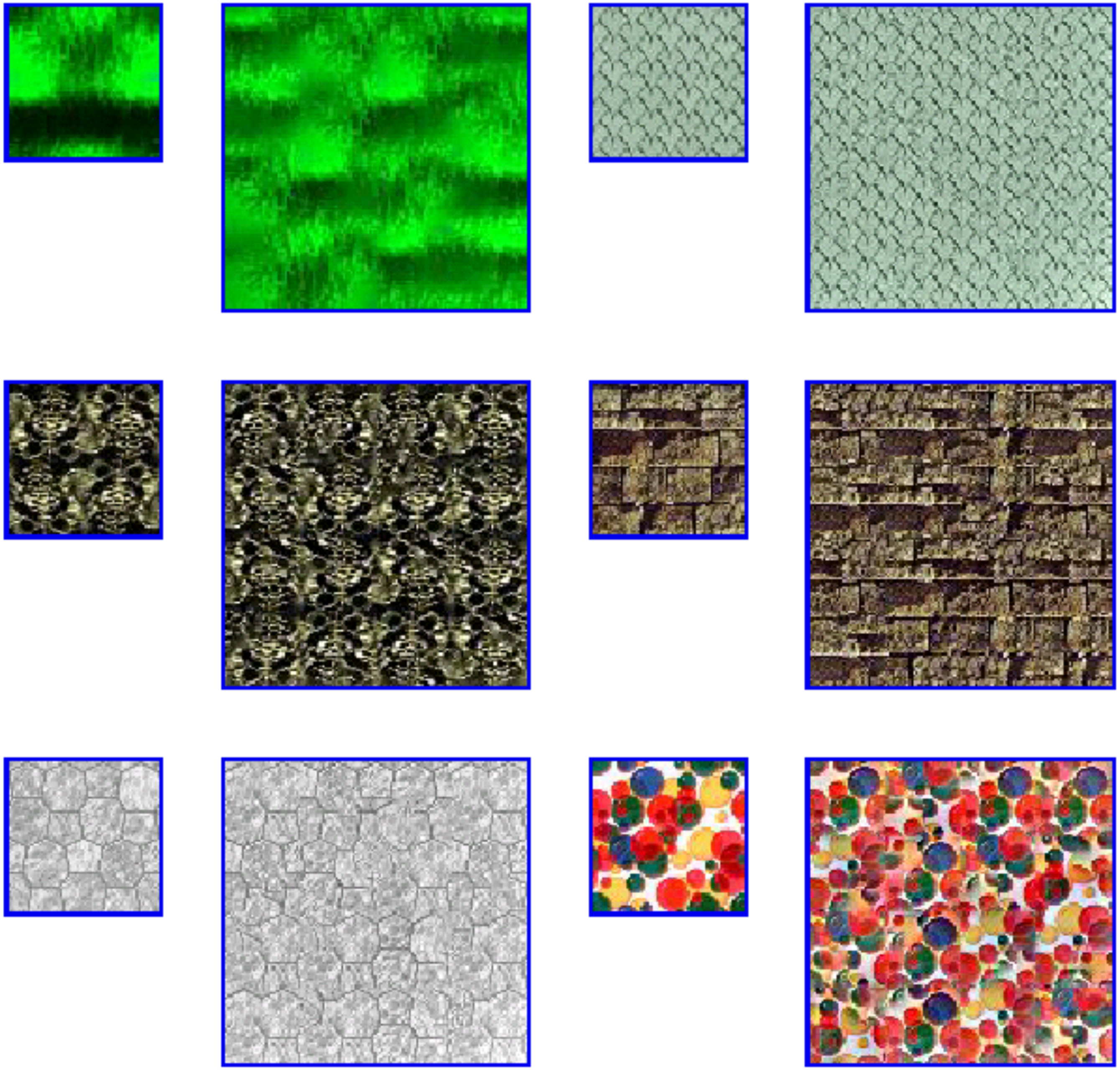
Example 1st Order Markov Model

- Each pixel is like neighbor to left + noise with some probability.
- These capture a much wider range of phenomena.

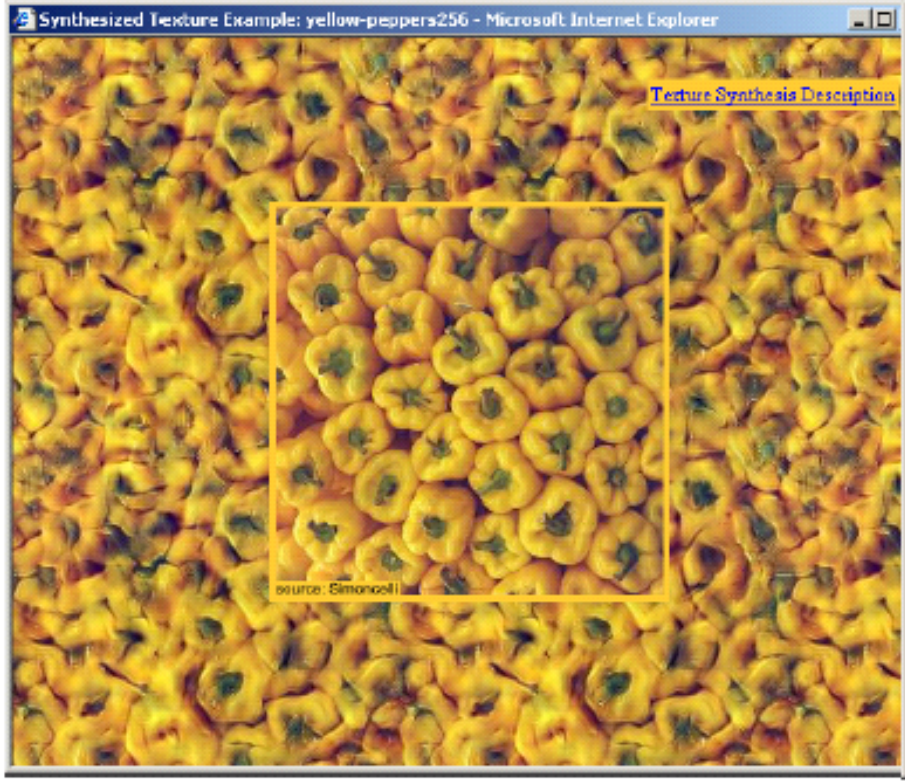
There are dependencies in Filter Outputs

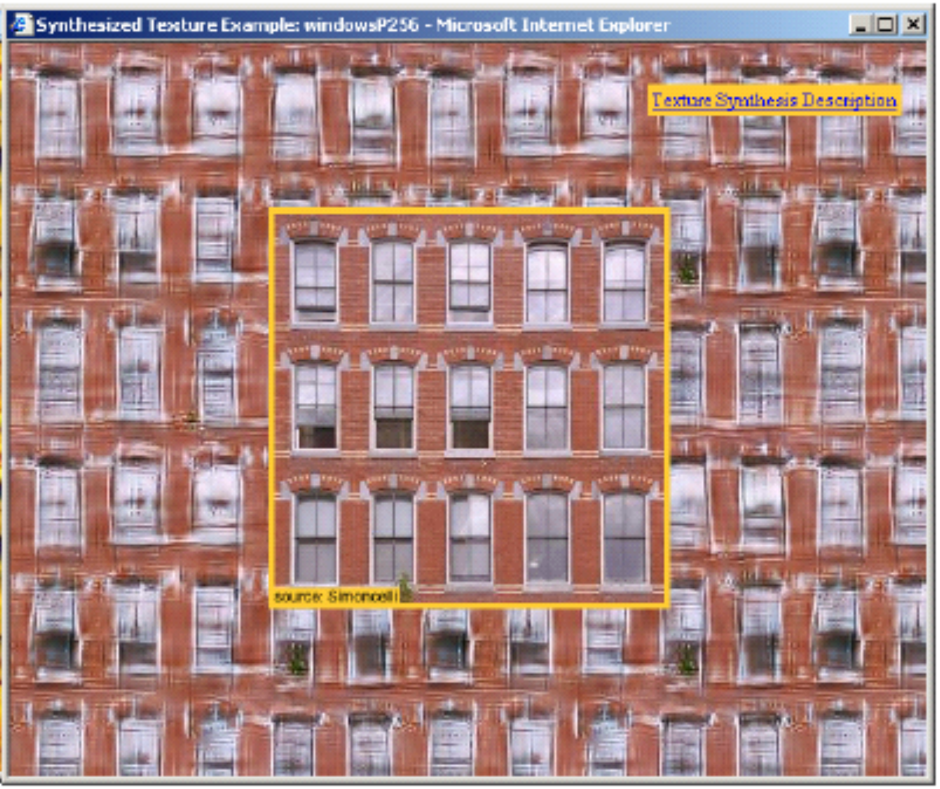
- Edge
  - Filter responds at one scale, often does at other scales.
  - Filter responds at one orientation, often doesn’t at orthogonal orientation.
- Synthesis using wavelets and Markov model for dependencies:
  - DeBonet and Viola
  - Portilla and Simoncelli

DeBonet



Portilla and Simoncelli


Synthesized Texture Example: yellow-peppers256 - Microsoft Internet Explorer
Texture Synthesis Description
source: Simoncelli

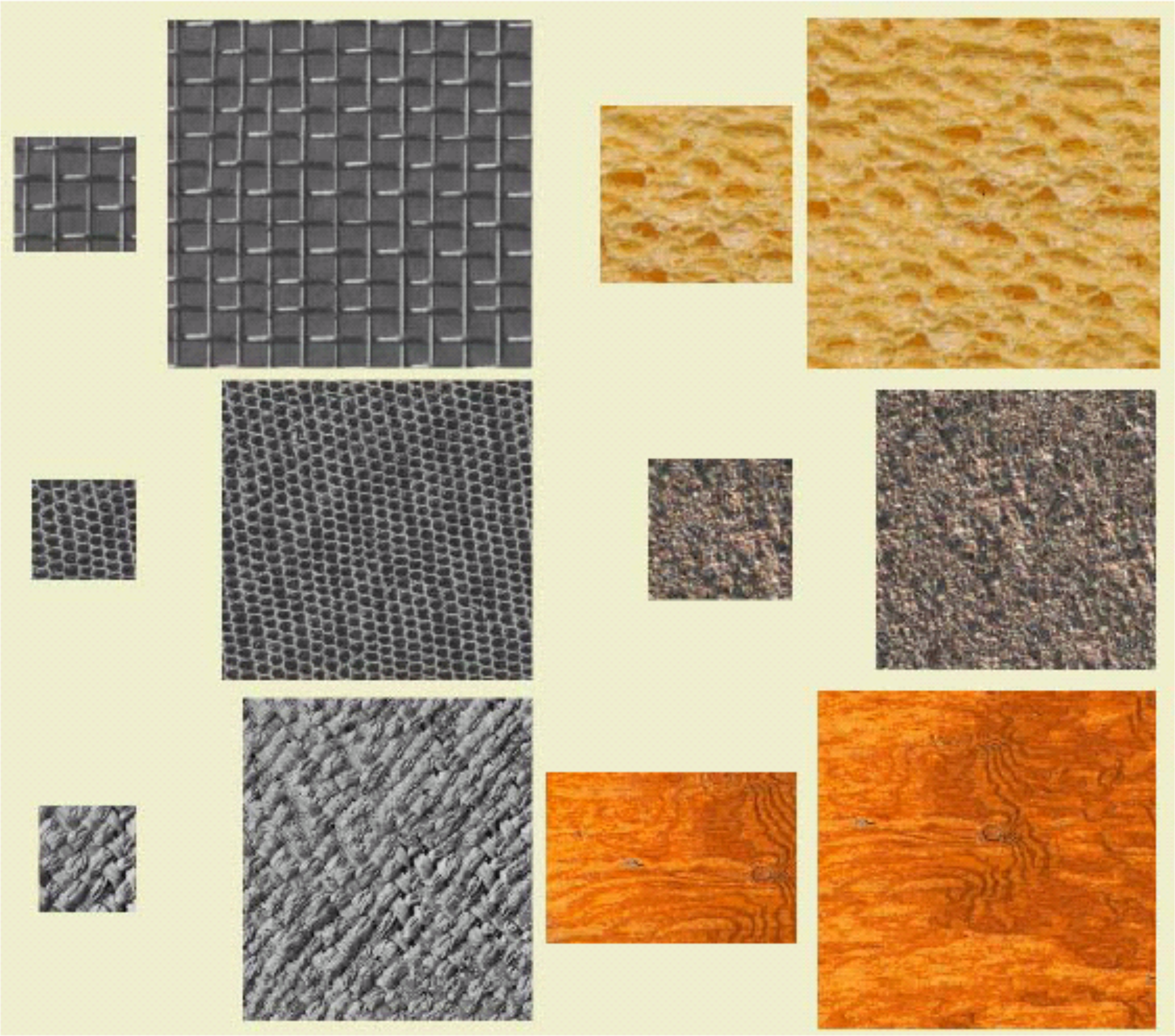

Synthesized Texture Example: windowsP256 - Microsoft Internet Explorer
Texture Synthesis Description
source: Simoncelli

We can do this without filters

- Each pixel depends on neighbors.

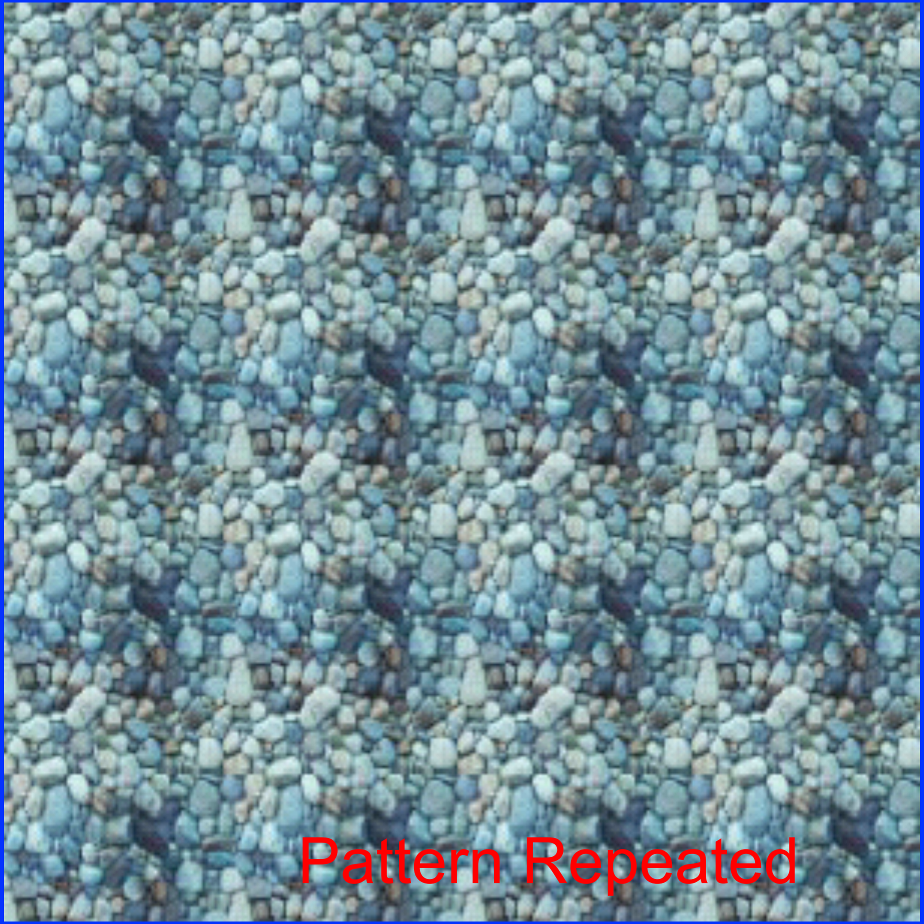
1. As you synthesize, look at neighbors.
2. Look for similar neighborhood in sample texture.
3. Copy pixel from that neighborhood.
4. Continue.

Efros and Leung



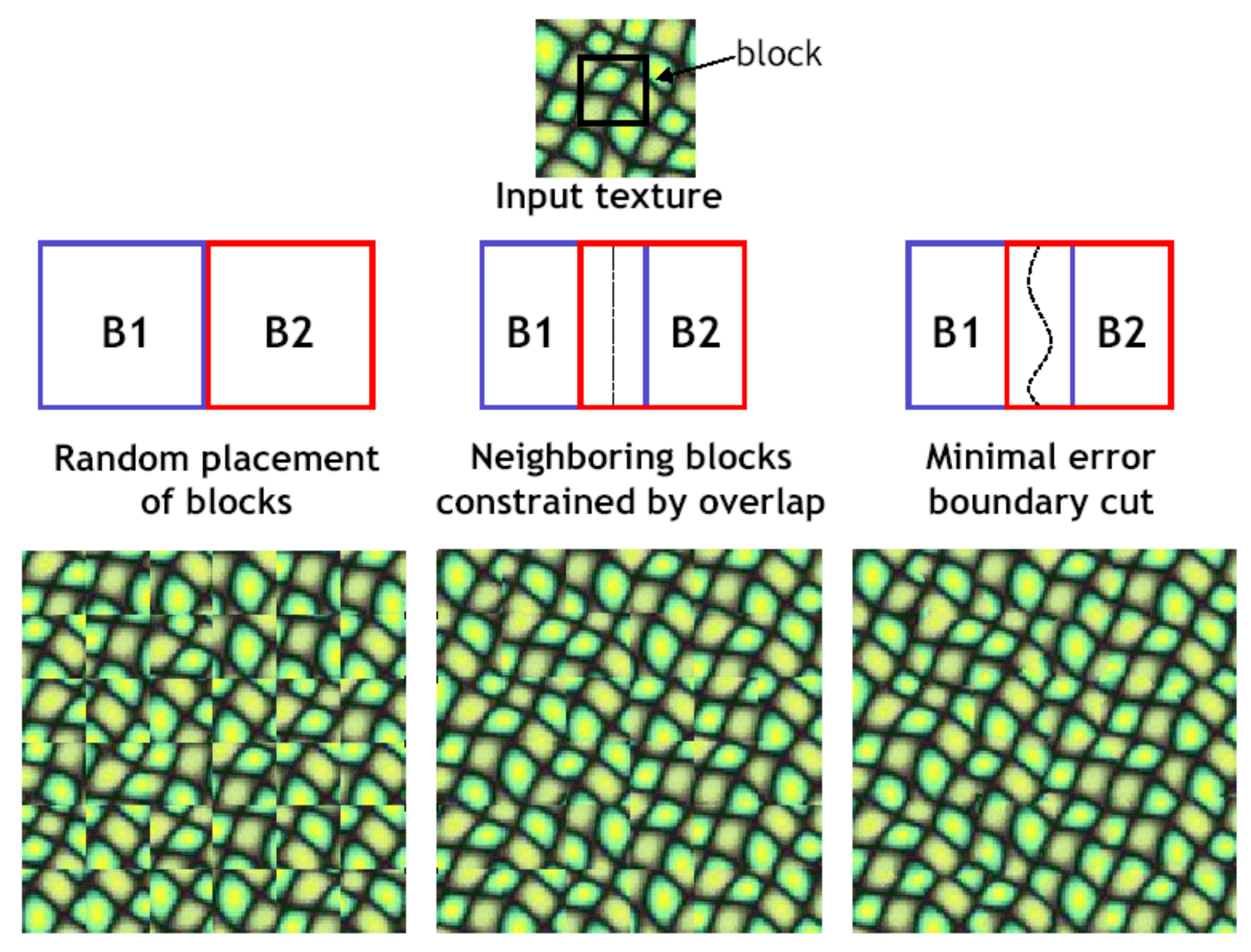
This is like copying, but not just repetition

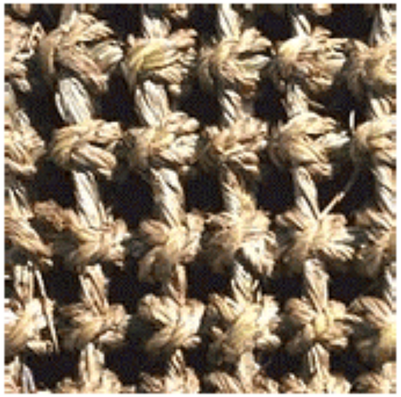
Photo

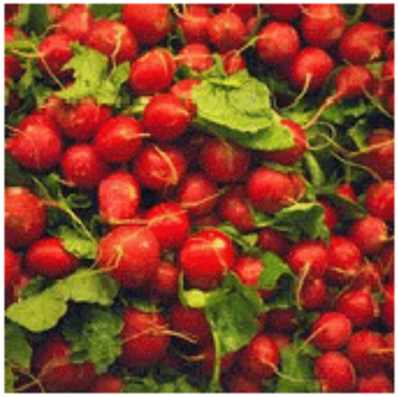


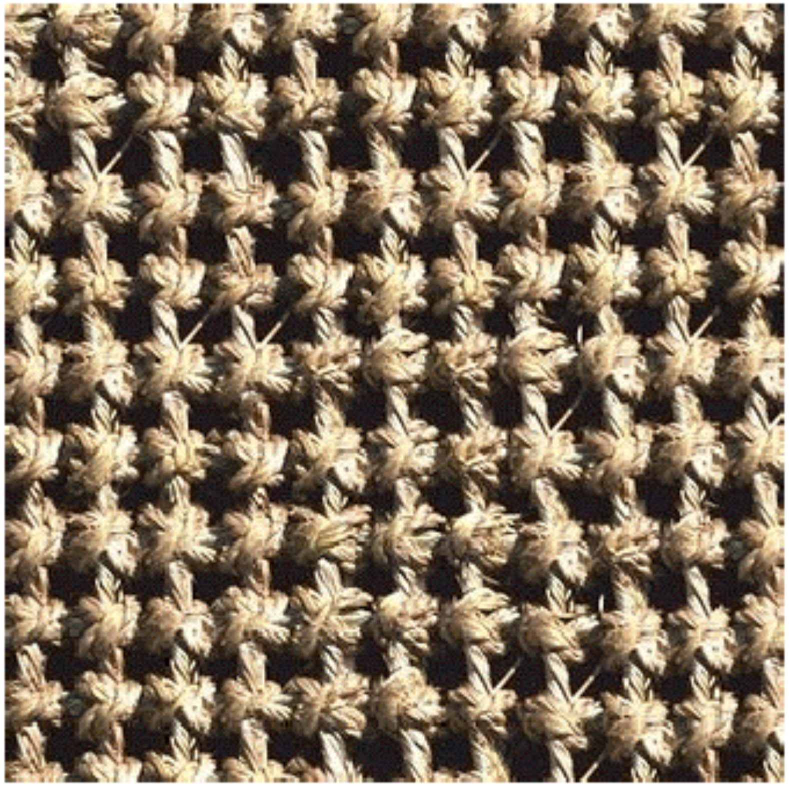
Pattern Repeated

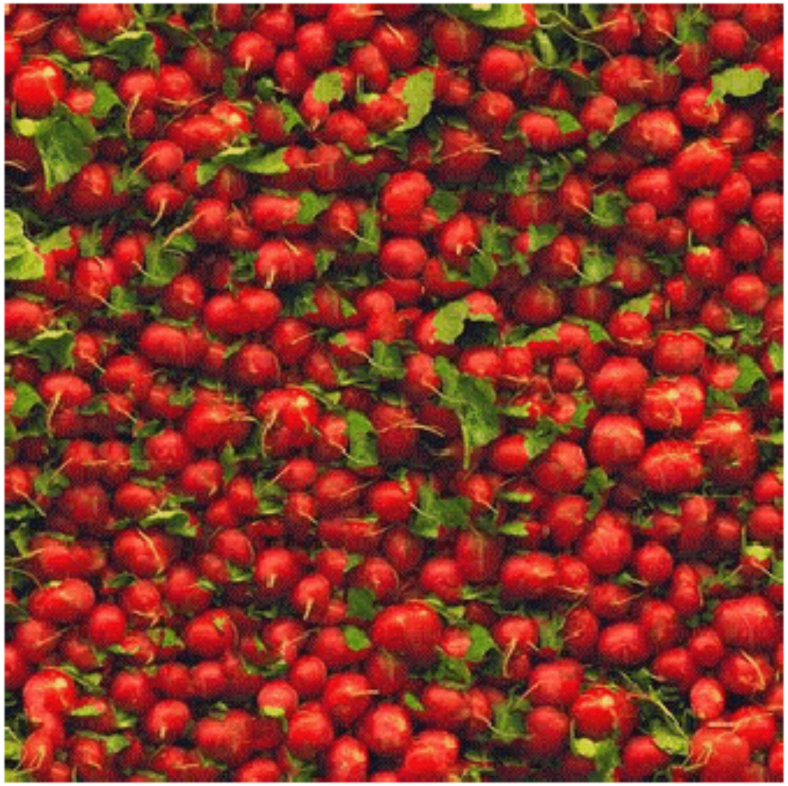
With Blocks


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

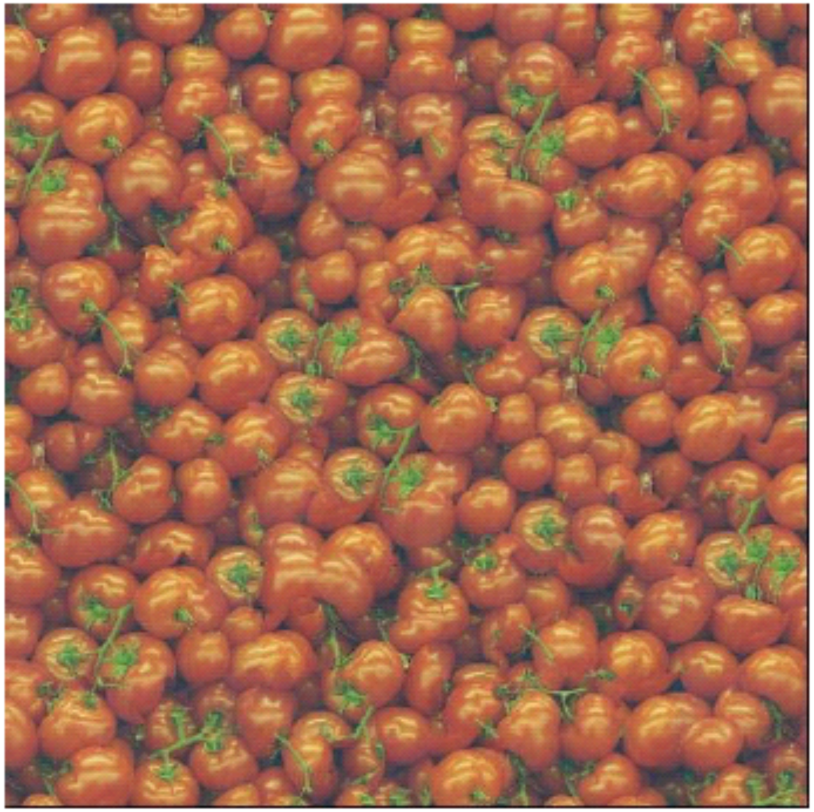


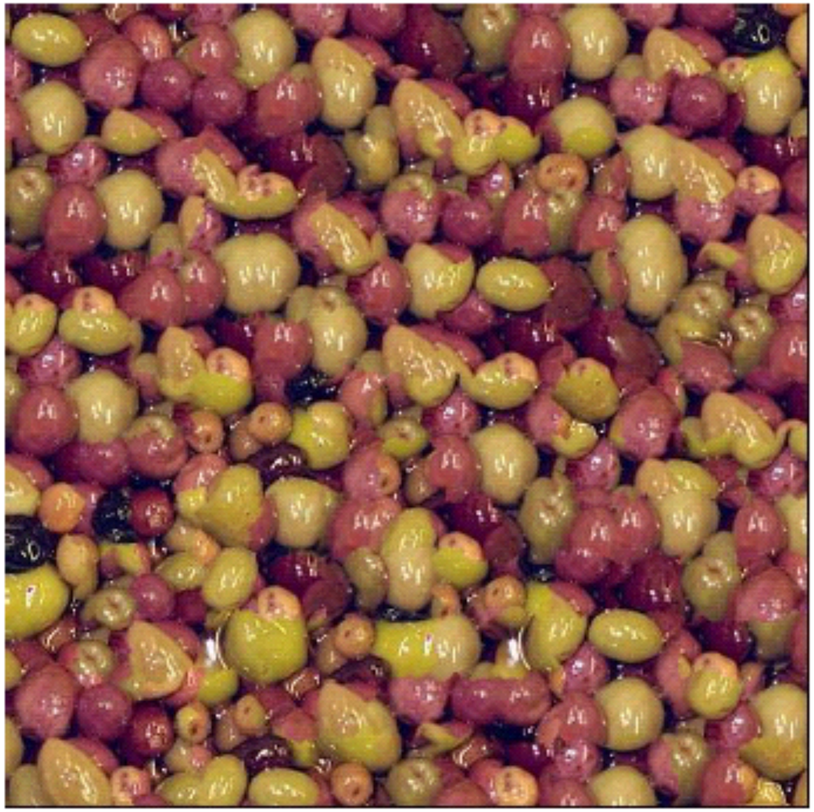






Failures
(Chernobyl
Harvest)





Conclusions

- Model texture as generated from random process.
- Discriminate by seeing whether statistics of two processes seem the same.
- Synthesize by generating image with same statistics.

To Think About

- 3D effects
  - Shape: Tiger’ s appearance depends on its shape.
  - Lighting: Bark looks different with light angle
- Given pictures of many chairs, can we generate a new chair?

Textons