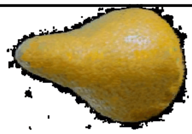


Lecture 6: Texture

Tuesday, Sept 18

Computer Vision
Fall 2007

Tues/Thurs 12:30 – 2:00 pm
Painter Hall 1 (PAI, down the stairs from the front entrance)

CS 378, Unique # 56705 (undergrads)
CS 395T, Unique# 56850 (grads)

Instructor: Prof. Kristen Grauman
Email: grauman -- put the at sign -- cs.utexas.edu
Office hours: Thurs 2:00-4:00 pm in TAY 4.118

TA: Sudheendra Vijayanarasimhan
Email: svnaras -- put the at sign -- cs.utexas.edu
Office hours: Mon 1:00-2:00 pm, Wed 12:00-1:00 pm in ENS 31NQ

The TA station is in the basement of ENS inside room 31NR. Directions to the TA stations are posted right outside the basement elevator, and also outside room 31NR.

Announcements Overview Requirements Schedule Links Papers

Announcements

Updated Problem set 1 is due Tuesday Sept 25.

Overview

Billions of images are hosted publicly on the web--how can you find one that "looks like" some image you are interested in? Could we interact with a computer in richer ways than a keyboard and mouse, perhaps with natural gestures or simply facial expressions? How can a robot identify objects in complex environments, or navigate uncharted territory? How can a video camera in the operating room help a surgeon plan a procedure more safely, or assist a radiologist in more efficiently detecting a tumor? Given some video sequence of a scene, can we synthesize new virtual views from arbitrary viewpoints that make a viewer feel as if they are in the movie?

Computer vision is at the heart of many such questions: the goal is to develop methods that enable a machine to "understand" or analyze images and videos. In this introductory computer vision course, we will explore various fundamental topics in the area, including image formation, feature detection, segmentation, multiple view geometry, recognition and learning, and motion and tracking. An outline of the syllabus is here.

This course is cross-listed for upper level undergraduate (CS 378) and graduate (CS 395T) students. Additional work is required of graduate students (see below).

Prerequisites

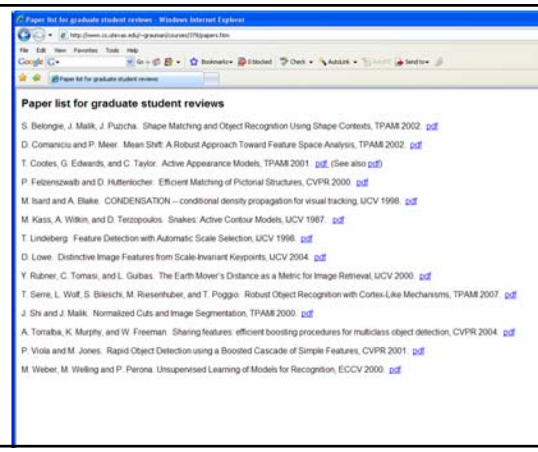
Basic knowledge of probability and linear algebra, data structures, algorithms, programming experience.

Previous experience with image processing, machine learning, and statistics will be useful but is not required. Problem sets will include some Matlab programming.

Computer Vision
Fall 2007

Please note - specifics of this schedule are subject to change.

Dates	Topic	Reading and references	Lectures	Assignments
8/30	Image formation	F&P Chapter 1	slides	Pset 0 files
9/4	Color	F&P Chapter 6 The foundations of color measurement and color perception by Brian A. Wandell (optional)	slides	
9/6 9/11 9/13 9/18	Features and texture	F&P Chapters 7, 9	slides (binary) slides (filters) slides (edges, etc.)	**Pset 0 due 9/6** Pset 1 files
9/20 9/25 9/27	Segmentation and fitting			**Pset 1 due 9/25**
10/2 10/4	Stereo			
10/9	Midterm exam			
10/11	Local invariant features			
10/16	Guest lecture			
10/18	Guest lecture			
10/23 10/25	Structure from motion			
10/30 11/1 11/6 11/8 11/13	Recognition and learning			
11/15	Optical flow			
11/20	Tracking			
11/22	Thanksgiving – no class			
11/27 11/29	Tracking, pose estimation			
12/4	Student presentations			
12/6	Wrap-up			
12/13	Final exam			



Paper list for graduate student reviews

S. Belongie, J. Malik, J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts, TPAMI 2002. pdf

D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis, TPAMI 2002. pdf

T. Cootes, G. Edwards, and C. Taylor. Active Appearance Models, TPAMI 2001. pdf (See also pdf)

P. Felzenszwalb and D. Huttenlocher. Efficient Matching of Pictorial Structures, CVPR 2000. pdf

M. Isard and A. Blake. CONDENSATION -- conditional density propagation for visual tracking, IJCV 1998. pdf

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models, IJCV 1987. pdf

T. Lindeberg. Feature Detection with Automatic Scale Selection, IJCV 1998. pdf

D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004. pdf

Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval, IJCV 2000. pdf

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust Object Recognition with Cortex-Like Mechanisms, TPAMI 2007. pdf

J. Shi and J. Malik. Normalized Cuts and Image Segmentation, TPAMI 2000. pdf

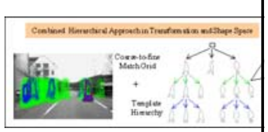
A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection, CVPR 2004. pdf

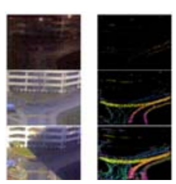
P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features, CVPR 2001. pdf

M. Weber, M. Welling and P. Perona. Unsupervised Learning of Models for Recognition, ECCV 2000. pdf

Graduate students Problem set 1 extension ideas

- Chamfer matching
  - Hierarchy of shape prototypes, search over translations
  - Comparisons with Hausdorff distance, L1 on silhouettes
  - Multi-view matching,…
- Background subtraction
  - Adaptive background model
  - Classify blobs based on shape cues
  - Collect some statistics of tracks over time,…





Texture

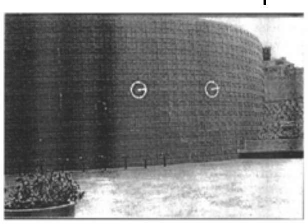
Scale: objects vs. texture

Texture problems

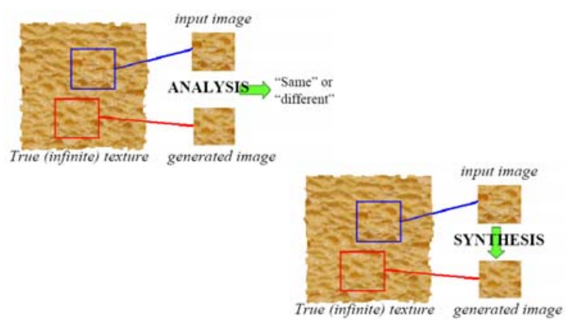
- Segmentation from texture cues
  - Analyze, represent texture
  - Group image regions with consistent texture
- Synthesis
  - Generate new texture patches/images given some examples
- Shape from texture
  - Estimate surface orientation or shape from image texture

Shape from texture

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



Analysis vs. Synthesis



Images:Bill Freeman, A. Efros

Why analyze texture?

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

Images from Malik and Perona, 1990

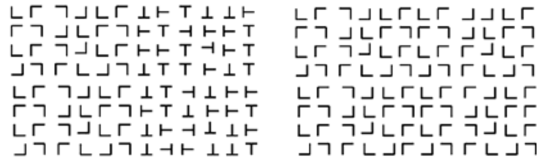
…and for this image?

Image credit: D. Forsyth

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

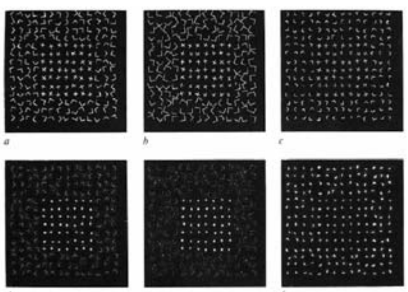
Psychophysics of texture

- Some textures distinguishable with preattentive perception [Julesz 1975]
- Analysis: need to measure “densities” of local pattern types… what are the fundamental units?



Same or different?

Capturing the local patterns with image measurements

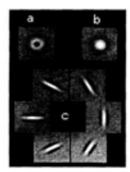


[Bergen & Adelson, *Nature* 1988]

Scale of patterns influences discriminability

Size-tuned linear filters

What filters?



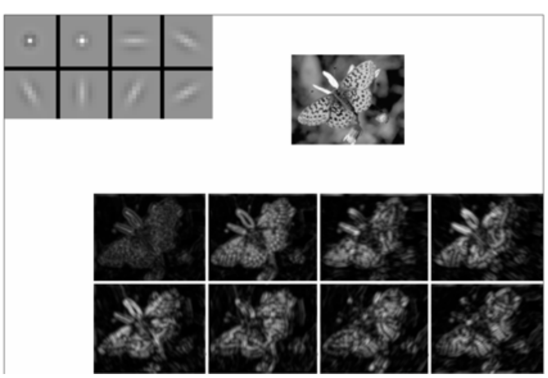
[Malik & Perona, 1990]

Weights: 1, -2, 1;
sigmas 0.62, 1,

Weights: 1,-1;
sigmas 0.71, 1.14,

Horizontal bar:
Weights: -1, 2, -1;
Sigmax = 2
Sigmay = 1
Offsets along y: 1, 0, -1

- Spots: weighted sum of two/three concentric, symmetric Gaussians
- Oriented bars: weighted sum of three oriented offset Gaussians



Forsyth & Ponce

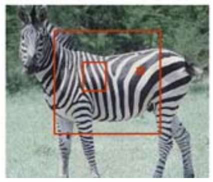
Texture representation

- Textures made up of repeated local patterns:
  - Find the patterns
    - Use filters that look like patterns (spots, bars,…)
    - Consider magnitude of response
  - Describe their statistics
    - Mean, standard deviation
    - Histograms

Texture representation

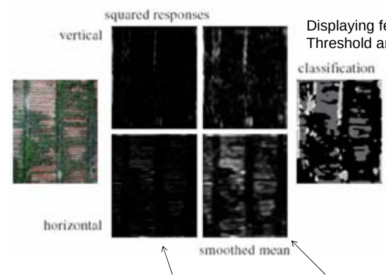
- Collect responses to collection of filters
  - Filters at multiple scales, orientations
  - Collect within window (assuming know relevant size of this window)

For example, collect mean of the squared filter outputs for a range of filters (d filters -> d dimensional vector for each window).



Texture representation: example

Displaying features: Threshold and combine



Lighter = stronger response

Smooth to get mean of squared response in some window

Forsyth & Ponce

Texture vocabularies

- *Textons*: 2D units of preattentive textures [Julesz, 1981]
- *Textons*: prototypical responses of images to a given filter bank [Leung & Malik, 1999]

Recognizing materials with textons

[Leung & Malik, 1999]

- Collect filter responses from sample of images (possibly over multiple viewing conditions)
- Vector quantize into textons
- Describe new images in terms of distribution of textons
- Compare histograms, e.g. chi-squared distance

$$\chi^2(h_1, h_2) = \frac{1}{2}\sum_{n=1}^{\#bins}\frac{(h_1(n) - h_2(n))^2}{h_1(n) + h_2(n)}$$

Related recent research:
[Varma and Zisserman, 2002]
[Lazebnik, Schmid, and Ponce, 2003]
[Hayman et al., 2004]

	Felt	Terrycloth	Rough Plastic	Leather	Sandpaper	Pebbles	Plaster–a	Plaster–b	Rough Paper	Artificial Grass	Rough Shingle	Aluminum Foil	Cork	Rough Tile
Felt	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Terrycloth	0.0	1.0	0.0	0.0	0.3	0.0	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Rough Plastic	0.0	0.0	0.9	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Leather	0.2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sandpaper	0.0	0.1	0.0	0.0	1.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pebbles	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Plaster–a	0.0	0.1	0.2	0.0	0.1	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
Plaster–b	0.0	0.2	0.1	0.0	0.0	0.0	0.8	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Rough Paper	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0
Artificial Grass	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.1	0.1	0.0	0.0
Roof Shingle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	1.0	0.1	0.0	0.0
Aluminum Foil	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	1.0	0.0	0.0
Cork	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.2
Rough Tile	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9

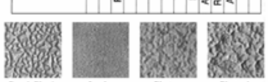
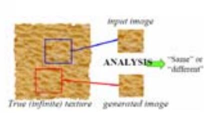


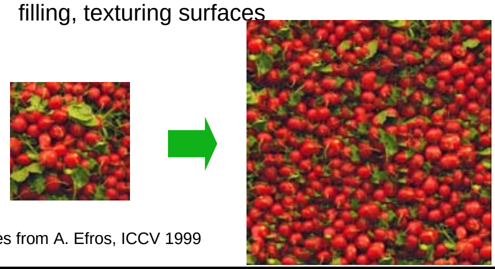
Figure 9. Similarity matrix for 14 materials. Each entry c_{ij} is given by the chi-square probability function (Eq. (2)) that samples of material j will be classified as material i. As shown in this figure, for example, "Leather" and "Rough Plastic" are likely to be classified correctly, while "Plaster-a" and "Plaster-b" are likely to be mistaken between them. Sample images from these four materials are shown as well.



[Leung & Malik, 1999]

Texture synthesis

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



Slides from A. Efros, ICCV 1999

The Challenge

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

Motivation from Language

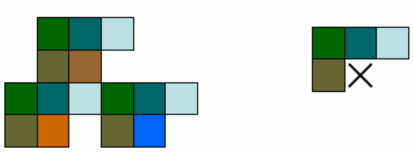
- [Shannon,'48] proposed a way to generate English-looking text using N-grams:
  - Assume a generalized Markov model
  - Use a large text to compute probability distributions of each letter given N-1 previous letters
  - Starting from a seed repeatedly sample this Markov chain to generate new letters
  - One can use whole words instead of letters too:

WE NEED TO EAT CAKE

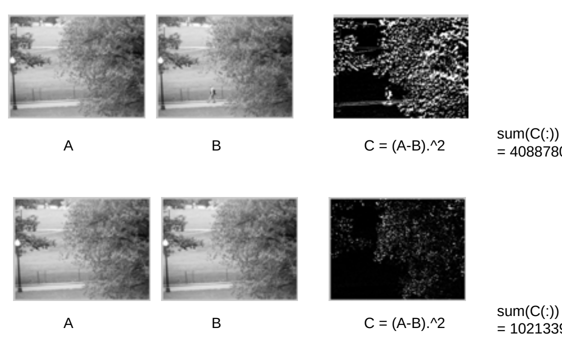
Motivation from language

- 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"*
- Notice how well local structure is preserved!
  - Now let's try this in 2D...

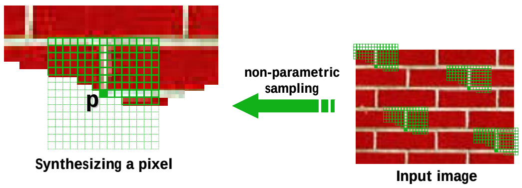
Dewdney, "A potpourri of programmed prose and prosody" *Scientific American,* 1989.



SSD: simple measure of patch similarity



Efros & Leung algorithm



- Assuming Markov property, compute P(**p**|N(**p**))
  - Building explicit probability tables infeasible
  - Instead, we *search the input image* for all similar neighborhoods — that's our pdf for **p**

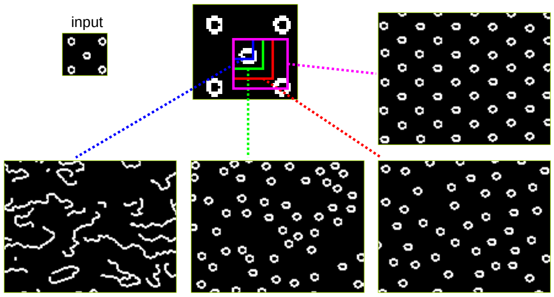
$$d = d_{SSD} * G$$

  - To sample from this pdf, just pick one match at random

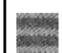
Efros & Leung algorithm

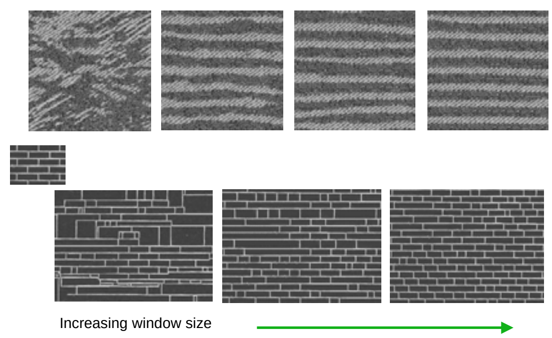
- Growing is in "onion skin" order
  - Within each "layer", pixels with most neighbors are synthesized first
  - If no close match can be found, the pixel is not synthesized until the end
- Using *Gaussian-weighted* SSD is very important
  - to make sure the new pixel agrees with its closest neighbors
  - Approximates reduction to a smaller neighborhood window if data is too sparse

Neighborhood Window



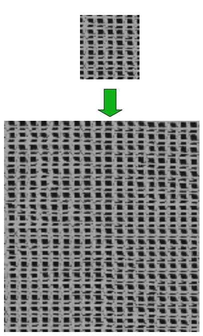
Varying Window Size



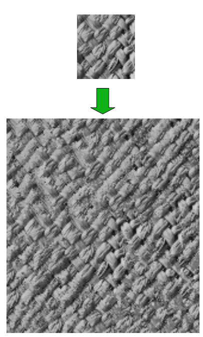


Synthesis results

french canvas

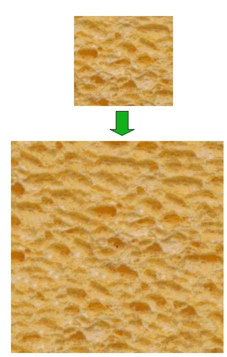


rafia weave

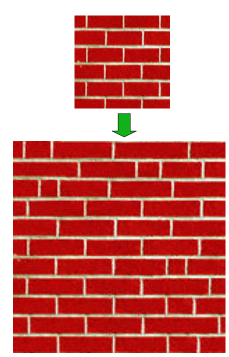


Synthesis results

white bread

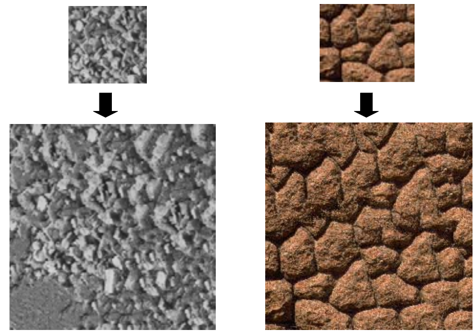


brick wall



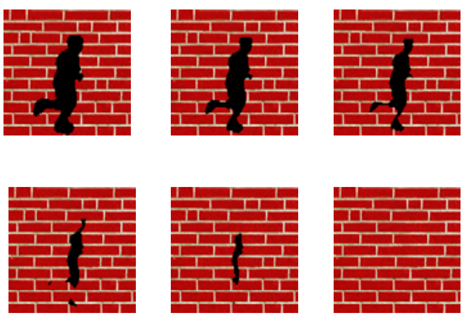
Synthesis results

Failure Cases

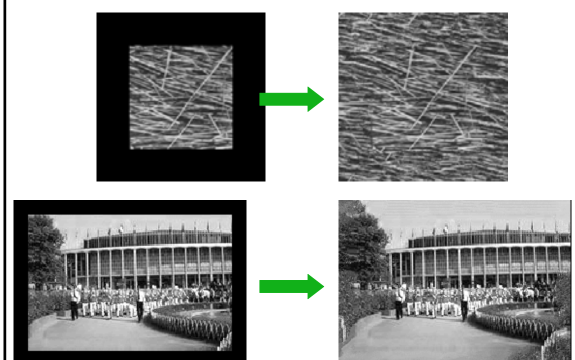


Growing garbage **Verbatim copying**

Hole Filling



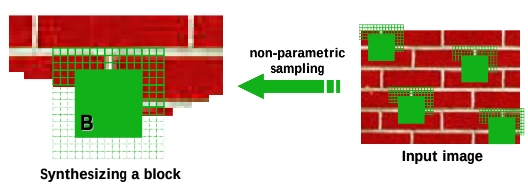
Extrapolation



Summary

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

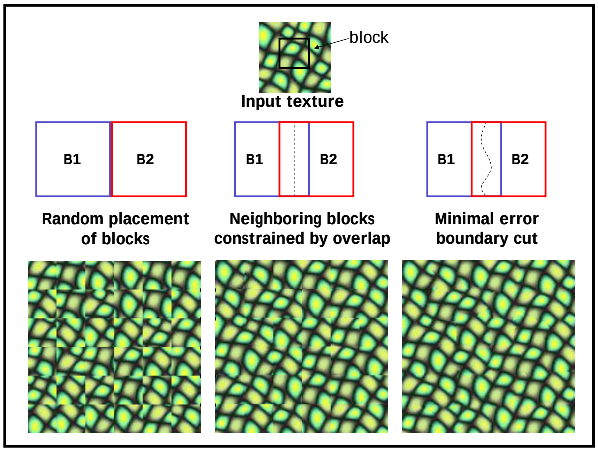
Image Quilting [Efros & Freeman 2001]

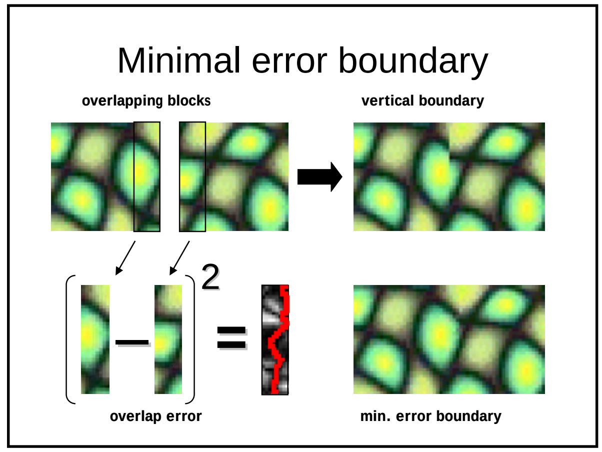


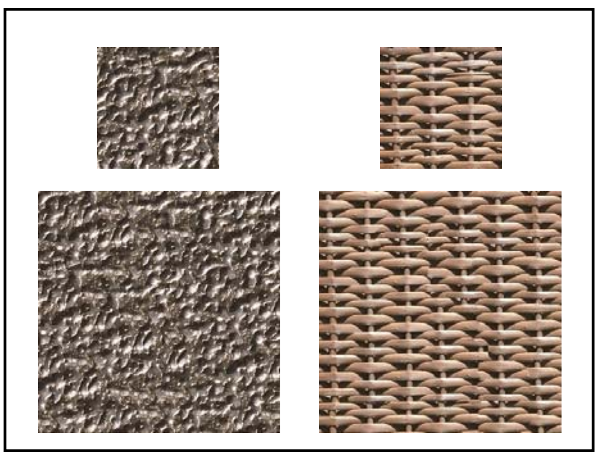
- <u>Observation:</u> neighbor pixels are highly correlated

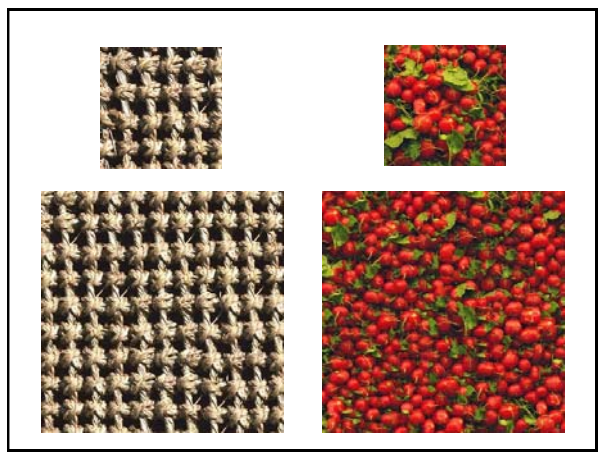
<u>Idea:</u> unit of synthesis = block

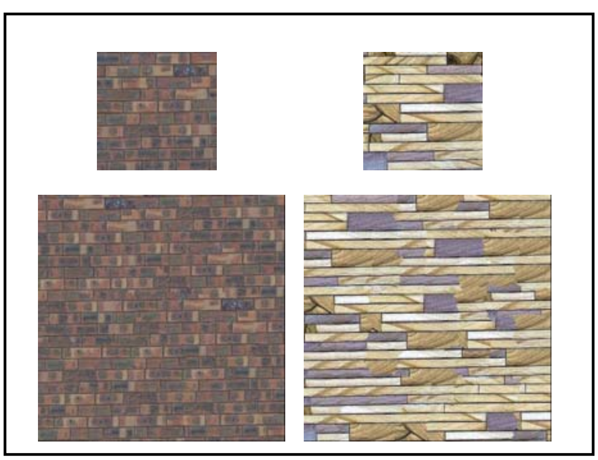
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

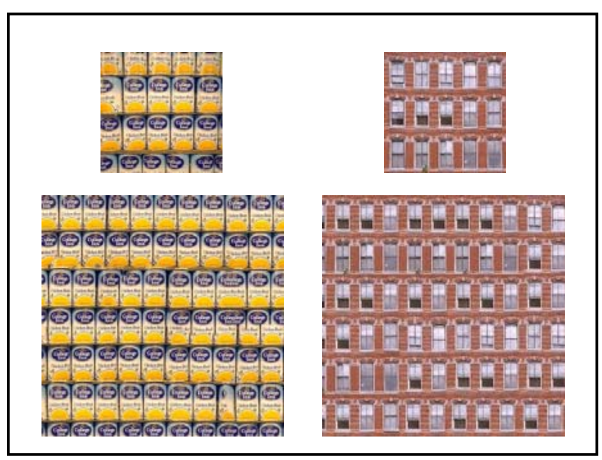

block
Input texture
B1
B2
Random placement of blocks
B1
B2
Neighboring blocks constrained by overlap
B1
B2
Minimal error boundary cut

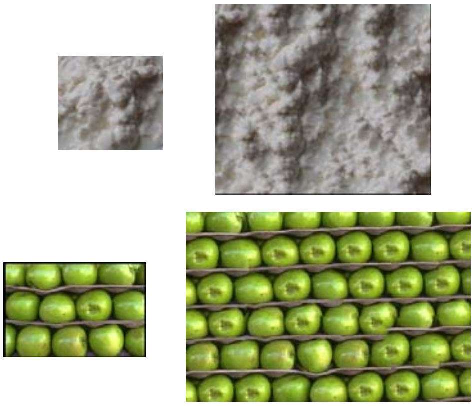

Minimal error boundary
overlapping blocks
vertical boundary
2
overlap error
min. error boundary

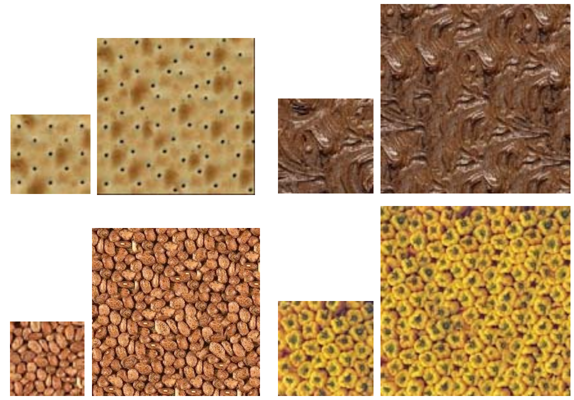








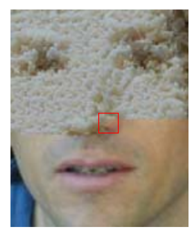




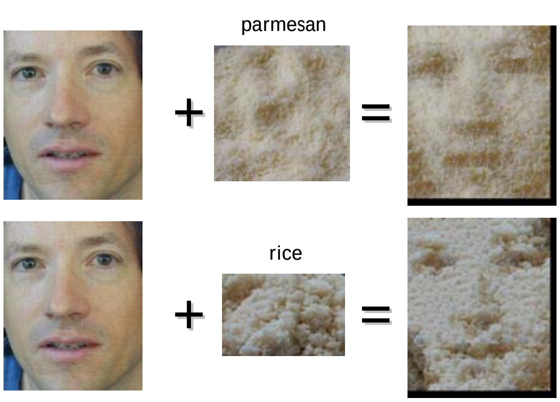
Failures
(Chernobyl Harvest)

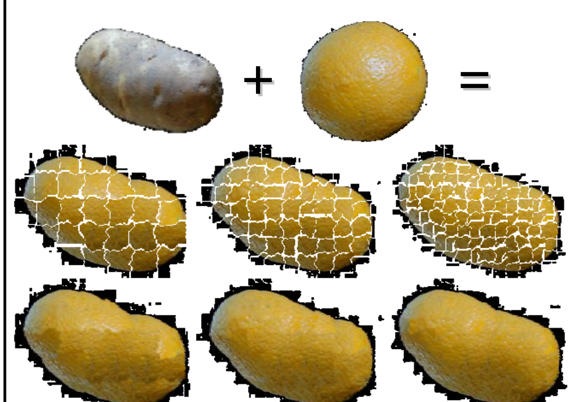
Texture Transfer

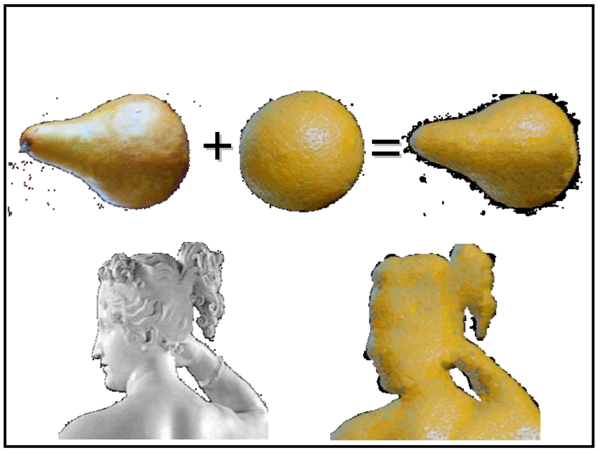
- Take the texture from one object and “paint” it onto another object
  – This requires separating texture and shape
  – That’s HARD, but we can cheat
  – Assume we can capture shape by boundary and rough shading



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







Coming up

- Problem set 1 due Tuesday
- Segmentation: read Chapter 14