CMPSCI 370HH: Intro. to Computer Vision

Texture synthesis

University of Massachusetts, Amherst
April 19, 2016

Instructor: Subhransu Maji

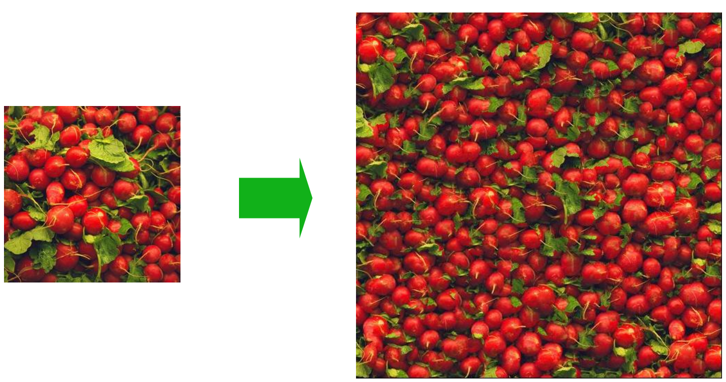
Slides credit: Kristen Grauman and others

Next class

- Presentation guidelines
  - 20 mins for each team (random order)
  - 15 mins presentation + 5 mins for questions
- Clearly describe
  - Problem statement
  - Preliminary results
  - What are you going to do the next week (write-up)
- 4-6 page writeup (May 6)
  - No deadline extension

Texture synthesis

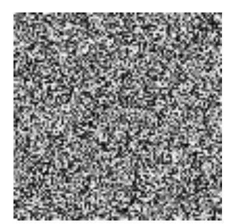
- **Goal:** create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces



The challenge

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

repeated



stochastic

Both?

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**:

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

- The above is actually a *first-order* Markov chain
- An *N'th-order* Markov chain:

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

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

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

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 $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-(n-1)})$
2. Given words $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k-1}$
   - compute $\mathbf{x}_k$ by sampling from $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-(n-1)})$

WE NEED TO EAT CAKE

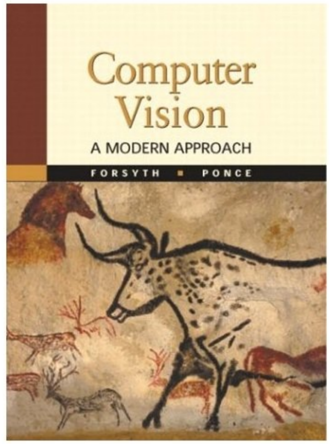
Text synthesis

- "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.

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/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*:

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

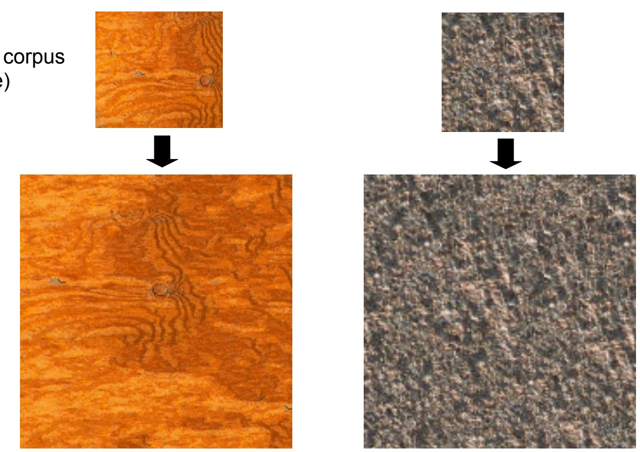
	A	
D	*X*	*B*
	C	

Texture synthesis

Can apply 2D version of text synthesis

Texture corpus (sample)

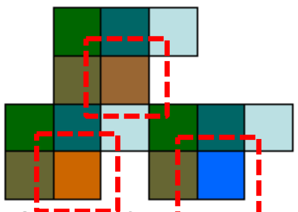
Output



[Efros & Leung, ICCV 99](#)

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.

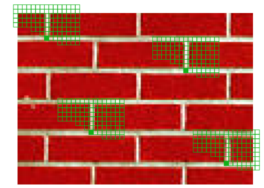


Sample of the texture
("corpus")

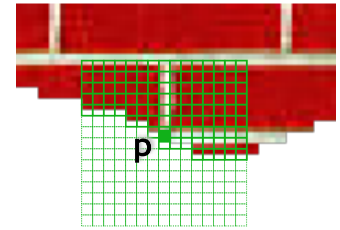
Place we want to insert next

Distribution of a value of a pixel is conditioned on its neighbors alone.

Synthesizing one pixel



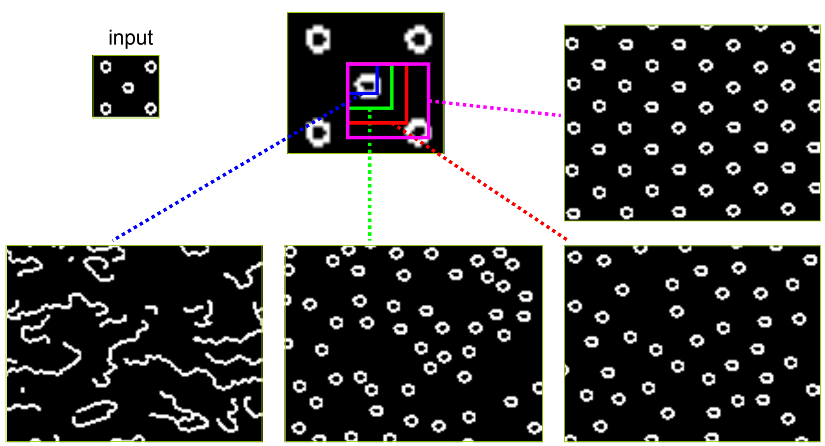
input image



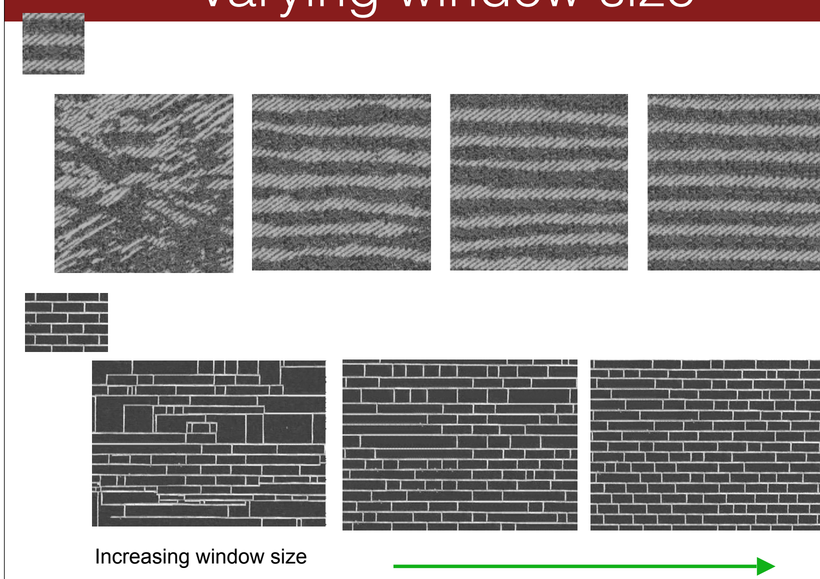
synthesized image

- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- 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

Neighborhood window



Varying window size



Increasing window size

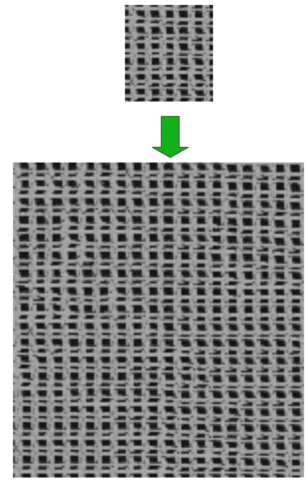
Growing Texture

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

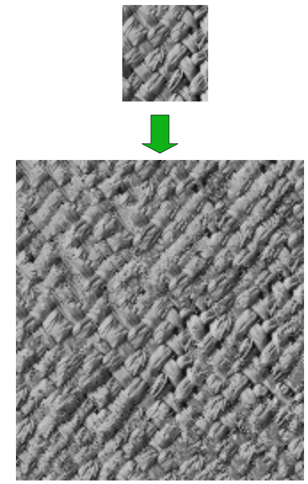
Slide from Alyosha Efros, ICCV 1999 17

Synthesis results

french canvas



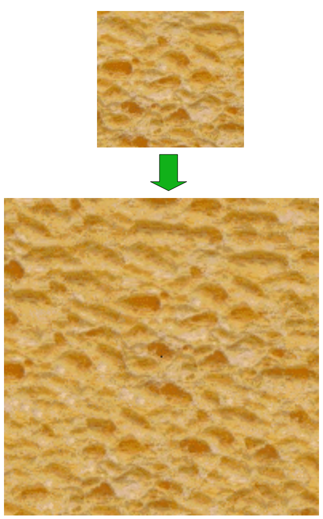
rafia weave



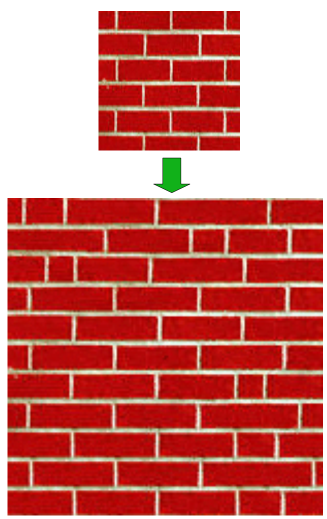
Slide from Alyosha Efros, ICCV 1999

Synthesis results

white bread

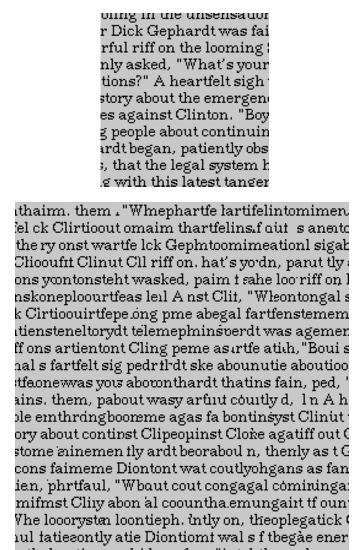


brick wall



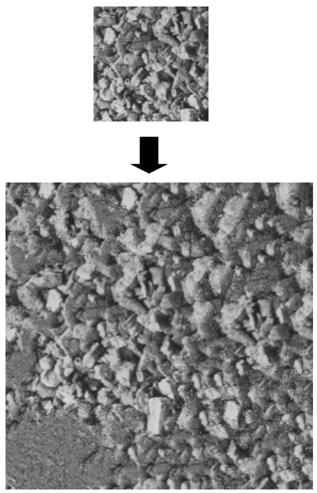
Slide from Alyosha Efros, ICCV 1999

Synthesis results

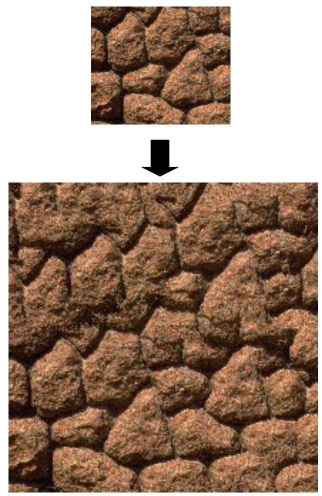


Slide from Alyosha Efros, ICCV 1999

Failure Cases

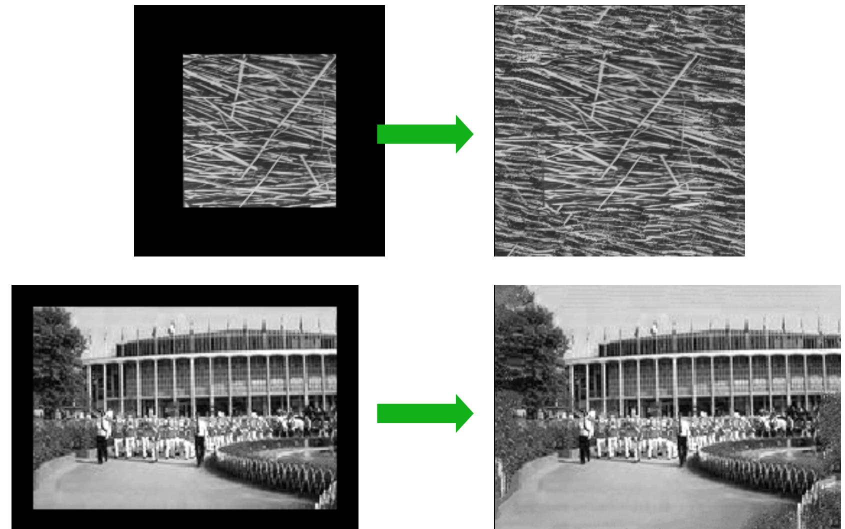


Growing garbage



Verbatim copying

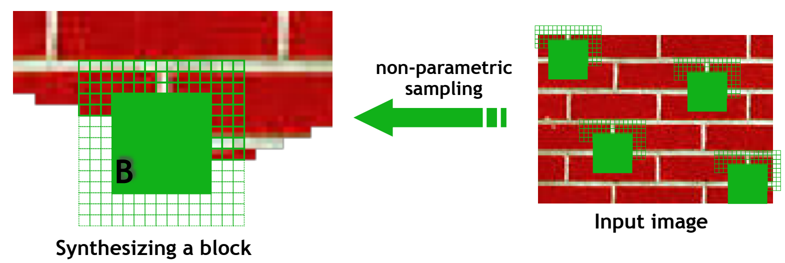
Extrapolation



Texture synthesis

- The Efros & Leung algorithm
  - Simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - … but can be very slow
    - [n m] image synthesis from [p q] image requires nxmxpxq patch comparisons

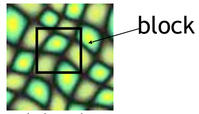
Image Quilting [Efros & Freeman 2001]



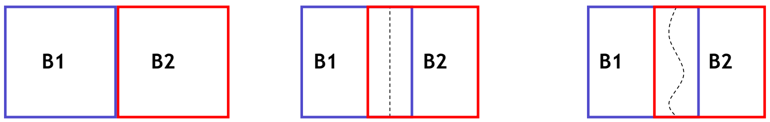
- <u>Observation:</u> neighbor pixels are highly correlated

<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



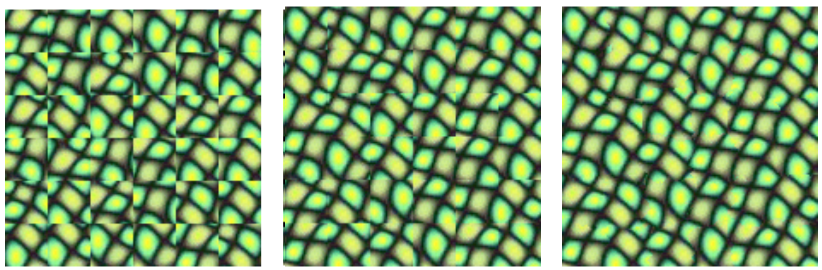
Input texture



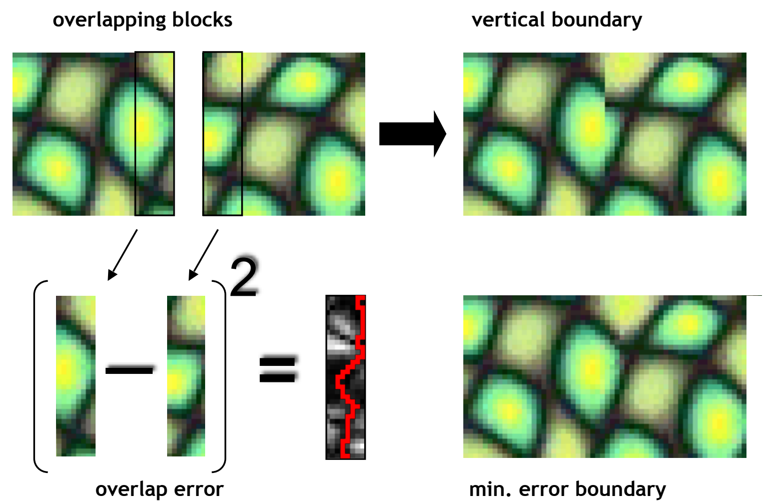
Random placement of blocks

Neighboring blocks constrained by overlap

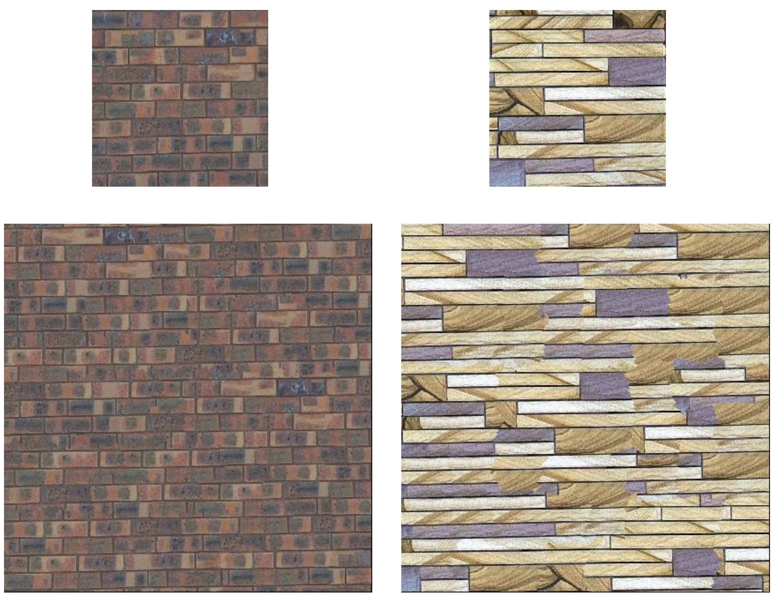
Minimal error boundary cut

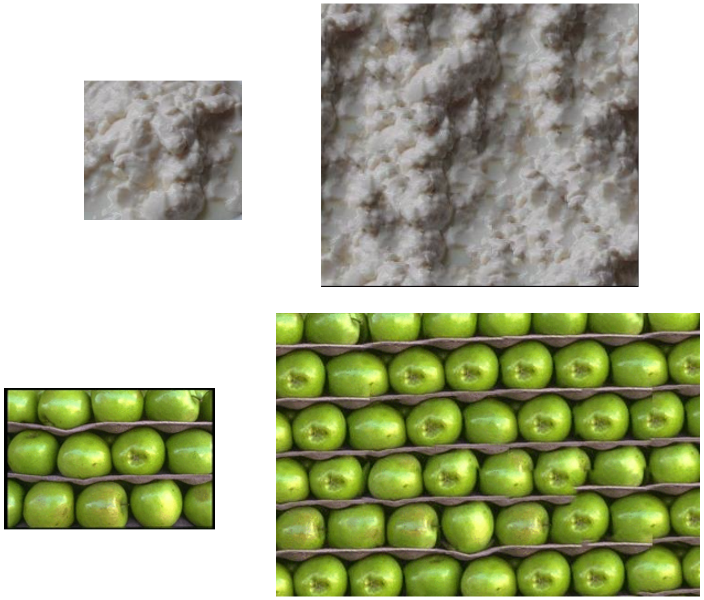


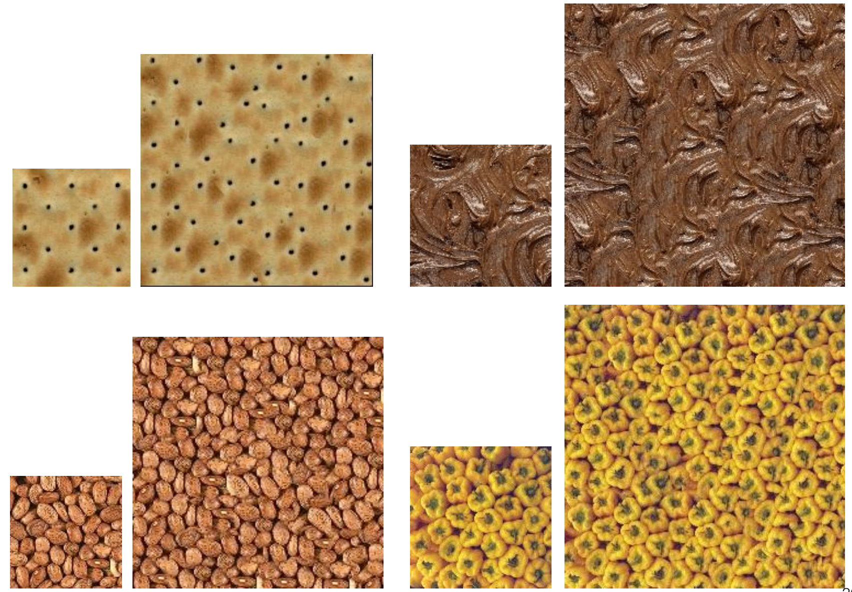
Minimal error boundary



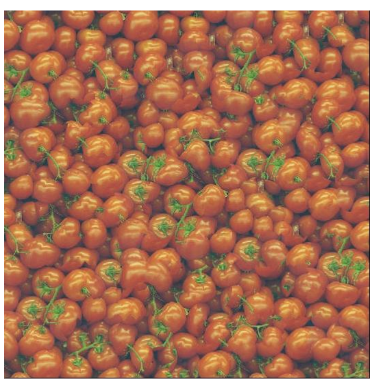
Slide from Alyosha Efros







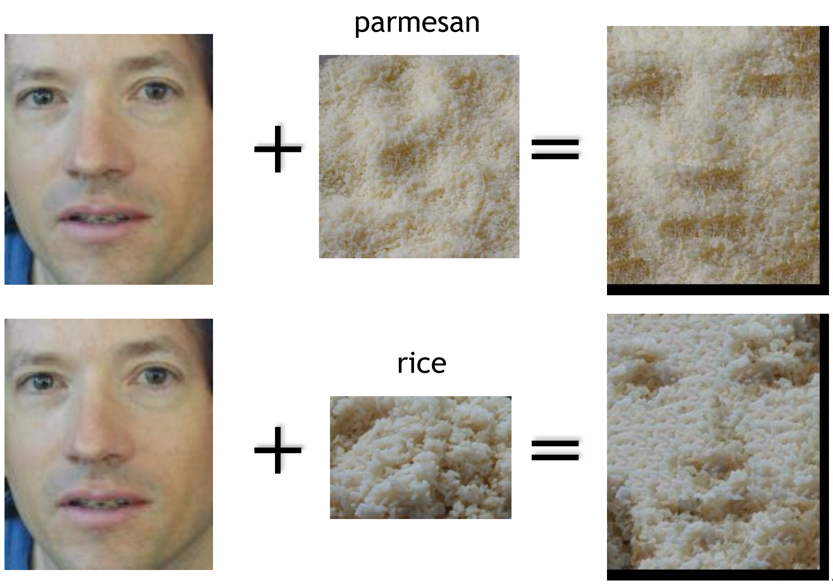
Failures
(Chernobyl Harvest)

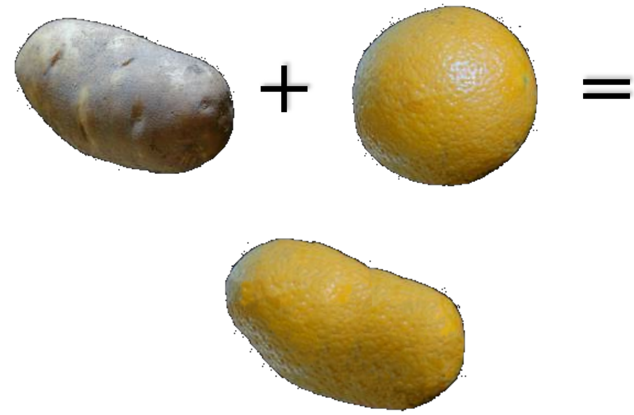


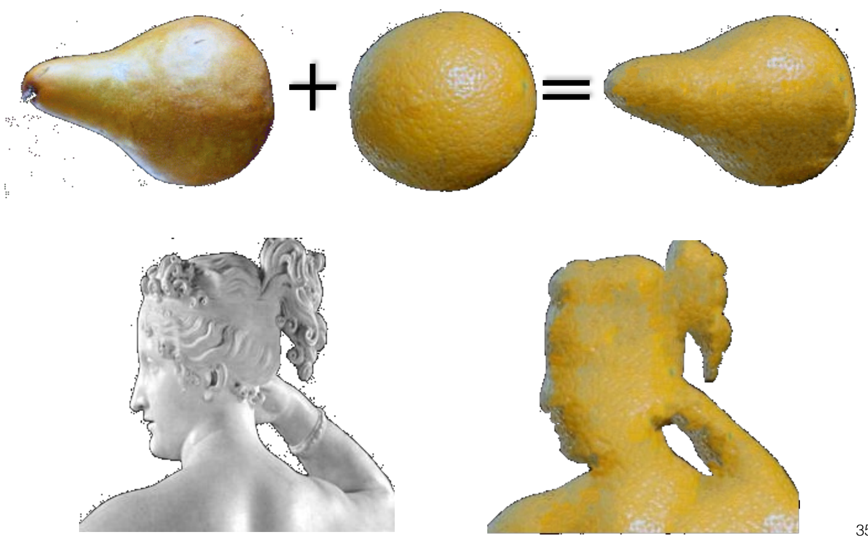
Texture transfer

- 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



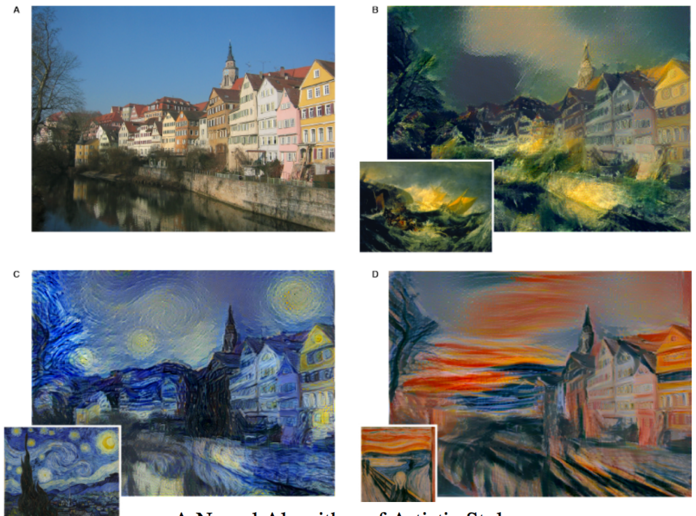




(Manual) texture synthesis in the media

http://www.dailykos.com/story/2004/10/27/22442/878

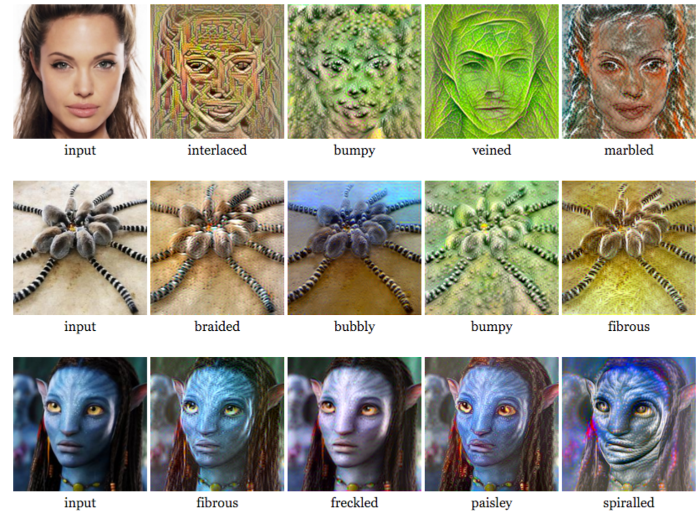
Style transfer using CNNs



A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

Style transfer with texture attributes



http://vis-www.cs.umass.edu/texture/

Tsung-Yu Lin, Subhransu Maji, CVPR 16