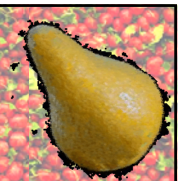


Texture

Tuesday, Sept 15
Kristen Grauman
UT-Austin

Announcements

- Write your CS login ID on the pset hardcopy

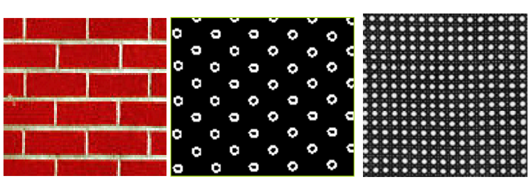
Review: last time

- Edge detection:
  - Filter for gradient
  - Threshold gradient magnitude, thin
- Binary image analysis
  - Connected components to find regions
  - Morphological operators to "clean up"

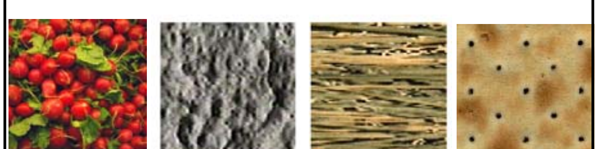
Texture

What defines a texture?

Includes: more regular patterns



Includes: more random patterns

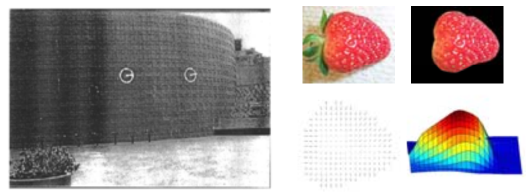


Texture-related tasks

- **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

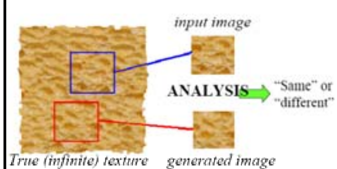


Pics from A. Loh: http://www.csse.uwa.edu.au/~angie/phdpics1.html

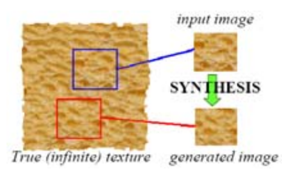
Texture-related tasks

- **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

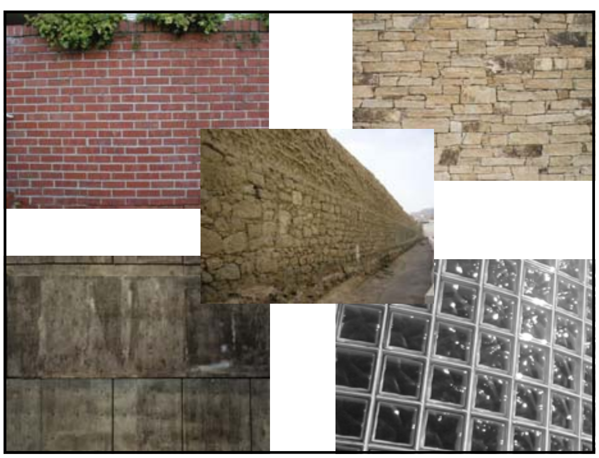
Analysis vs. Synthesis

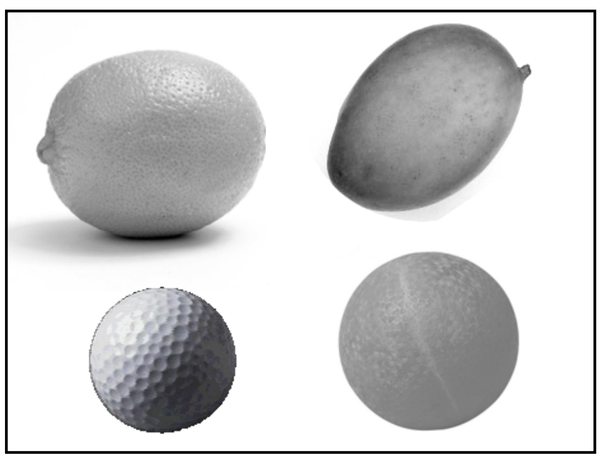


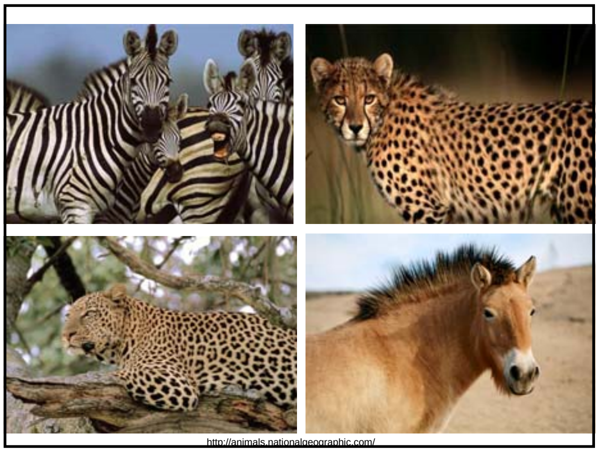
Why analyze texture?



Images:Bill Freeman, A. Efros







http://animals.nationalgeographic.com/

What kind of response will we get with an edge detector for these images?

Images from Malik and Perona, 1990

…and for this image?

Image credit: D. Forsyth

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

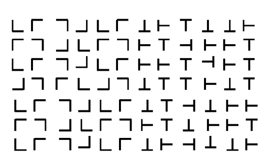
Technically:

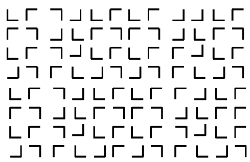
- Representation-wise, we want a feature one step above "building blocks" of filters, edges.

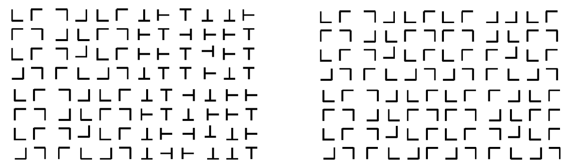
Psychophysics of texture

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

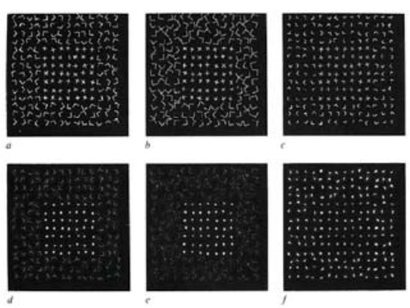
Same or different?







Capturing the local patterns with image measurements



[Bergen & Adelson, *Nature* 1988]

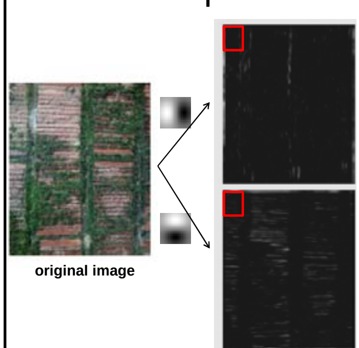
Scale of patterns influences discriminability

Size-tuned linear filters

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

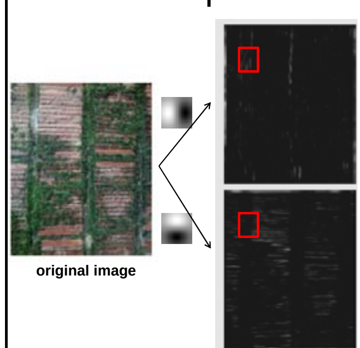


	mean d/dx value	mean d/dy value
Win. #1	4	10

⋮

statistics to summarize patterns in small windows

Texture representation: example

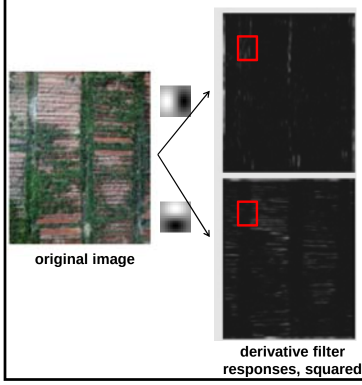


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7

⋮

statistics to summarize patterns in small windows

Texture representation: example

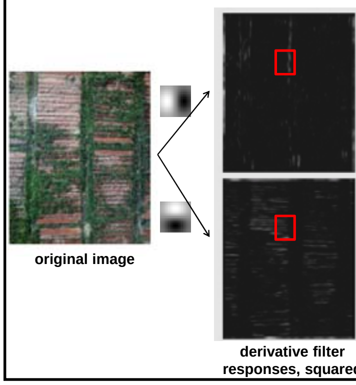


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7

⋮

statistics to summarize patterns in small windows

Texture representation: example

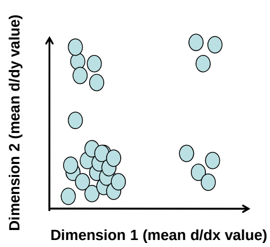


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Texture representation: example

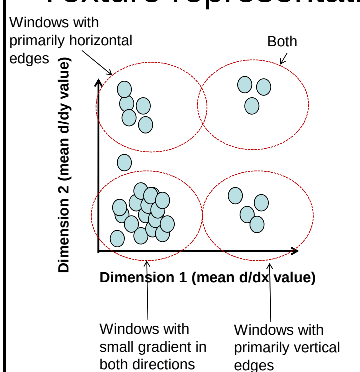


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Texture representation: example

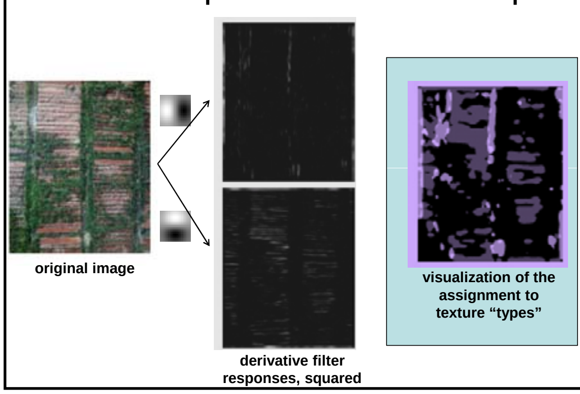


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

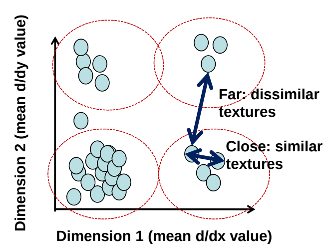
⋮

statistics to summarize patterns in small windows

Texture representation: example



Texture representation: example

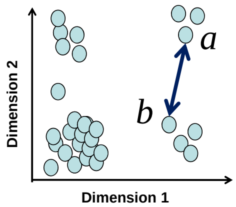


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

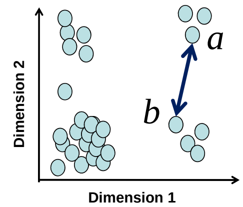
statistics to summarize patterns in small windows

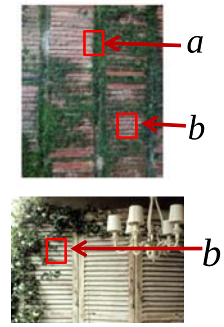
Texture representation: example



$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Texture representation: example

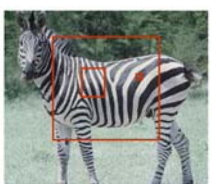




Distance reveals how dissimilar texture from window a is from texture in window b.

Texture representation: window scale

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

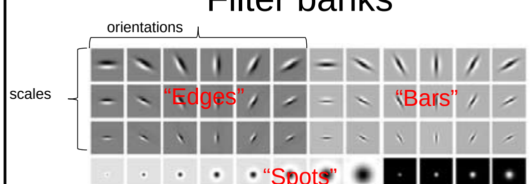


Possible to perform scale selection by looking for window scale where texture description not changing.

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

Filter banks

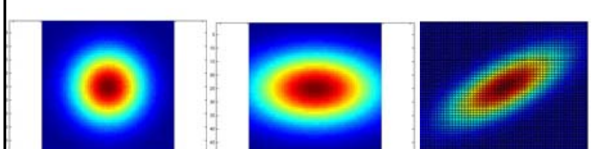


- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

Matlab code available for these examples:
http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Multivariate Gaussian

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$



$$\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}$$

Filter bank

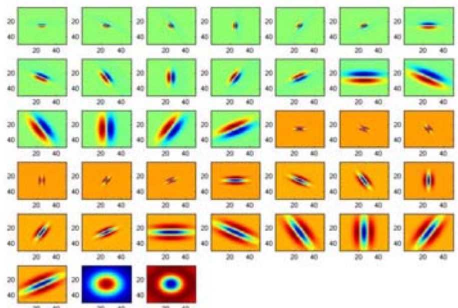
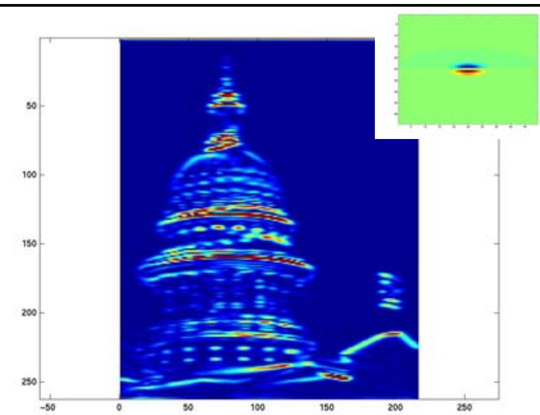
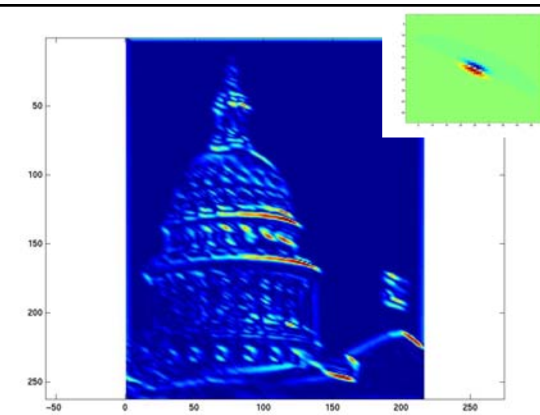
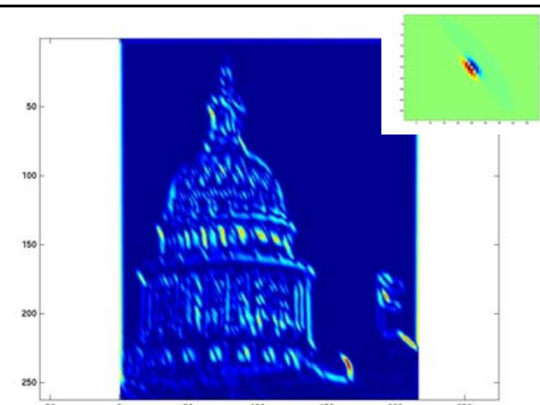
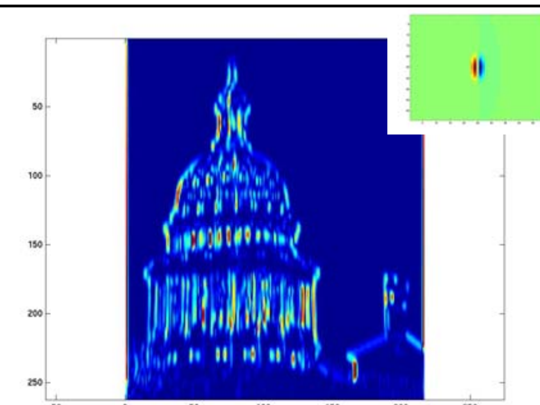


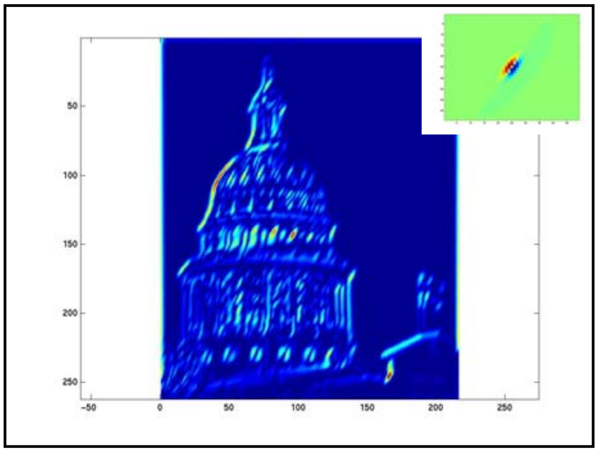
Image from http://www.texasexplorer.com/austincap2.jpg

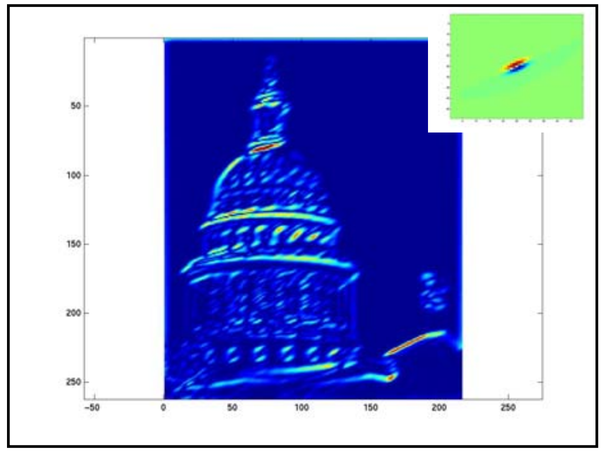


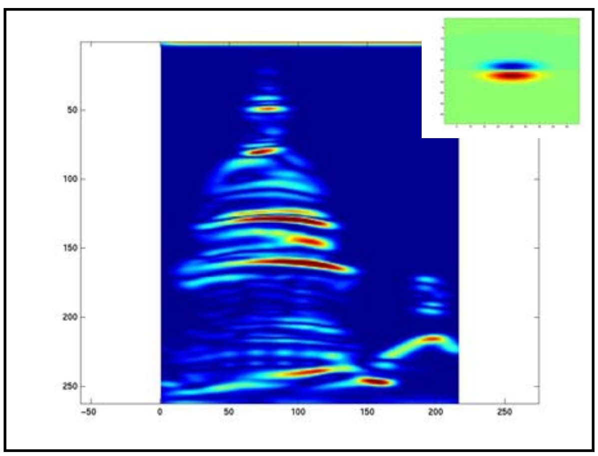


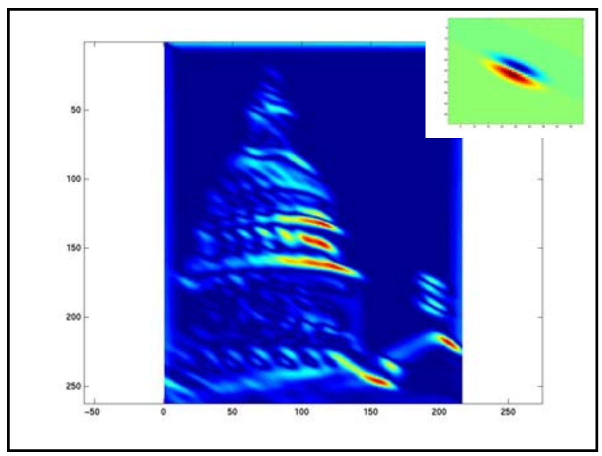


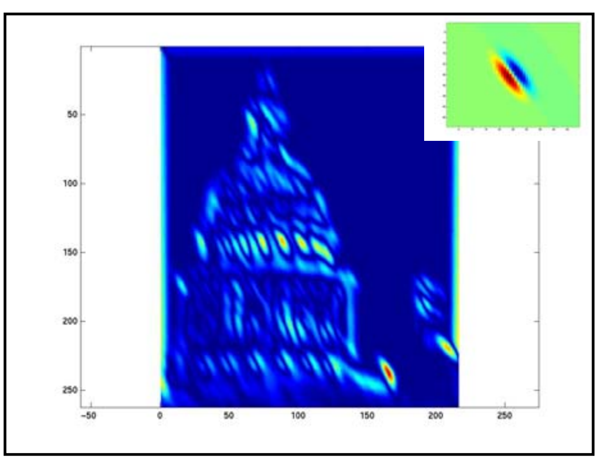


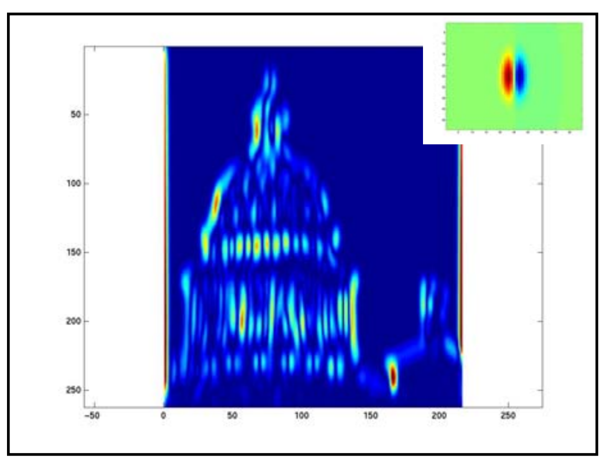

50
100
150
200
250
-50 0 50 100 150 200 250

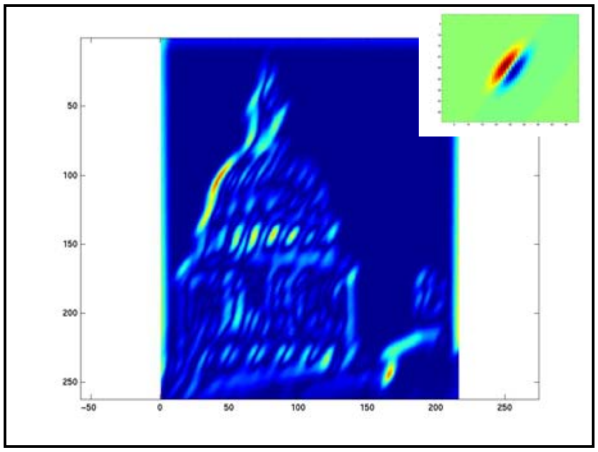

50
100
150
200
250
-50 0 50 100 150 200 250

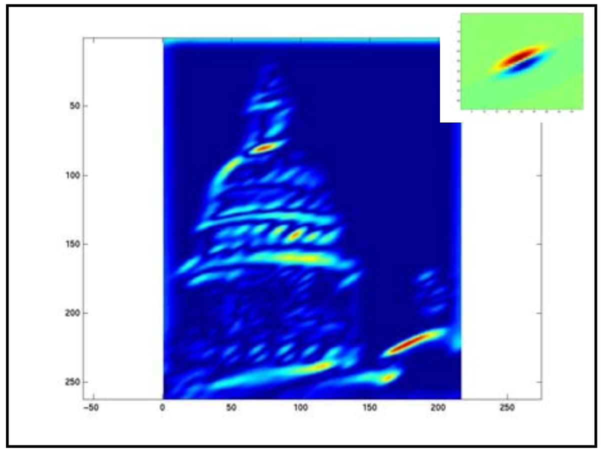

50
100
150
200
250
-50 0 50 100 150 200 250

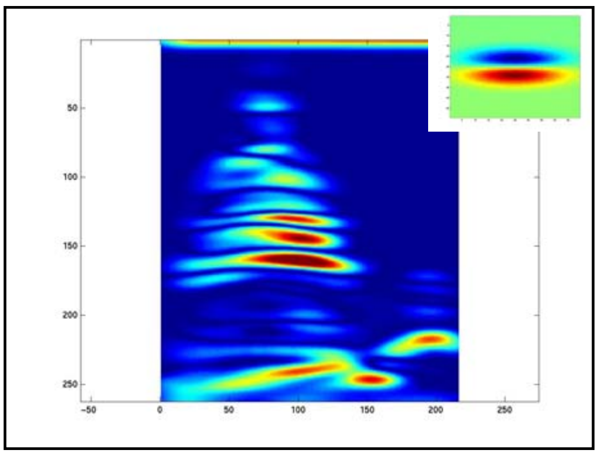

50
100
150
200
250
-50 0 50 100 150 200 250

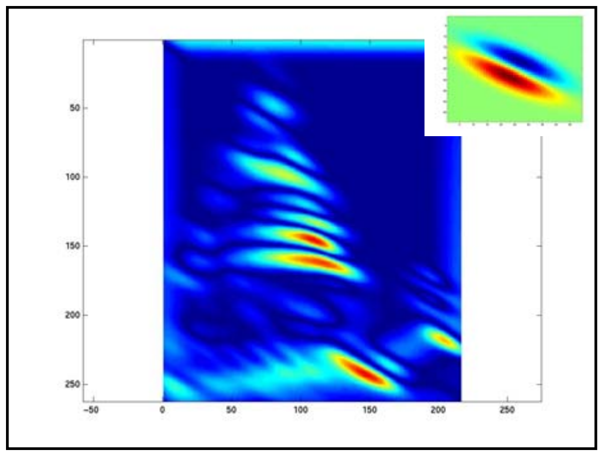

50
100
150
200
250
-50 0 50 100 150 200 250

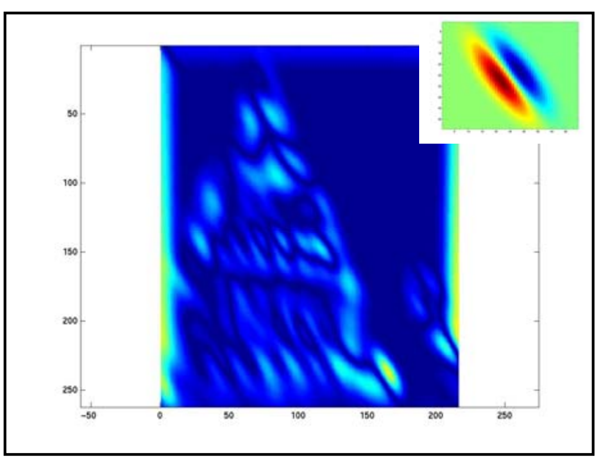

50
100
150
200
250
-50 0 50 100 150 200 250

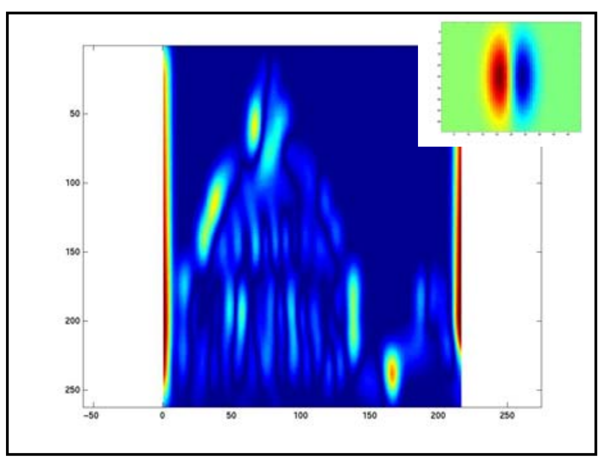

50
100
150
200
250
-50 0 50 100 150 200 250

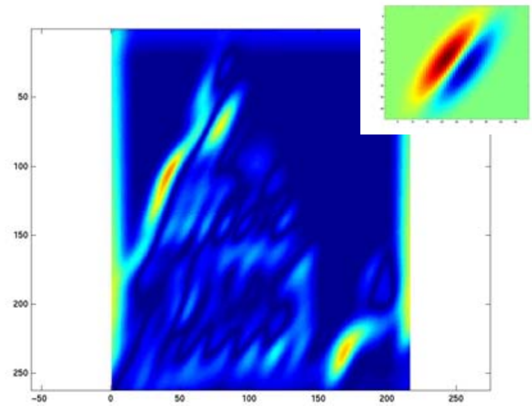

50
100
150
200
250
-50 0 50 100 150 200 250

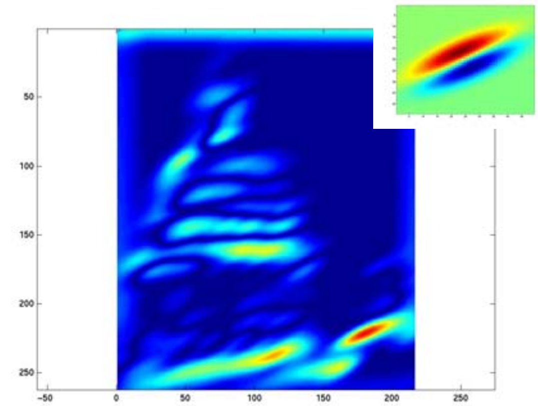

50
100
150
200
250
-50 0 50 100 150 200 250

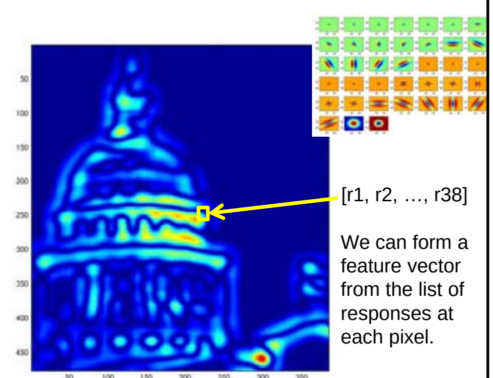

50
100
150
200
250
-50 0 50 100 150 200 250


50
100
150
200
250
-50 0 50 100 150 200 250


50
100
150
200
250
-50 0 50 100 150 200 250





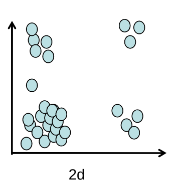


We can form a feature vector from the list of responses at each pixel.

d-dimensional features

$$D(a,b) = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2}$$

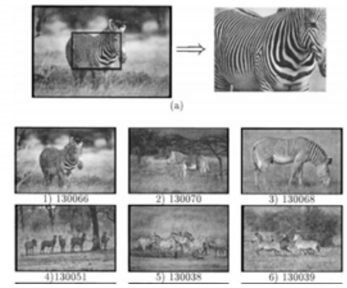
General definition of inter-point Euclidean distance (L_2).



Example uses of texture in vision: analysis

Classifying materials, “stuff”

Figure by Varma & Zisserman



Texture features for image retrieval

Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99-121, November 2000,

---

Characterizing scene categories by texture

L. W. Renninger and J. Malik. When is scene identification just texture recognition? Vision Research 44 (2004) 2301–2311

Segmenting aerial imagery by textures

http://www.airventure.org/2004/gallery/images/073104_satellite.jpg

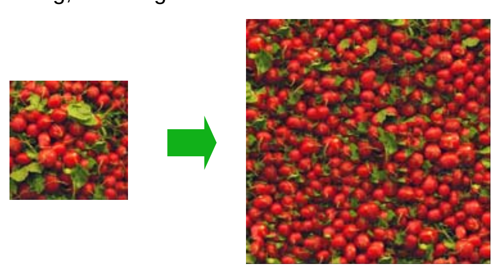
---

Texture-related tasks

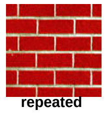
- **Shape from texture**
  - Estimate surface orientation or shape from image texture
- **Segmentation/classification** from texture cues
  - Analyze, represent texture
  - Group image regions with consistent texture
- **Synthesis**
  - Generate new texture patches/images given some examples

Texture synthesis

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



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.

Markov Chains

Markov Chain

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

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

- **Markov assumption**: each state is dependent only on the previous one
  - dependency given by a **conditional probability**:

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

Markov Chain Example: Text

"A dog is a man's best friend. It's a dog eat dog world out there."

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

	a	dog	is	man's	best	friend	it's	eat	world	out	there	.
a		2/3		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					

Source: S. Seitz

Text synthesis

Create plausible looking poetry, love letters, term papers, etc.

Most basic algorithm

1. Build probability histogram
   - find all blocks of N consecutive words/letters in training documents
   - compute probability of occurrence $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-(n-1)})$

WE NEED TO EAT CAKE

Source: S. Seitz

Text synthesis

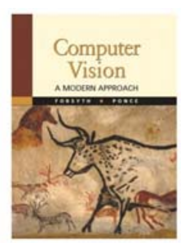
- Results:
  - *"As I've commented before, really relating to someone involves standing next to impossible."*
  - *"One morning I shot an elephant in my arms and kissed him."*
  - *"I spent an interesting evening recently with a grain of salt"*

Dewdney, "A potpourri of programmed prose and prosody" *Scientific American,* 1989.

Slide from Alyosha Efros, ICCV 1999

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 × 2k +1 × 2k +1 × 2k + 1 — required for the images separately.

Markov Random Field

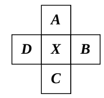
A Markov random field (MRF)

- 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*:

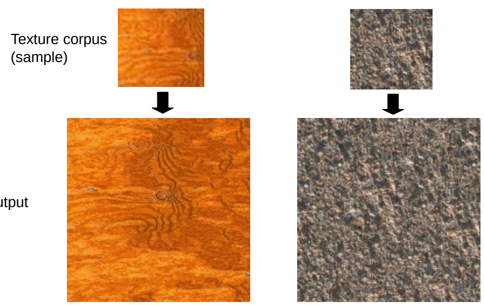
$$P(\mathbf{X}|\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$$



Source: S. Seitz

Texture Synthesis [Efros & Leung, ICCV 99]

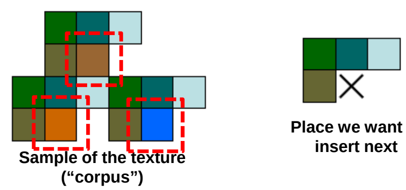
Can apply 2D version of text synthesis



Texture synthesis: intuition

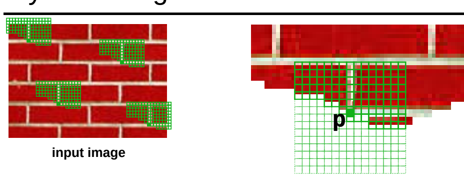
Before, we inserted the next word based on existing nearby words…

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

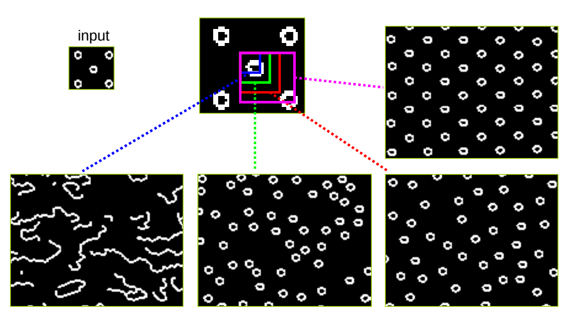


input image **synthesized image**

- What is $P(\mathrm{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- To synthesize **x**
  - pick one matching window at random
  - assign **x** to be the center pixel of that window
- An **exact** neighbourhood match might not be present, so find the **best** matches using **SSD error** and randomly choose between them, preferring better matches with higher probability

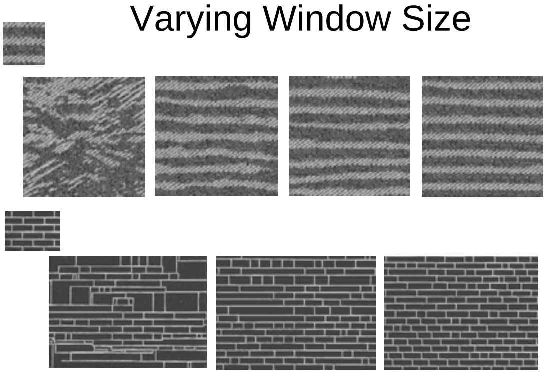
Slide from Alyosha Efros, ICCV 1999

Neighborhood Window



Slide from Alyosha Efros, ICCV 1999

Varying Window Size



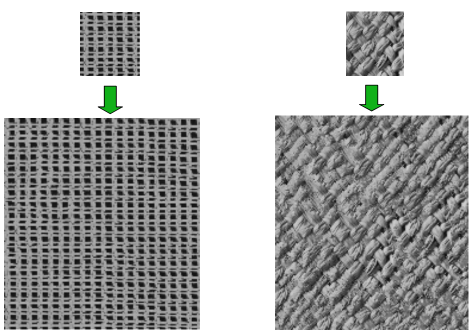
Increasing window size →

Growing Texture

- Starting from the initial image, "grow" the texture one pixel at a time

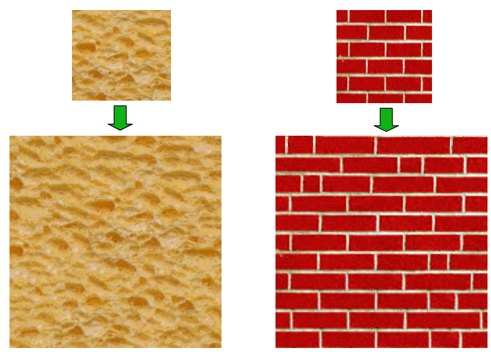
Synthesis results

french canvas   rafia weave

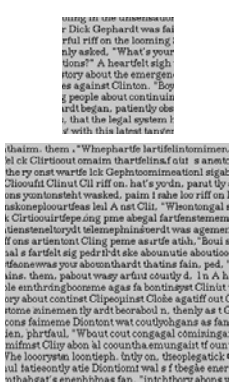


Synthesis results

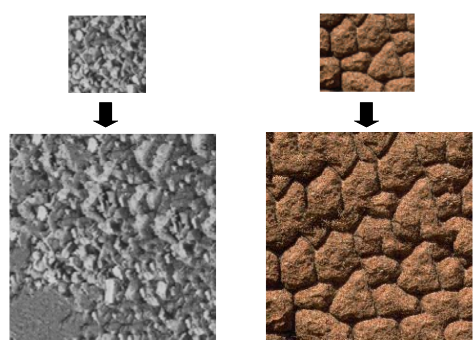
white bread   brick wall



Synthesis results

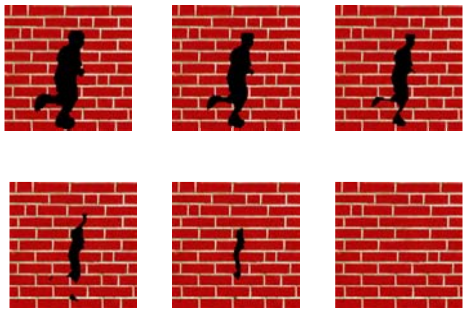


Failure Cases



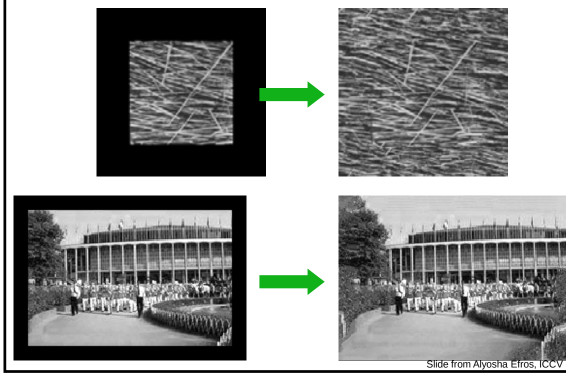
Growing garbage   **Verbatim copying**

Hole Filling



Slide from Alyosha Efros, ICCV 1999

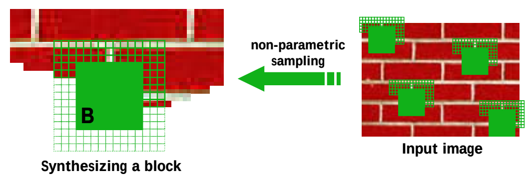
Extrapolation



Slide from Alyosha Efros, ICCV 1999

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

Image Quilting [Efros & Freeman 2001]

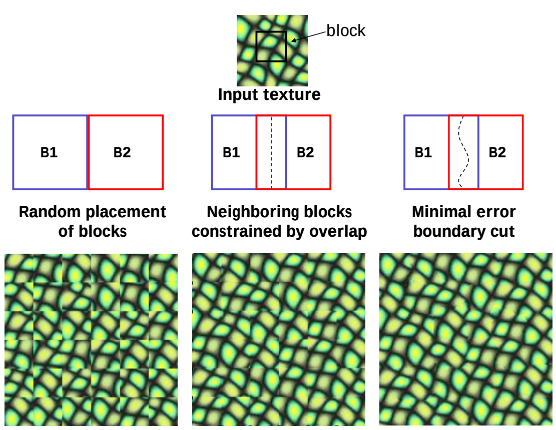


- Observation: neighbor pixels are highly correlated

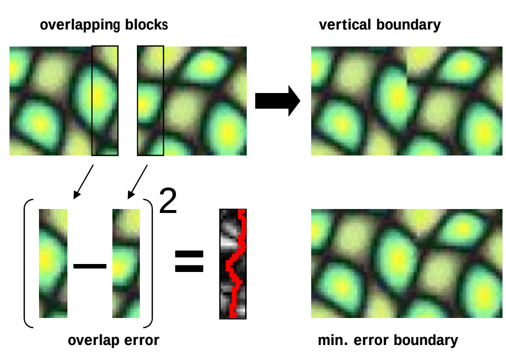
Idea: unit of synthesis = block

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

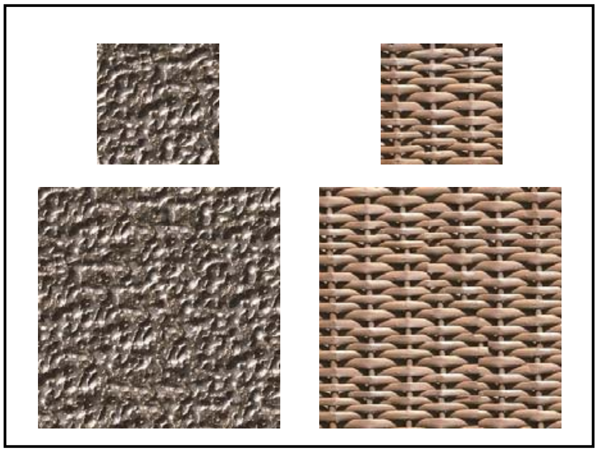
Slide from Alyosha Efros, ICCV 1999

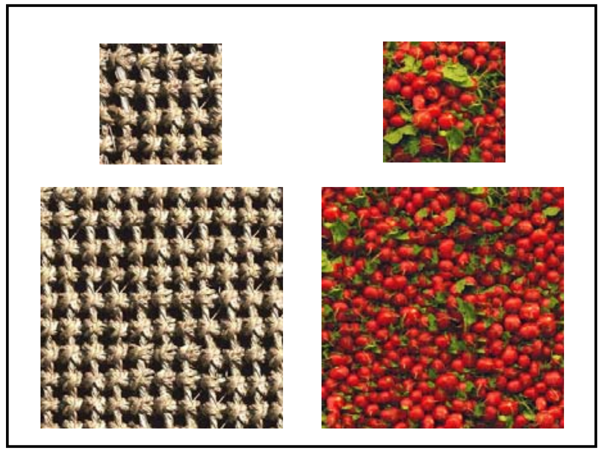


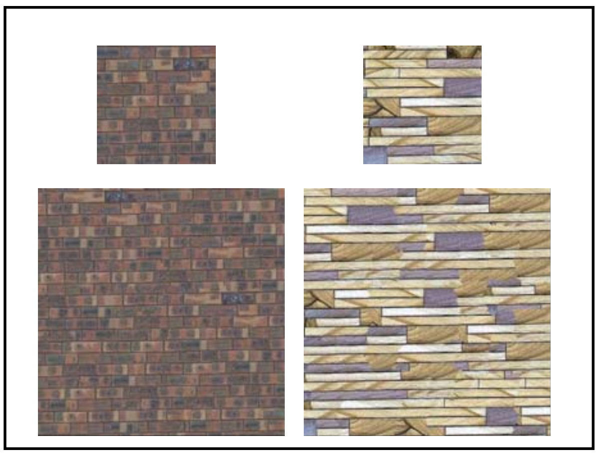
Minimal error boundary

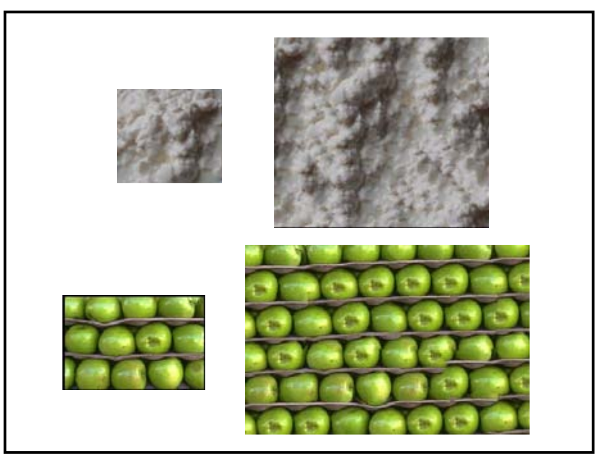


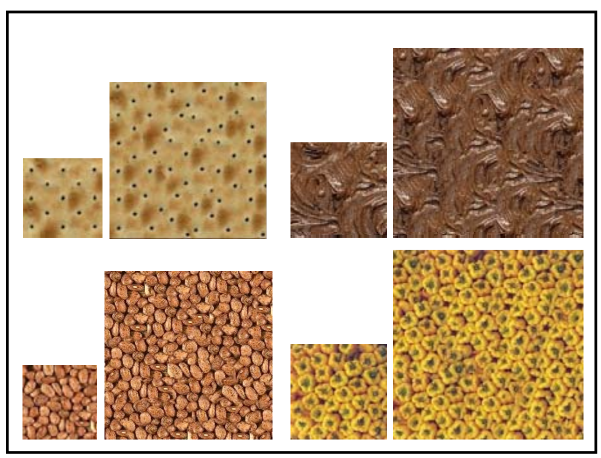
Slide from Alyosha Efros







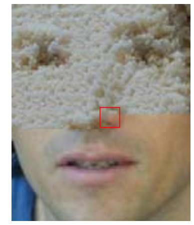




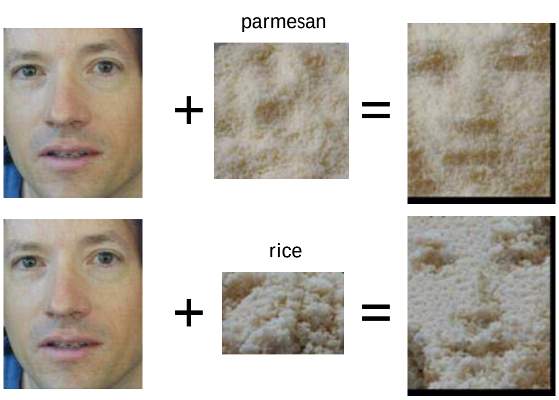

Failures
(Chernobyl Harvest)

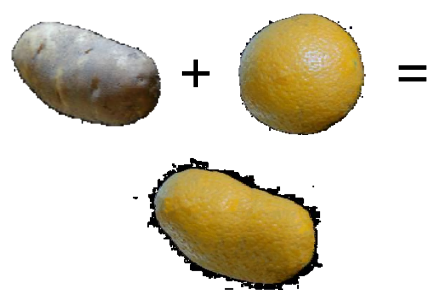
Texture Transfer

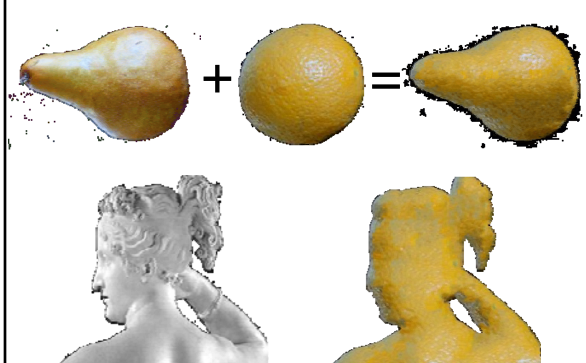
- Take the texture from one object and "paint" it onto another object
  - This requires separating texture and shape
  - That's HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading



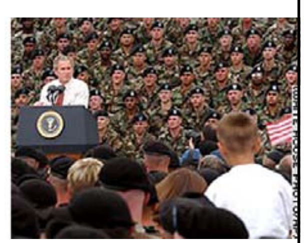
- **Then, just add another constraint when sampling: similarity to underlying image at that spot**







(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/

Synthesizing textures when constructing 3d models of archaeological sites

Figure 12. The Nymphaeum at the upper agora of Sagalassos with differently textured pillars. Overview of one half of the building (symmetric)

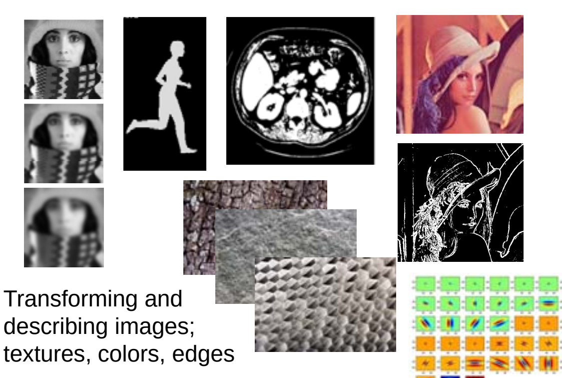
Figure 14. Nymphaeum pillars and back wall fragments in detail

A. Zalesny et al., Realistic Textures for Virtual Anastylosis

Summary

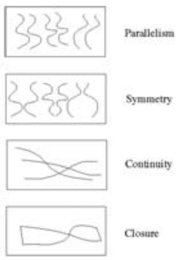
- Texture is a useful property that is often indicative of materials, appearance cues
- **Texture representations** attempt to summarize repeating patterns of local structure
- **Filter banks** useful to measure redundant variety of structures in local neighborhood
  - Feature spaces can be multi-dimensional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

So far: features and filters



Transforming and describing images; textures, colors, edges

Next: Grouping & fitting



[fig from Shi et al]

Clustering, segmentation, fitting; what parts belong together?

Coming up

- Next time: Segmentation and grouping
  - For Thurs: read F&P Chapter 14

- Reminder:
  - Problem set 1 due Sept 21 (Monday) 11:59 PM