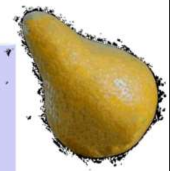



Texture

Thurs Feb 1, 2018

Kristen Grauman
UT Austin

Announcements

- Reminder: A1 due this Friday

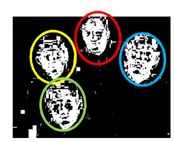
Recap: last time

- Edge detection:
  – Filter for gradient
  – Threshold gradient magnitude, thin
- Chamfer matching
  – to compare shapes (in terms of edge points)
  – Distance transform
- Binary image analysis
  – Thresholding
  – Morphological operators to "clean up"
  – Connected components to find regions

Issues

- What to do with "noisy" binary outputs?
  – Holes
  – Extra small fragments

- How to demarcate multiple regions of interest?
  – Count objects
  – Compute further features per object



Connected components

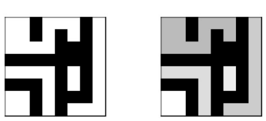
- Identify distinct regions of "connected pixels"

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

a) binary image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

b) connected components labeling

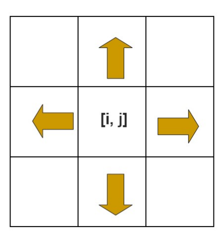


c) binary image and labeling, expanded for viewing

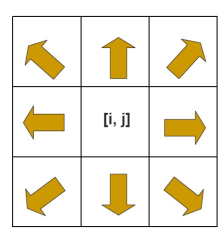
Shapiro and Stockman

Connectedness

- Defining which pixels are considered neighbors



4-connected

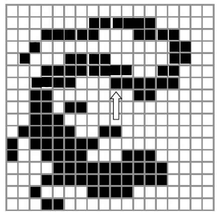


8-connected

Source: Chaitanya Chandra

Connected components

- We'll consider a sequential algorithm that requires only 2 passes over the image.
- **Input**: binary image
- **Output**: "label" image, where pixels are numbered per their component
- Note: foreground here is denoted with black pixels.



Sequential connected components

- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object New object

What happens in these cases?

Adapted from J. Neira

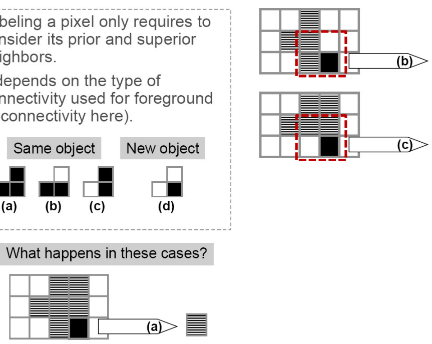
Sequential connected components

- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object New object

(a) (b) (c) (d)

What happens in these cases?



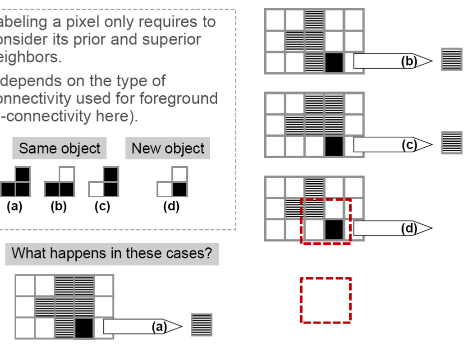
Sequential connected components

- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object New object

(a) (b) (c) (d)

What happens in these cases?



Sequential connected components

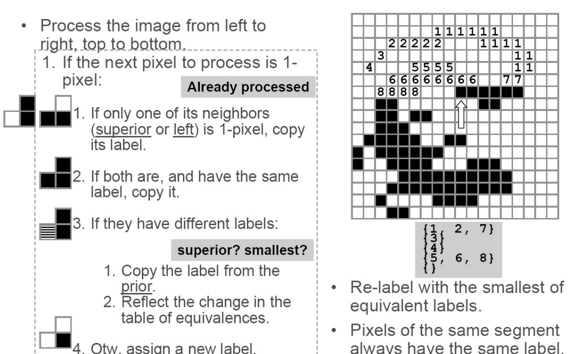
- Process the image from left to right, top to bottom.
  1. If the next pixel to process is 1-pixel:

     Already processed

     1. If only one of its neighbors (superior or left) is 1-pixel, copy its label.
     2. If both are, and have the same label, copy it.
     3. If they have different labels:

        superior? smallest?

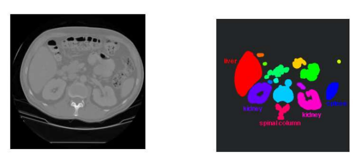
        1. Copy the label from the prior.
        2. Reflect the change in the table of equivalences.
     4. Otw, assign a new label.
  2. More pixels? Go to step 1.



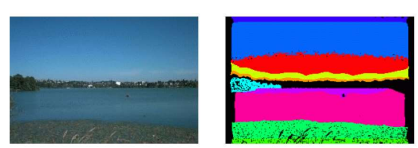
{1, 2, 7}
{3}
{4}
{5, 6, 8}
{}

- Re-label with the smallest of equivalent labels.
- Pixels of the same segment always have the same label.

Connected components



connected components of 1's from thresholded image



connected components of cluster labels

Slide credit: Pinar Duygulu

Region properties

- Given connected components, can compute simple features per blob, such as:
  - Area (num pixels in the region)
  - Centroid (average x and y position of pixels in the region)
  - Bounding box (min and max coordinates)
  - Circularity (ratio of mean dist. to centroid over std)

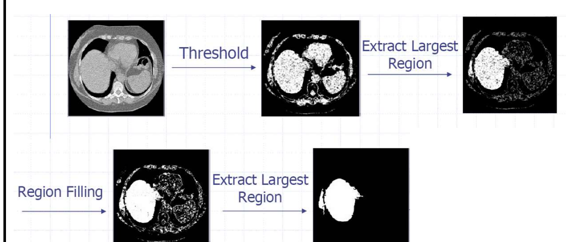
Binary image analysis: basic steps (recap)

- Convert the image into binary form
  - Thresholding
- Clean up the thresholded image
  - Morphological operators
- Extract separate blobs
  - Connected components
- Describe the blobs with region properties

Example using binary image analysis: OCR

[Luis von Ahn et al. http://recaptcha.net/learnmore.html]

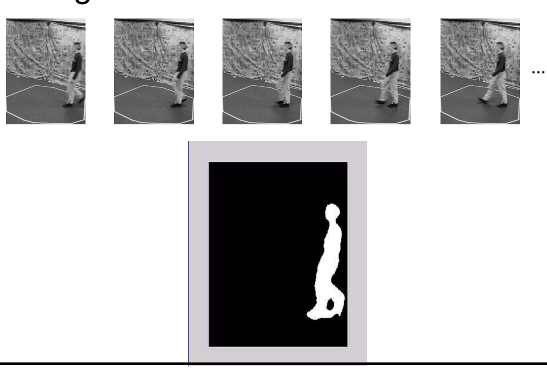
Example using binary image analysis: segmentation of a liver



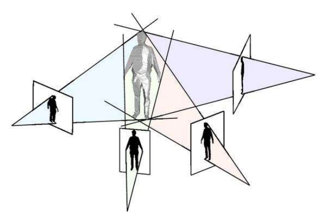
Slide credit: Li Shen

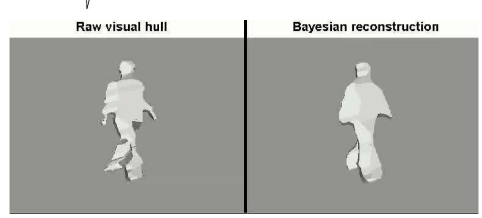
Application by Jie Zhu, Cornell University

Example using binary image analysis: Bg subtraction + blob detection



Visual hulls





Example using binary image analysis: Bg subtraction + blob detection

University of Southern California
http://iris.usc.edu/~icohen/projects/vace/detection.htm

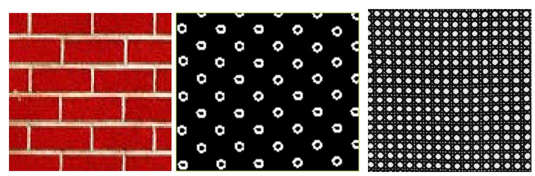
Binary images

- Pros
  - Can be fast to compute, easy to store
  - Simple processing techniques available
  - Lead to some useful compact shape descriptors
- Cons
  - Hard to get “clean” silhouettes
  - Noise common in realistic scenarios
  - Can be too coarse of a representation
  - Not 3d

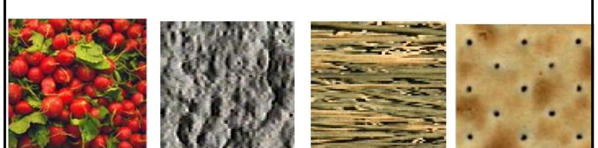
Today: Texture

What defines a texture?

Includes: more regular patterns



Includes: more random patterns



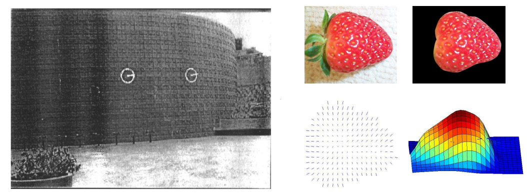
Scale and texture

Texture-related tasks

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

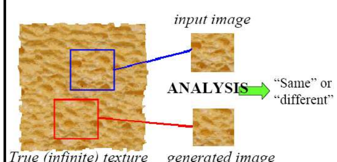
Shape from texture

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

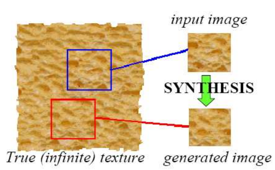


Pics from A. Loh: http://www.csse.uwa.edu.au/~angie/phdpics1.html

Analysis vs. Synthesis



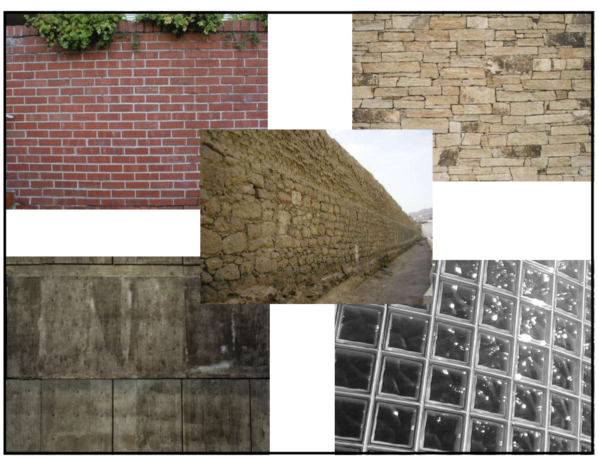
Why analyze texture?

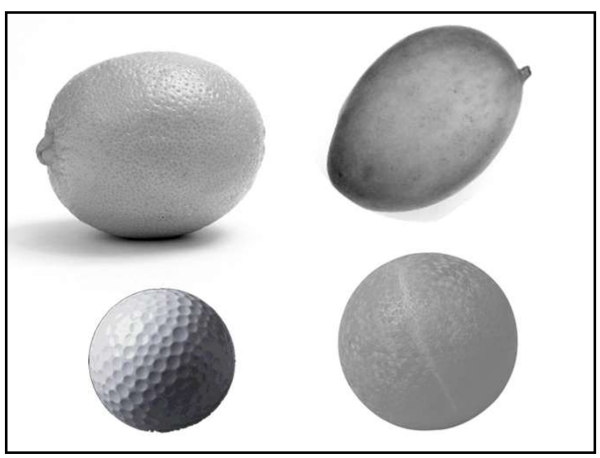


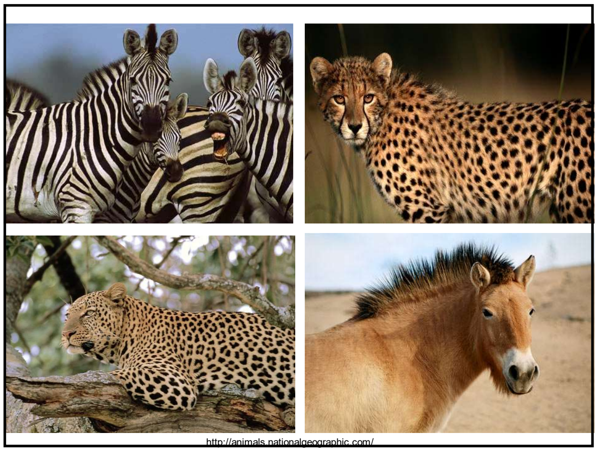
Images:Bill Freeman, A. Efros

Texture-related tasks

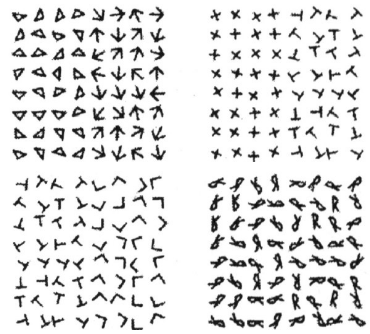
- **Shape from texture**
  - Estimate surface orientation or shape from image texture
- **Segmentation/classification** from texture cues
  - Analyze, represent texture
  - Group image regions with consistent texture
- **Synthesis**
  - Generate new texture patches/images given some examples







http://animals.nationalgeographic.com/



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

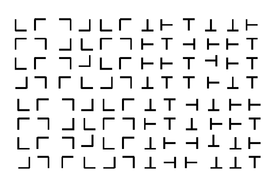
Technically:

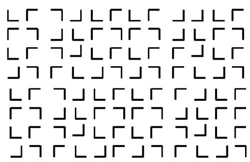
- Representation-wise, we want a feature one step above "building blocks" of filters, edges.

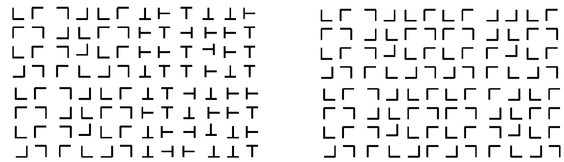
Psychophysics of texture

- 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]

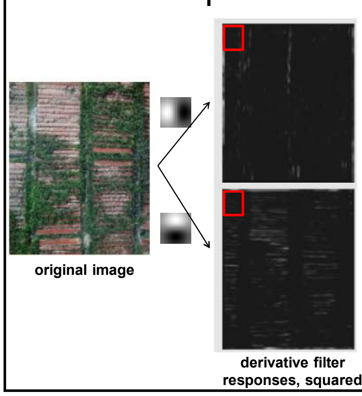
Scale of patterns influences discriminability

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, e.g.,
    - Mean, standard deviation
    - Histogram
    - Histogram of "prototypical" feature occurrences

Texture representation: example

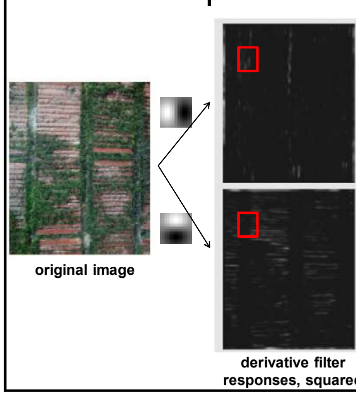


	mean d/dx value	mean d/dy value
Win. #1	4	10

⋮

statistics to summarize patterns in small windows

Texture representation: example

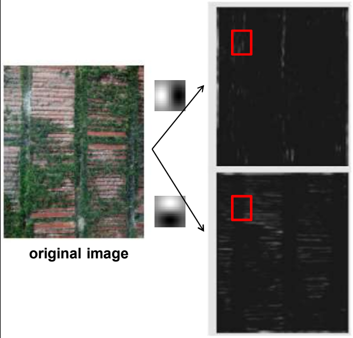


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7

⋮

statistics to summarize patterns in small windows

Texture representation: example



original image

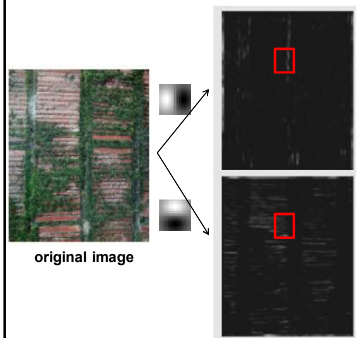
derivative filter responses, squared

	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



original image

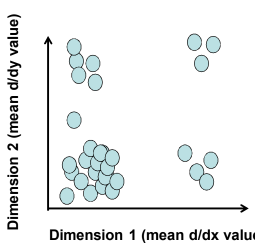
derivative filter responses, squared

	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

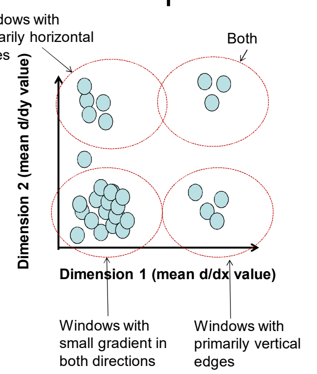


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Texture representation: example

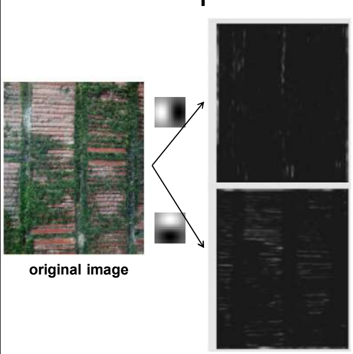


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Texture representation: example

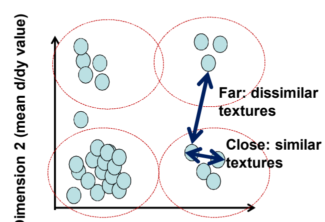


original image

derivative filter responses, squared

visualization of the assignment to texture "types"

Texture representation: example

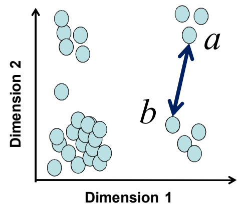


	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

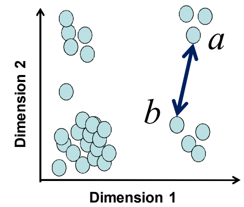
statistics to summarize patterns in small windows

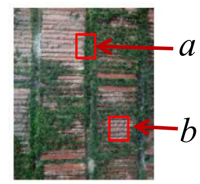
Texture representation: example



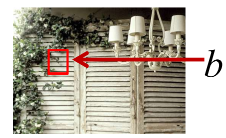
$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Texture representation: example



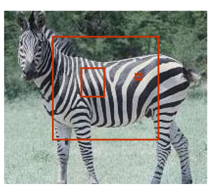


Distance reveals how dissimilar texture from window a is from texture in window b.



Texture representation: window scale

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

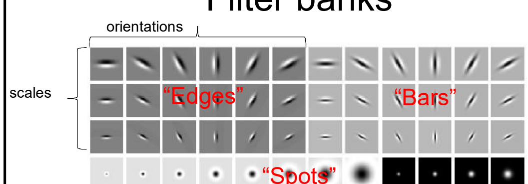


Possible to perform **scale selection** by looking for window scale where texture description not changing.

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

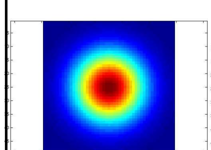


- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

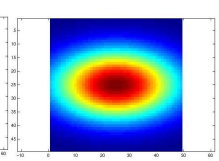
Matlab code available for these examples:
http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Multivariate Gaussian

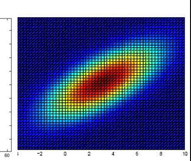
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$



$$\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}$$



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



$$\Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}$$

Filter bank

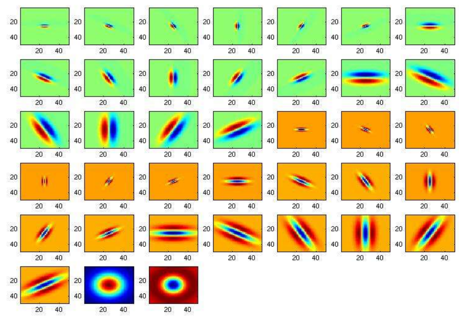
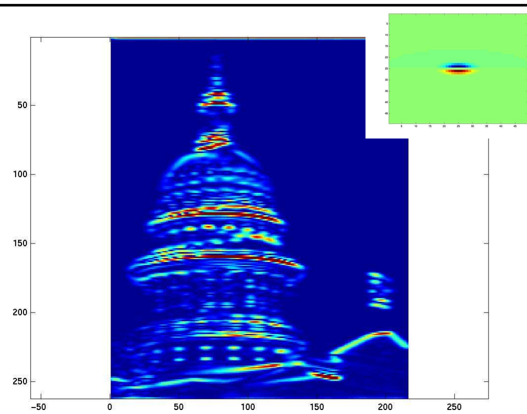
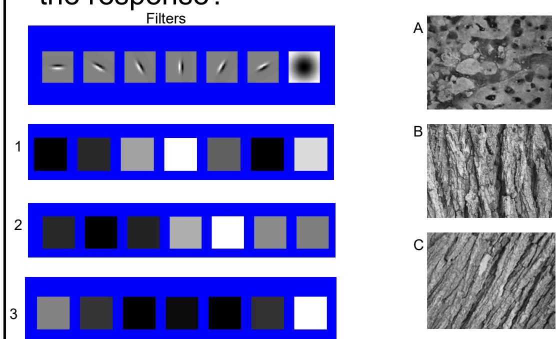



Image from http://www.texasexplorer.com/austincap2.jpg

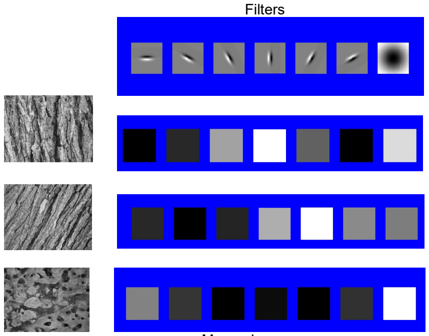


You try: Can you match the texture to the response?

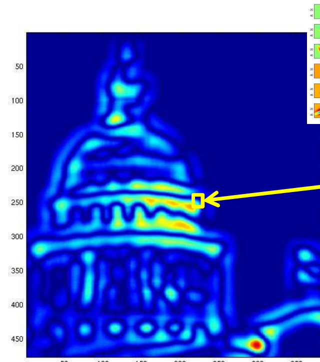


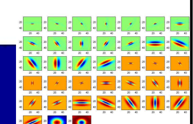
Derek Hoiem

Representing texture by mean abs response



Derek Hoiem





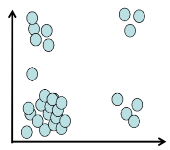
[r1, r2, …, r38]

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

d-dimensional features

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

Euclidean distance (L_2)

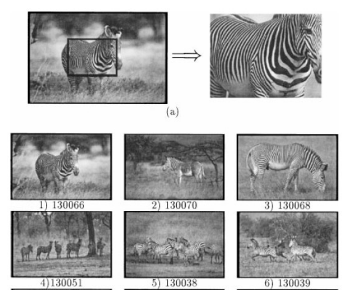


2d

Example uses of texture in vision: analysis

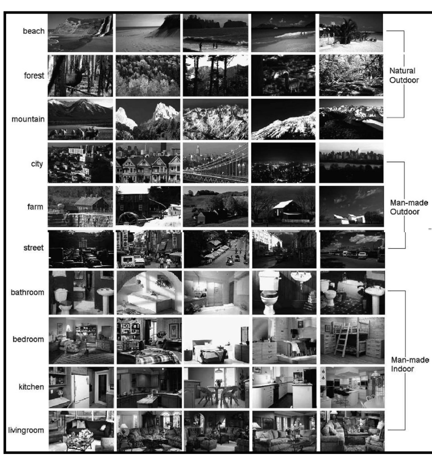
Classifying materials, “stuff”

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

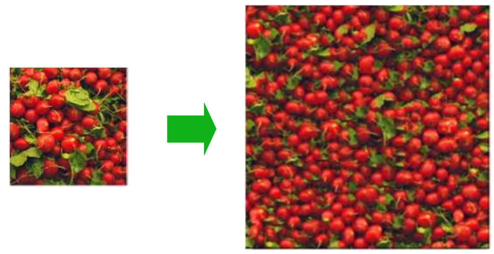
http://www.airventure.org/2004/gallery/images/073104_satellite.jpg

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

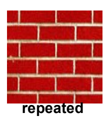
- Goal: create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces



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.


repeated


stochastic


Both?

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 Chain Example: Text

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

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

$\mathbf{x}_{t-1}$ \ $\mathbf{x}_t$	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

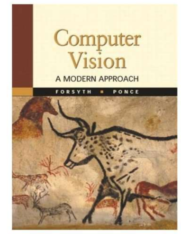
- Results:
  - *"As I've commented before, really relating to someone involves standing next to impossible."*
  - *"One morning I shot an elephant in my arms and kissed him."*
  - *"I spent an interesting evening recently with a grain of salt"*

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 a 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/

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.

Synthesized UTCS code of conduct

- You should be on the day your assignment is due.
- Remember that the work available to the bookstore, buy books, read them, and write some code without ever signing up for a class.
- In this document, a group of the grade will go down rather than up.
- To make this process work, you have made prior arrangements with the instructor.
- But remember that the instructor responded to such issues.

Synthesized UTCS code of conduct

- For example, don't write to your instructor.
- For example, don't write to your instructor.
- But, whenever you do in the field.
- Classes that use different exams each semester may have very different score distributions from one semester to the day your assignment is due.
- (It's on the class to file a complaint about the grading of your work, you have the right to expect your instructor has read a lot of problems, and then chosen, from all of that material, 14 weeks of the one week from the time of preregistration.

Markov Random Field

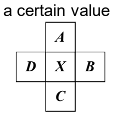
A Markov random field (MRF)

- 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*:

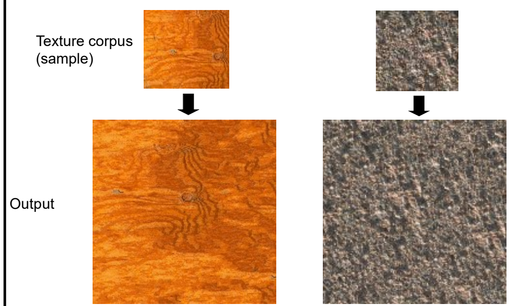
$$P(\mathrm{X}|\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D})$$



Source: S. Seitz

Texture Synthesis [Efros & Leung, ICCV 99]

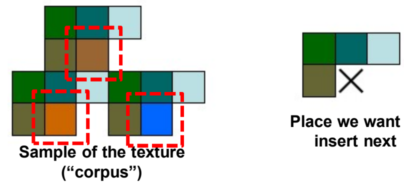
Can apply 2D version of text synthesis



Texture synthesis: intuition

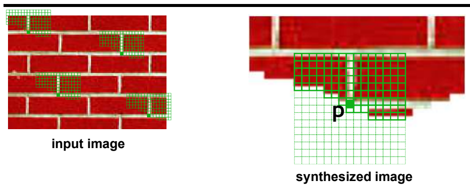
Before, we inserted the next word based on existing nearby words…

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.

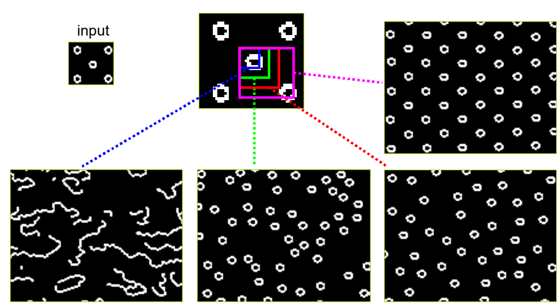
Synthesizing One Pixel



- 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

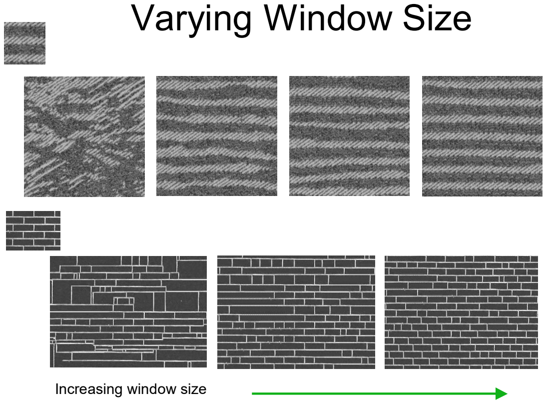
Slide from Alyosha Efros, ICCV 1999

Neighborhood Window



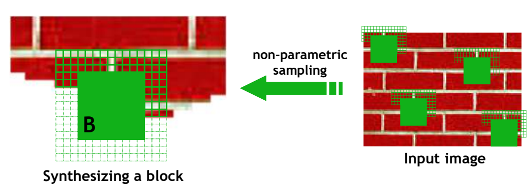
Slide from Alyosha Efros, ICCV 1999

Varying Window Size



Slide from Alyosha Efros, ICCV 1999

Image Quilting [Efros & Freeman 2001]

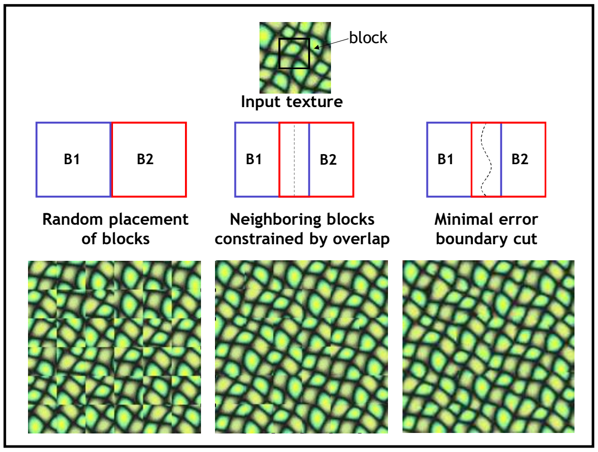


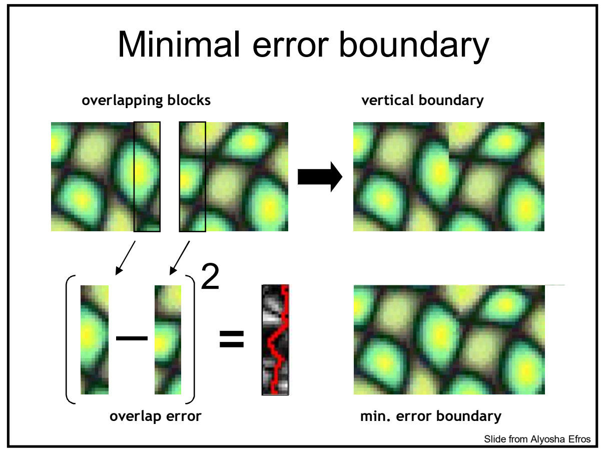
- Observation: neighbor pixels are highly correlated

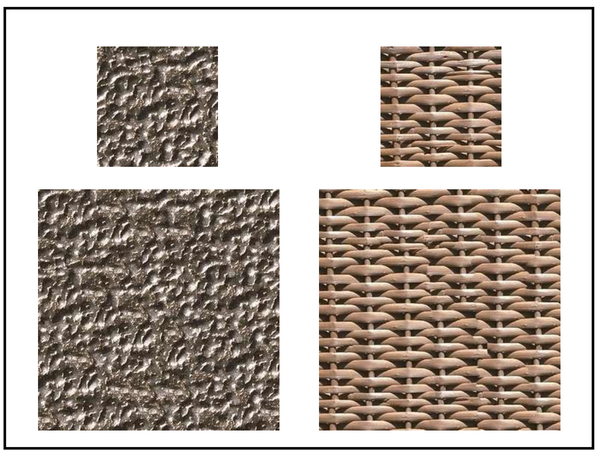
Idea: unit of synthesis = block

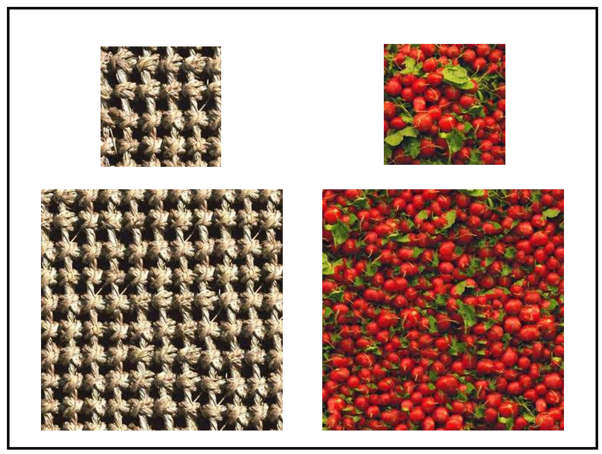
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

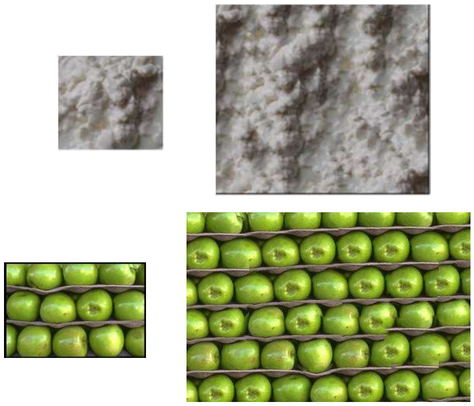
Slide from Alyosha Efros, ICCV 1999


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


Minimal error boundary
overlapping blocks
vertical boundary
2
=
overlap error
min. error boundary
Slide from Alyosha Efros



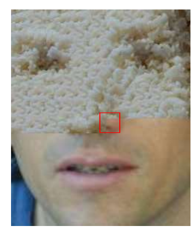




Failures
(Chernobyl Harvest)

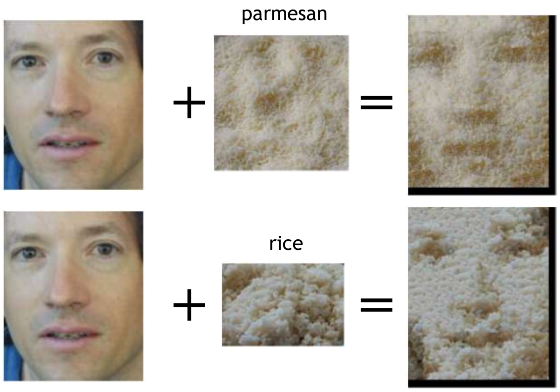
Texture Transfer

- 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

Slide credit: Freeman & Efros



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