Course Calendar

Lecture	Date	Description	Readings	Assignments	Materials
1	2/1	Course Introduction Cameras and Lenses	Req: FP 1.1, 2.1, 2.2, 2.3, 3.1, 3.2	PS0 out	
2	2/3	Image Filtering	Req: FP 7.1 - 7.6		
3	2/8	Image Representations: Pyramids	Req: FP 7.7, 9.2		
4	2/10	Image Statistics		PS0 due	
5	2/15	Texture	Req: FP 9.1, 9.3, 9.4	PS1 out	
6	2/17	Color	Req: FP 6.1-6.4		
7	2/22	Guest Lecture: Context in vision			
8	2/24	Guest Lecture: Medical Imaging		PS1 due	
9	3/1	Multiview Geometry	Req: Mikolajczyk and Schmid; FP 10	PS2 out	
10	3/3	Local Features	Req: Shi and Tomasi; Lowe		

Course Calendar

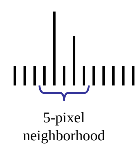
	Lecture	Date	Description	Readings	Assignments	Materials
Today	5	2/15	Texture	Req: FP 9.1, 9.3, 9.4	PS1 out	

Non-linear filtering example

Median filter

Replace each pixel by the median over N pixels (5 pixels, for these examples). Generalizes to “rank order” filters.

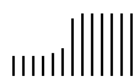
In:



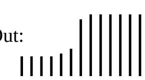
Out:

Spike noise is removed

In:



Out:



Monotonic edges remain unchanged

Degraded image

Radius 1 median filter

Because the filter is non-linear, it has the opportunity to remove the scratch noise without blurring edges.

Radius 2 median filter

Comparison with linear blur of the amount needed to remove the scratches

CCD color sampling

Color sensing, 3 approaches

- Scan 3 times (temporal multiplexing)
- Use 3 detectors (3-ccd camera, and color film)
- Use offset color samples (spatial multiplexing)

Typical errors in temporal multiplexing approach

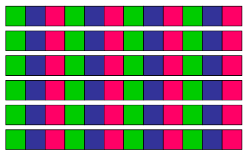
Color offset fringes

Typical errors in spatial multiplexing approach.

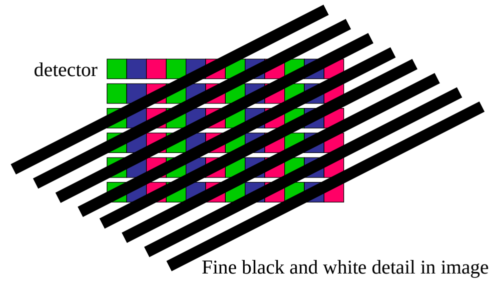
Color fringes.

CCD color filter pattern

detector



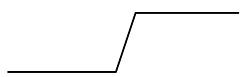
The cause of color moire



Fine black and white detail in image mis-interpreted as color information.

Black and white edge falling on color CCD detector

Black and white image (edge)

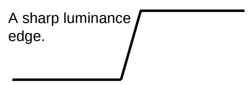


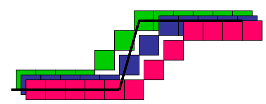
Detector pixel colors

Color sampling artifacts

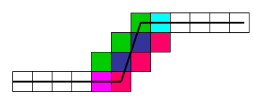
Interpolated pixel colors, for grey edge falling on colored detectors (linear interpolation). The edge is aliased (undersampled) in the samples of any one color. That aliasing manifests itself in the spatial domain as an incorrect estimate of the precise position of the edge. That disagreement about the position of the edge results in a color fringe artifact.

A sharp luminance edge.





The response of independently interpolated color bands to an edge.



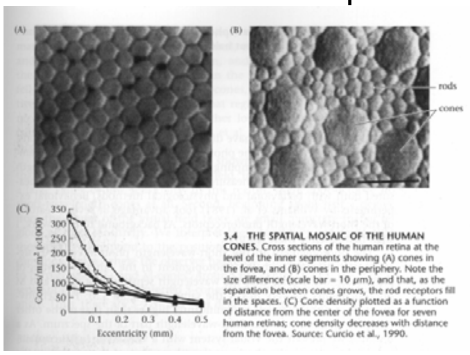
The mis-estimated edge yields color fringe artifacts.

Typical color moire patterns

Blow-up of electronic camera image. Notice spurious colors in the regions of fine detail in the plants.

Color sampling artifacts

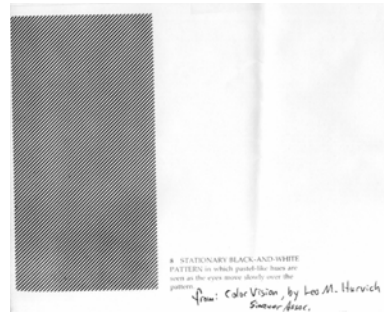
Human Photoreceptors



(From Foundations of Vision, by Brian Wandell, Sinauer Assoc.)

Brewster's colors example (subtle).

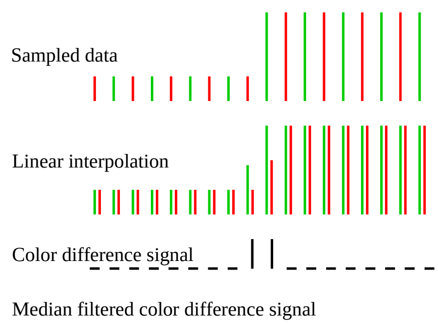
Scale relative to human photoreceptor size: each line covers about 7 photoreceptors.



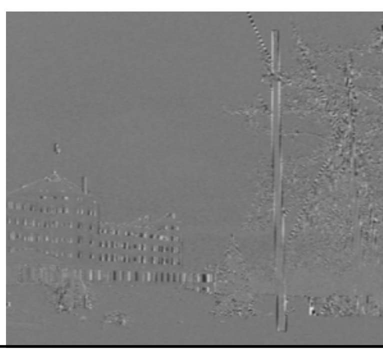
Median Filter Interpolation

1) Perform first interpolation on isolated color channels.
2) Compute color difference signals.
3) Median filter the color difference signal.
4) Reconstruct the 3-color image.

Two-color sampling of BW edge



R-G, after linear interpolation



R – G, median filtered (5x5)

Recombining the median filtered colors

Linear interpolation　　Median filter interpolation

References on color interpolation

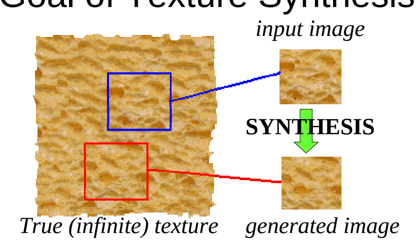
- Brainard
- Shree nayar.

Image texture

Texture

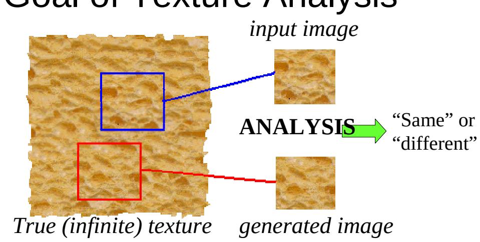
- Key issue: representing texture
  - Texture based matching
    - little is known
  - Texture segmentation
    - key issue: representing texture
  - Texture synthesis
    - useful; also gives some insight into quality of representation
  - Shape from texture
    - cover superficially

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"

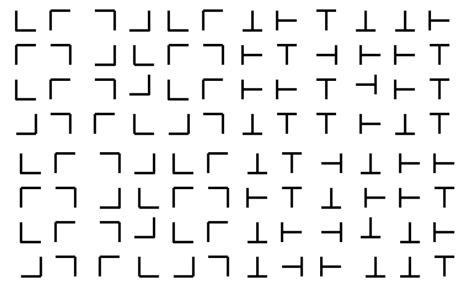
The Goal of Texture Analysis



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

Pre-attentive texture discrimination

Pre-attentive texture discrimination

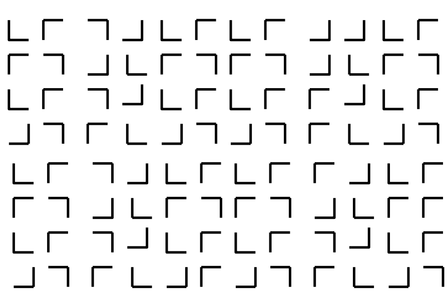


Pre-attentive texture discrimination

Same or different textures?

Pre-attentive texture discrimination

Pre-attentive texture discrimination

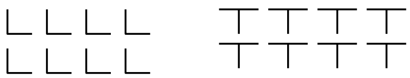


Pre-attentive texture discrimination

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...



Influential paper:

Early vision and texture perception

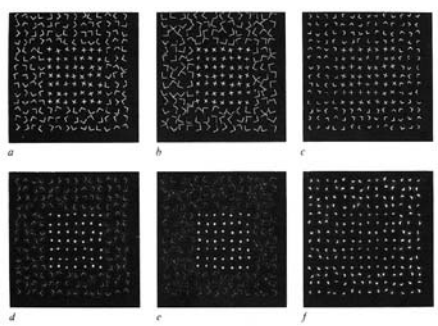
James R. Bergen* & Edward H. Adelson**

* 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.

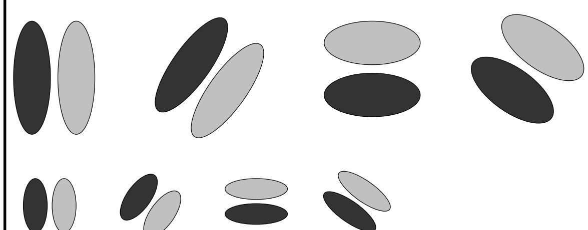
Bergen and Adelson, Nature 1988

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*.



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

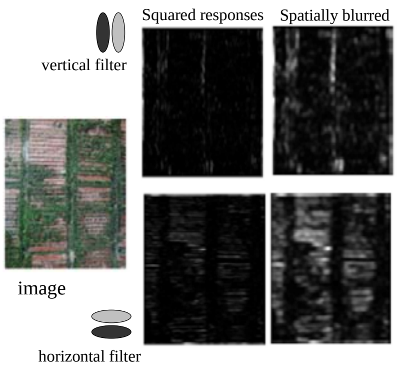
Malik and Perona



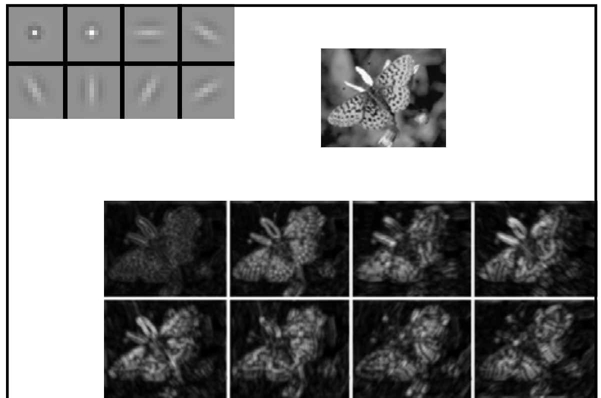
Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms. J OPT SOC AM A 7: (5) 923-932 MAY 1990

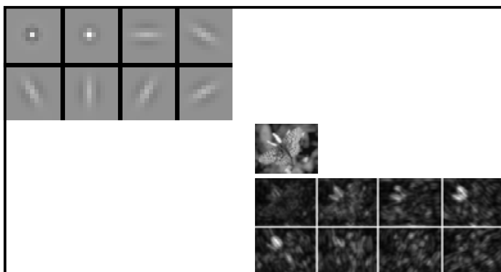
Representing textures

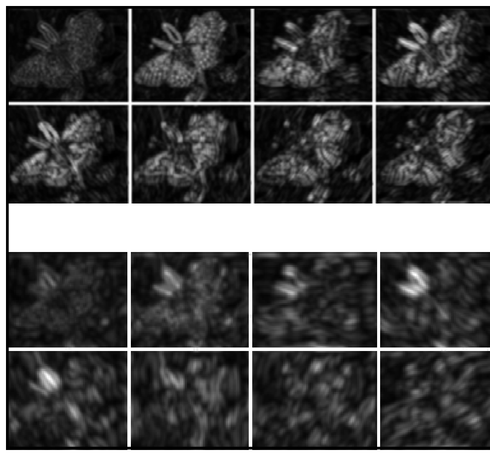
- Textures are made up of quite stylised subelements, repeated in meaningful ways
- Representation:
  - find the subelements, and represent their statistics
- But what are the subelements, and how do we find them?
  - recall normalized correlation
  - find subelements by applying filters, looking at the magnitude of the
- What filters?
  - experience suggests spots and oriented bars at a variety of different scales
  - details probably don't matter
- What statistics?
  - within reason, the more the merrier.
  - At least, mean and standard deviation
  - better, various conditional histograms.

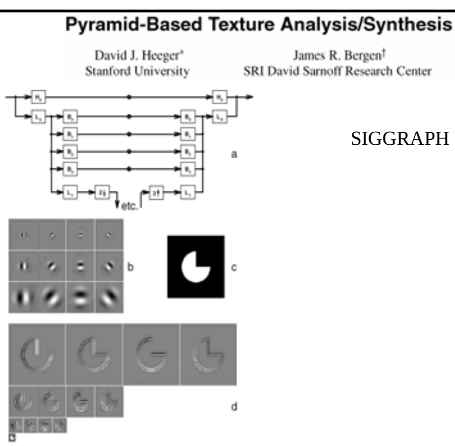


Threshold squared, blurred responses, then categorize texture based on those two bits









SIGGRAPH 1994

Show block diagram of heeger bergen

- And demonstrate it working with matlab code. Ask ted for example.

Learn: use filter marginal statistics.

Bergen and Heeger

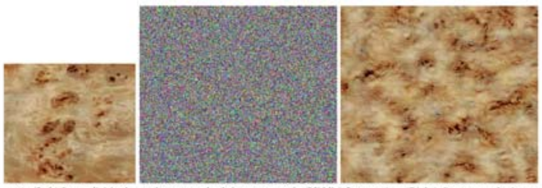


Figure 2: (Left) Input digitized sample texture: burled mappa wood. (Middle) Input noise. (Right) Output synthetic texture that matches the appearance of the digitized sample. Note that the synthesized texture is larger than the digitized sample; our approach allows generation of as much texture as desired.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

Matlab examples

Bergen and Heeger results

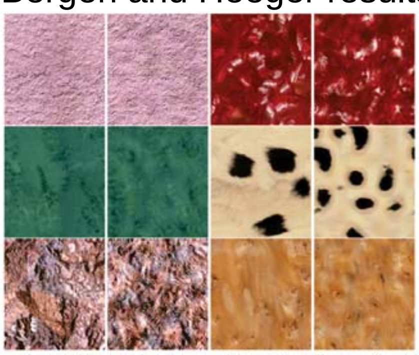


Figure 3: In each pair left image is original and right image is synthetic: stucco, iridescent ribbon, green marble, panda fur, slag stone, figured yew wood.

Bergen and Heeger failures

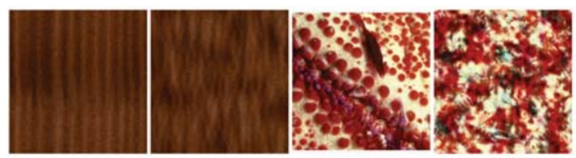


Figure 8: Examples of failures: wood grain and red coral.

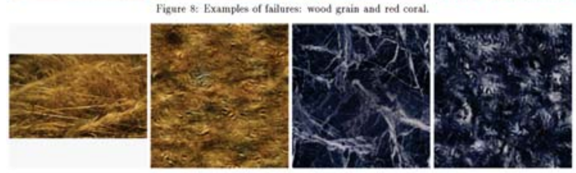


Figure 9: More failures: hay and marble.

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

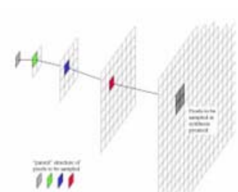


Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the "parent" structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

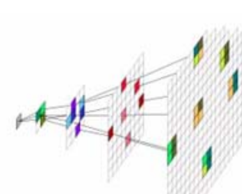
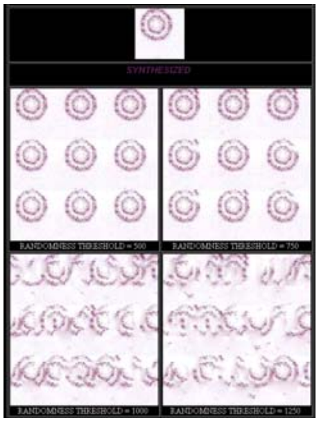
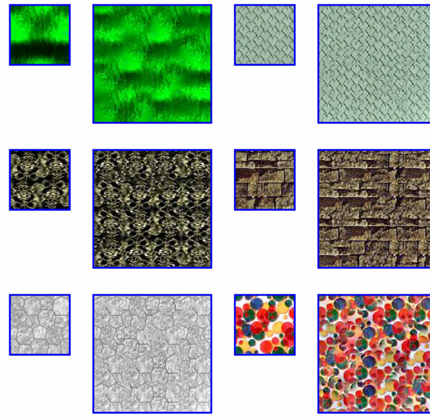


Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.



DeBonet



DeBonet

Portilla and Simoncelli

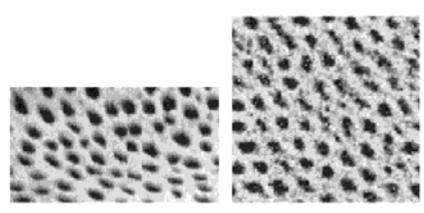
- Parametric representation.
- About 1000 numbers to describe a texture.
- Ok results; maybe as good as DeBonet.

Portilla and Simoncelli

Zhu, Wu, & Mumford, 1998

- Principled approach.
- Synthesis quality not great, but ok.

Zhu, Wu, & Mumford

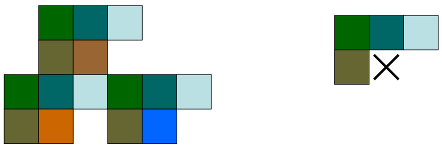


- Cheetah Synthetic

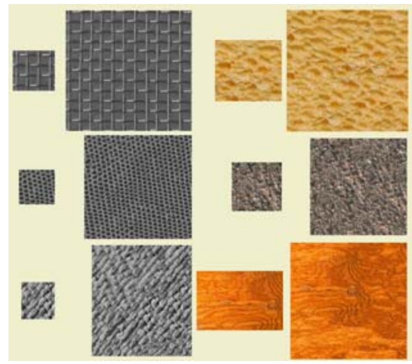
IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

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



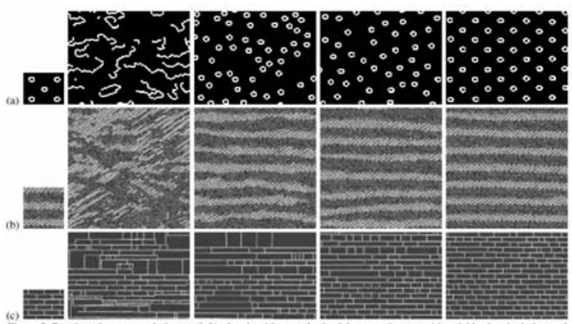


Figure 2. Results: given a sample image (left), the algorithm synthesized four new images with neighborhood windows of width 5, 11, 15, and 23 pixels respectively. Notice how perceptually intuitively the window size corresponds to the degree of randomness in the resulting textures. Input images are: (a) synthetic rings, (b) Brodatz texture D11, (c) brick wall.

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.

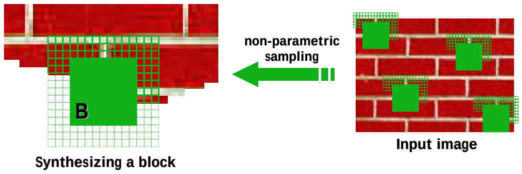
What we learned from Efros and Leung regarding texture synthesis

- 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 '99

- 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]

Efros & Leung ’99 extended



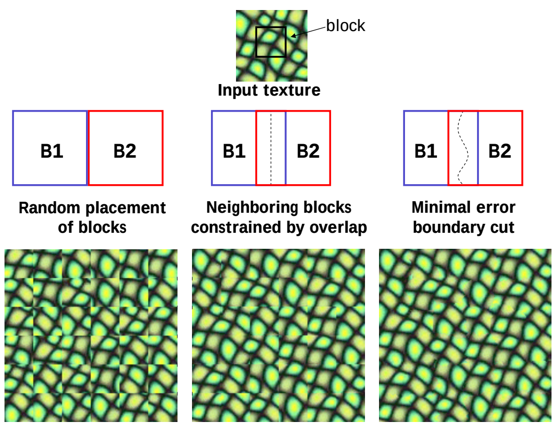
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

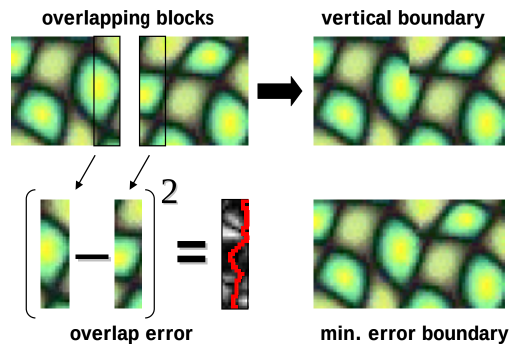
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

Image Quilting

- Idea:
  - let’s combine random block placement of Chaos Mosaic with spatial constraints of Efros & Leung
- Related Work (concurrent):
  - Real-time patch-based sampling [Liang et.al. ’01]
  - Image Analogies [Hertzmann et.al. ’01]



Minimal error boundary

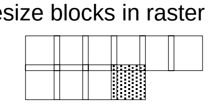


Our Philosophy

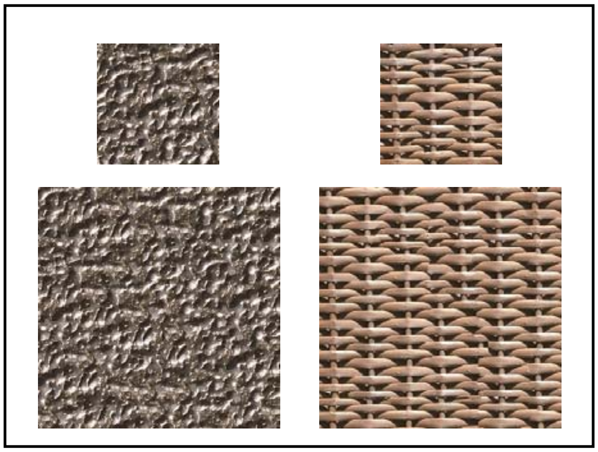
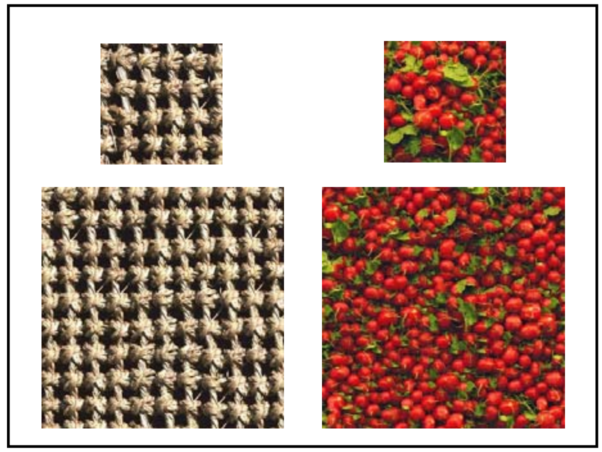
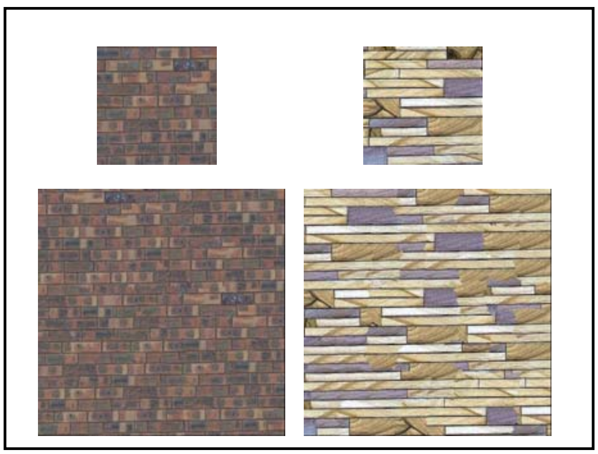
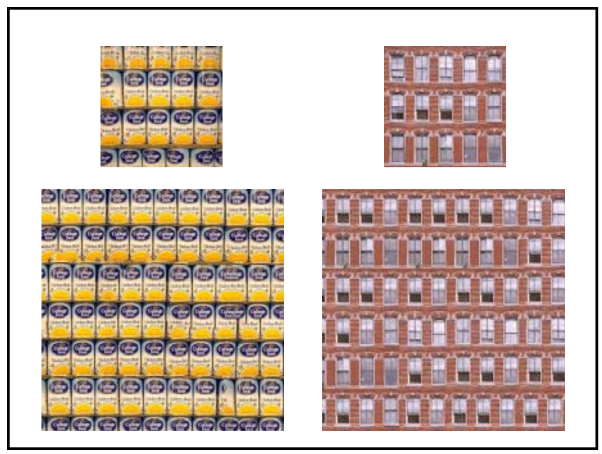
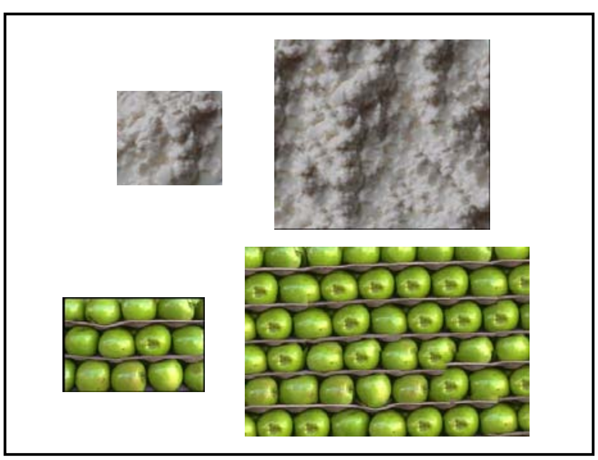
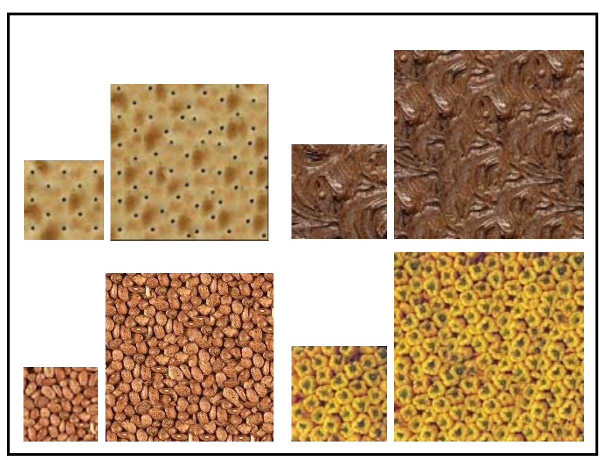
- The “Corrupt Professor’s Algorithm”:
  - Plagiarize as much of the source image as you can
  - Then try to cover up the evidence
- Rationale:
  - Texture blocks are by definition correct samples of texture so problem only connecting them together

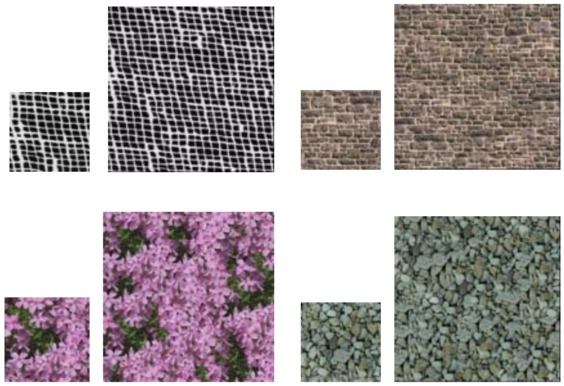
Algorithm

- 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

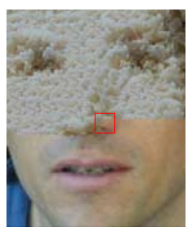









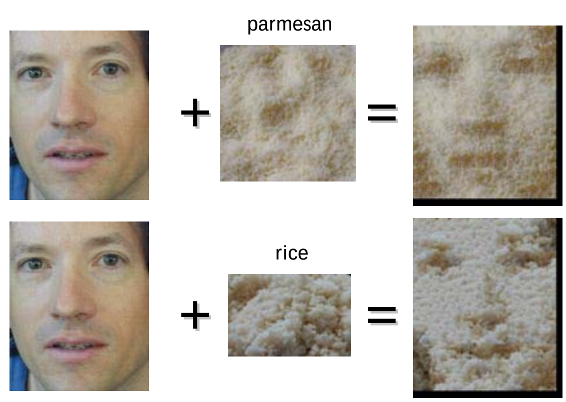
Failures
(Chernobyl Harvest)

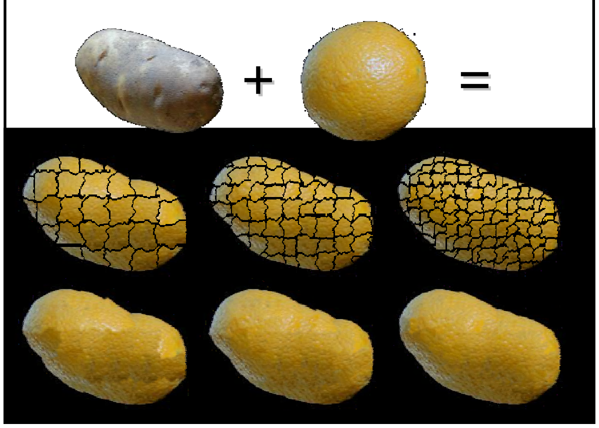
Texture Transfer

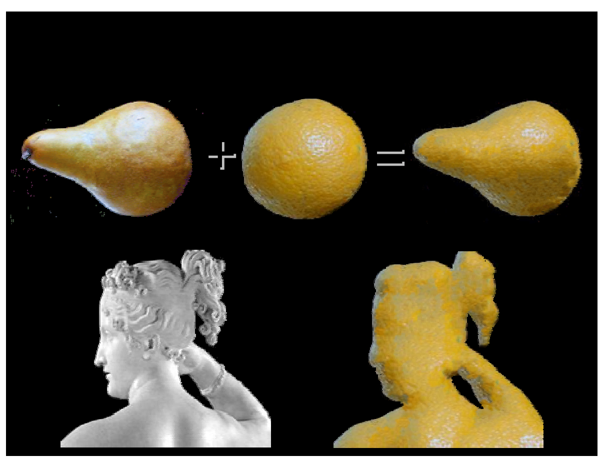
- Take the texture from one object and "paint" it onto another object
  - This requires separating texture and shape
  - That's HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading

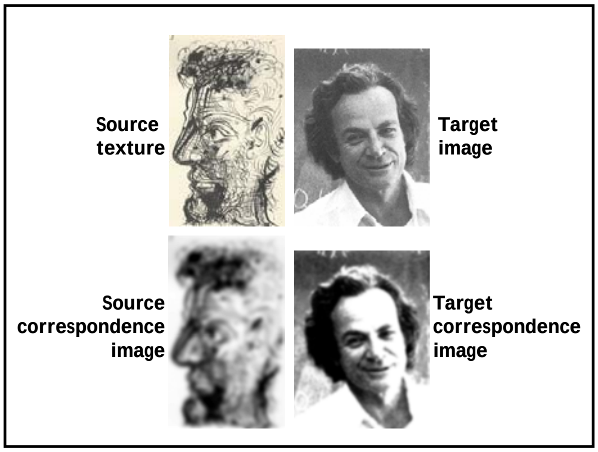


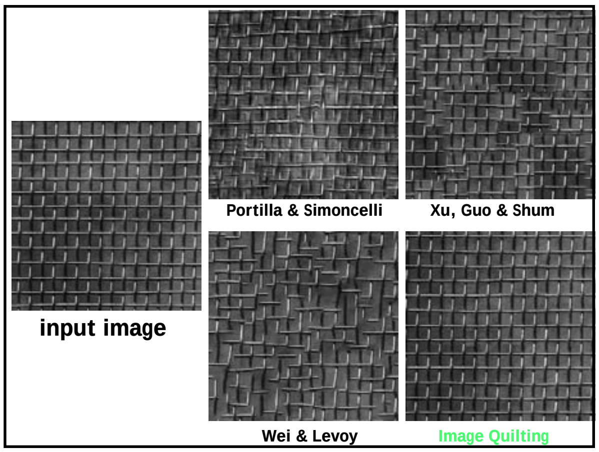
Then, just add another constraint when sampling: similarity to underlying image at that spot

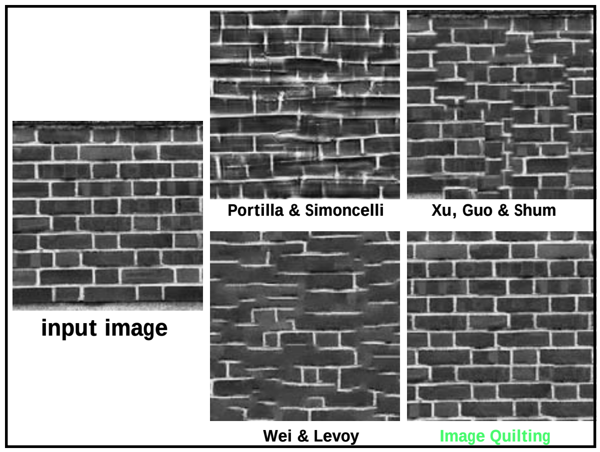


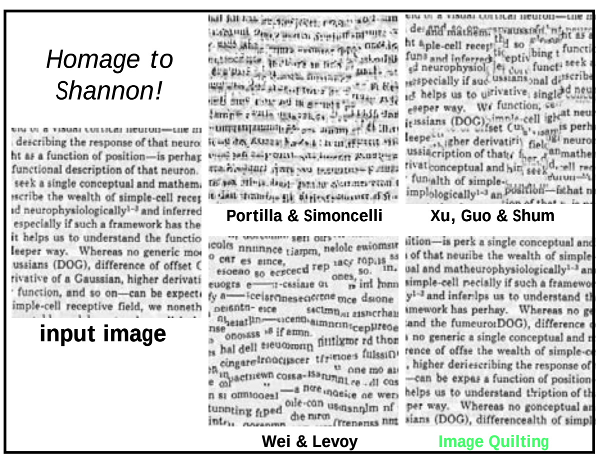






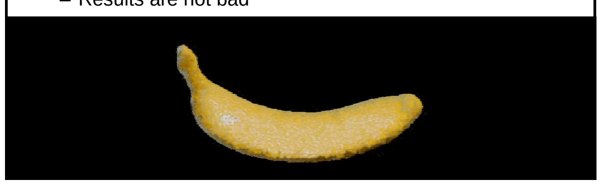






Summary of image quilting

- 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



end