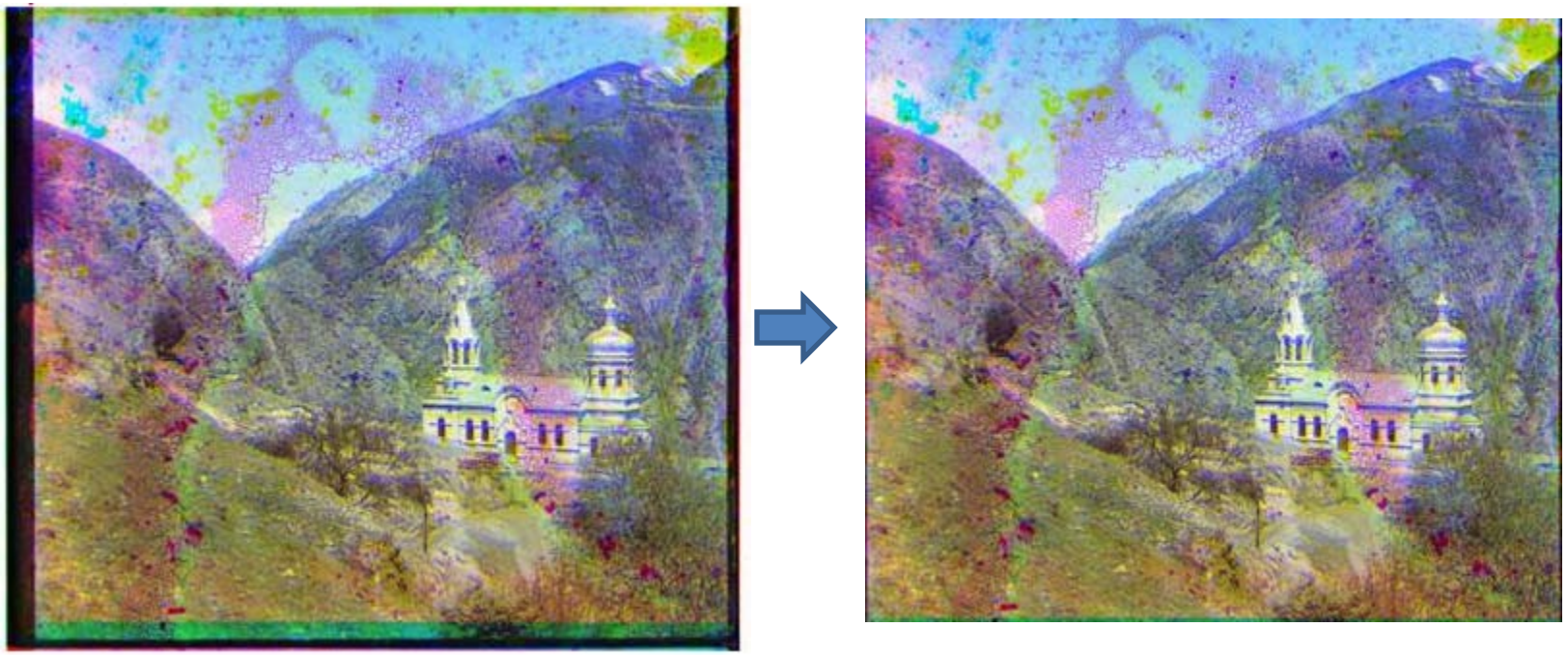
Texture Synthesis and Hole-Filling

Computational Photography

Derek Hoiem, University of Illinois

Project 2

Cropping



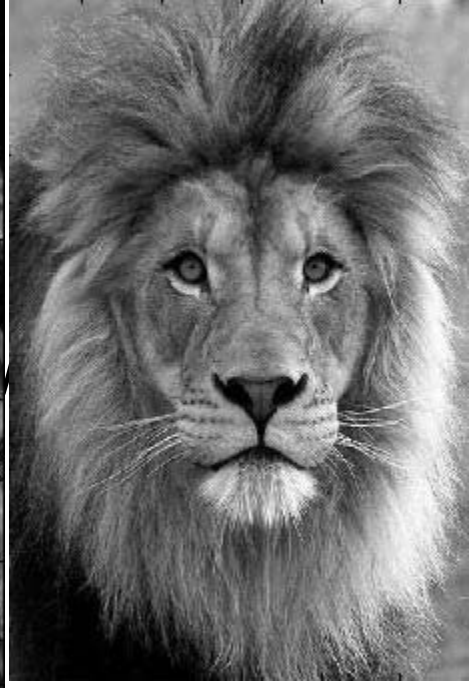
Project 1

- Class favorite projects
  - Jia-Bin Huang
  - Charles Park

Project 1

Class favorite results

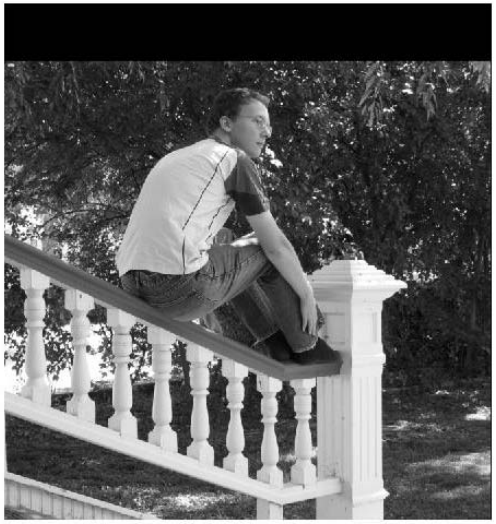
- “Underwear woman(lion???) t



Project 1

Class favorite results

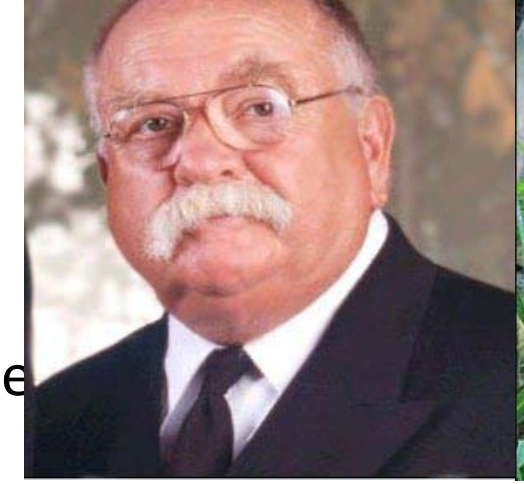
- “Who’s sitting here?” by C

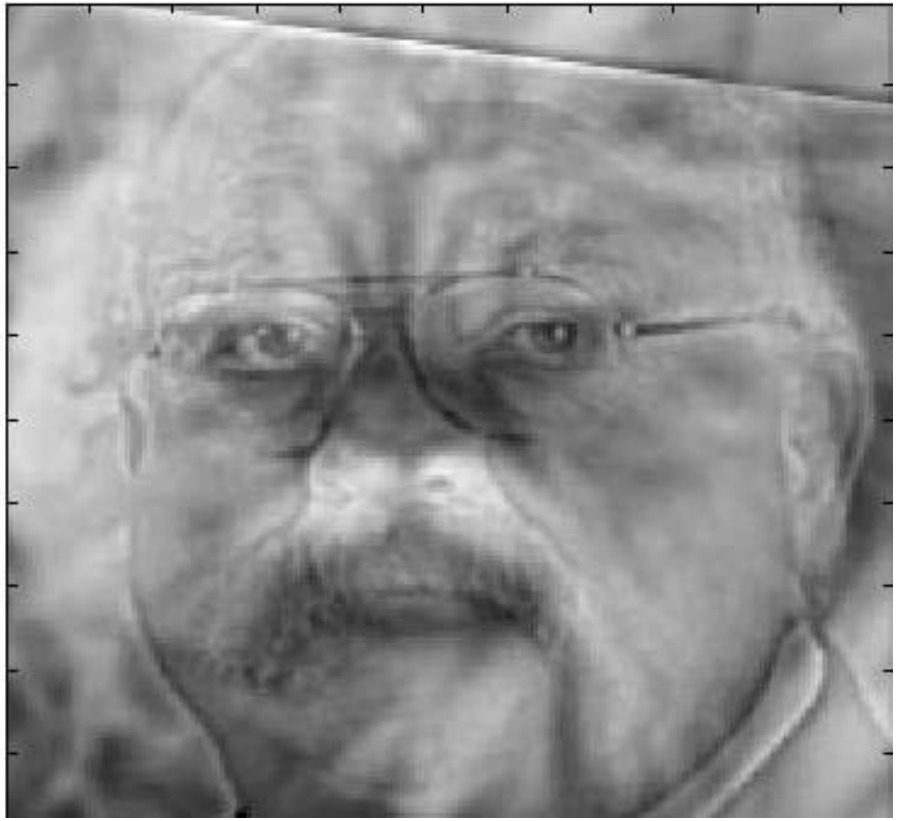


Project 1

Class favorite results

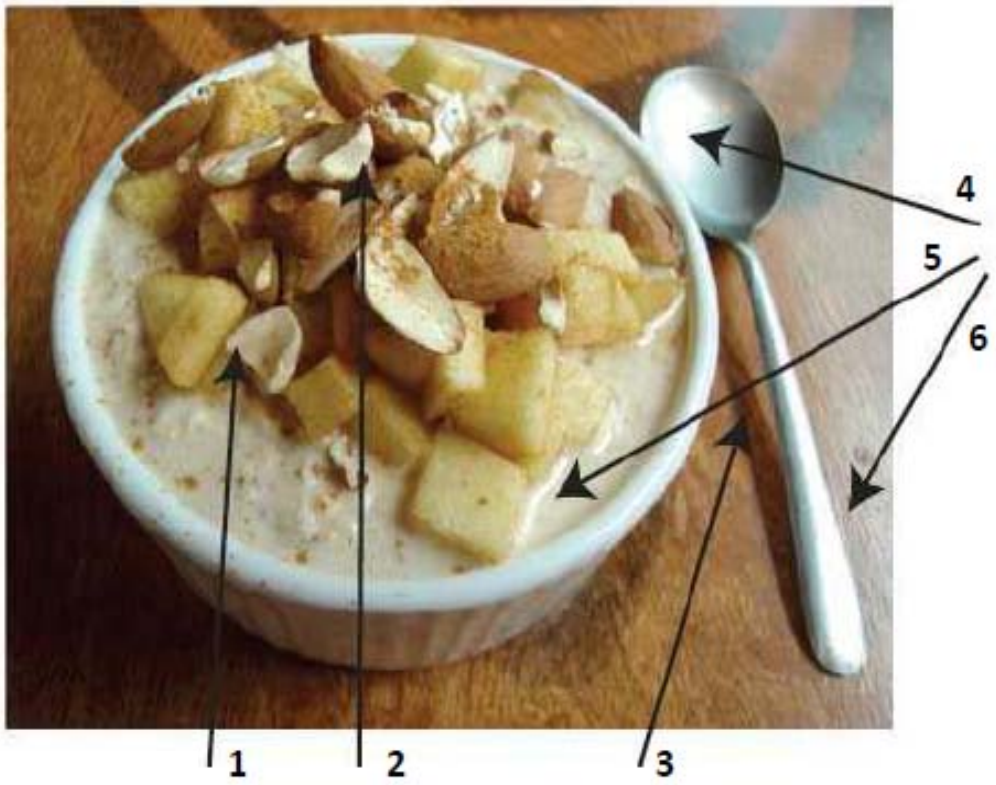
- Brimeley and Cat (Set #1) by Ste





Take-home questions

1.



A. For each of the arrows in the above image, name the reasons the pixel near the end of the arrow has its brightness value and explain very briefly. The arrow pointing to milk is pointing to the thin bright line at the edge of the piece of apple; the arrow pointing to the spoon handle is pointing to the bright area on the handle.

Possible factors: albedo, shadows, texture, specularities, curvature, lighting direction

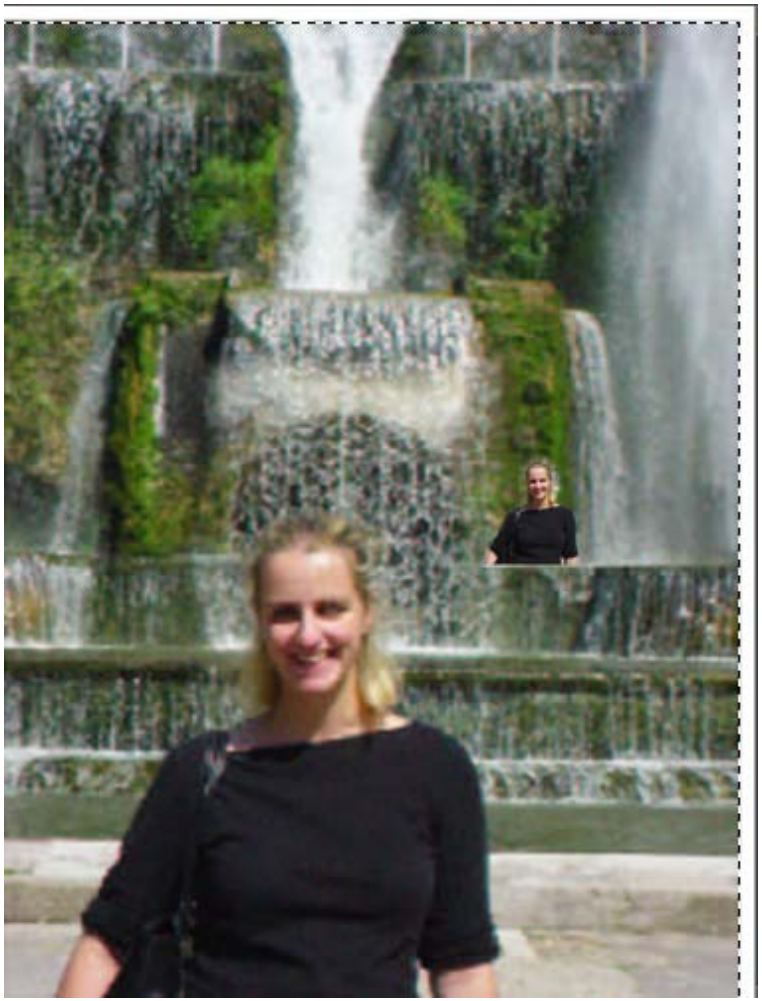
Slides: http://www.cs.illinois.edu/~dhoiem/courses/vision_spring10/lectures/LightandColor.pdf

Take-home questions

2. What would be the result in “Intelligent Scissors” if all of the edge costs were set to 1?

3. Typically, “GrabCut” will not work well on objects with thin structures. How could you change the boundary costs to better segment such objects?

Last class: Cut someone out to make more of them



But what if we want less of somebody?

Today's Class

- Texture synthesis and hole-filling

Texture

- Texture depicts spatially repeating patterns
- Textures appear naturally and frequently

radishes

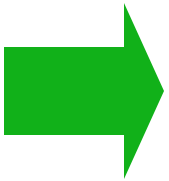
rocks

yogurt

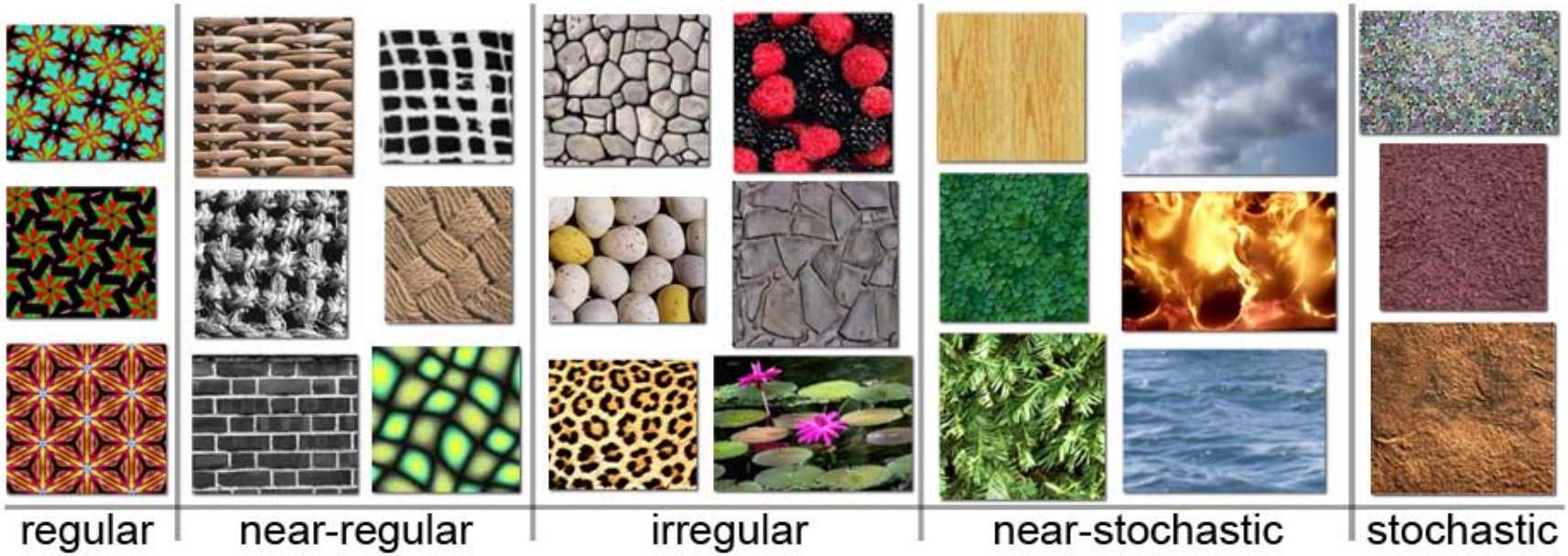
Many slides from James Hays

Texture Synthesis

- Goal of Texture Synthesis: 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

One idea: Build Probability Distributions

Basic idea

1. Compute statistics of input texture (e.g., histogram of edge filter responses)
2. Generate a new texture that keeps those same statistics

- D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95.*
- E. P. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *ICIP 1998.*

One idea: Build Probability Distributions

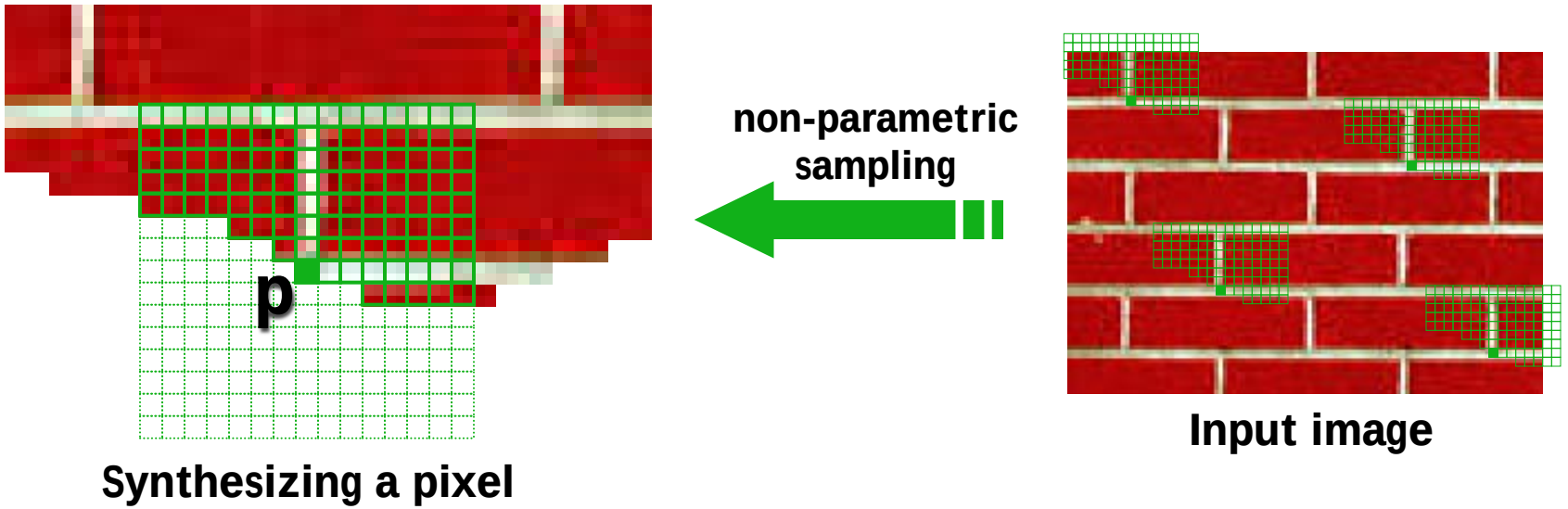
But it (usually) doesn't work

- Probability distributions are hard to model well

Input

Synthesized

Another idea: Sample from the image



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

Efros and Leung 1999 SIGGRAPH

Details

- How to match patches?
  - Gaussian-weighted SSD (more emphasis on nearby pixels)
- What order to fill in new pixels?
  - “Onion skin” order: pixels with most neighbors are synthesized first
  - To synthesize from scratch, start with a randomly selected small patch from the source texture

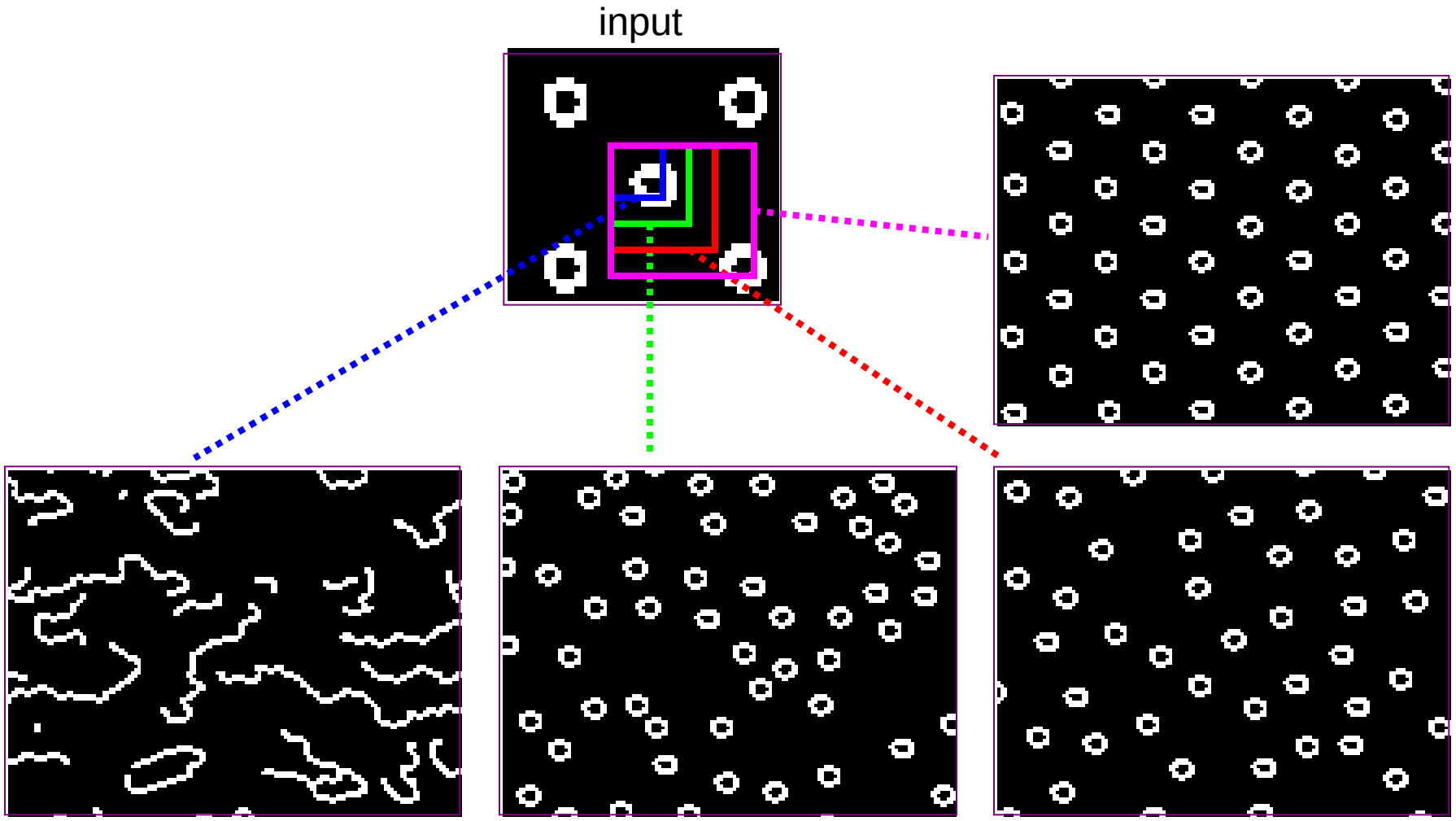
Idea from Shannon (Information Theory)

- Generate English-sounding sentences by modeling the probability of each word given the previous words (n-grams)

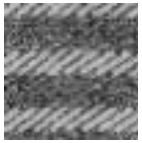
- Large “n” will give more structured sentences

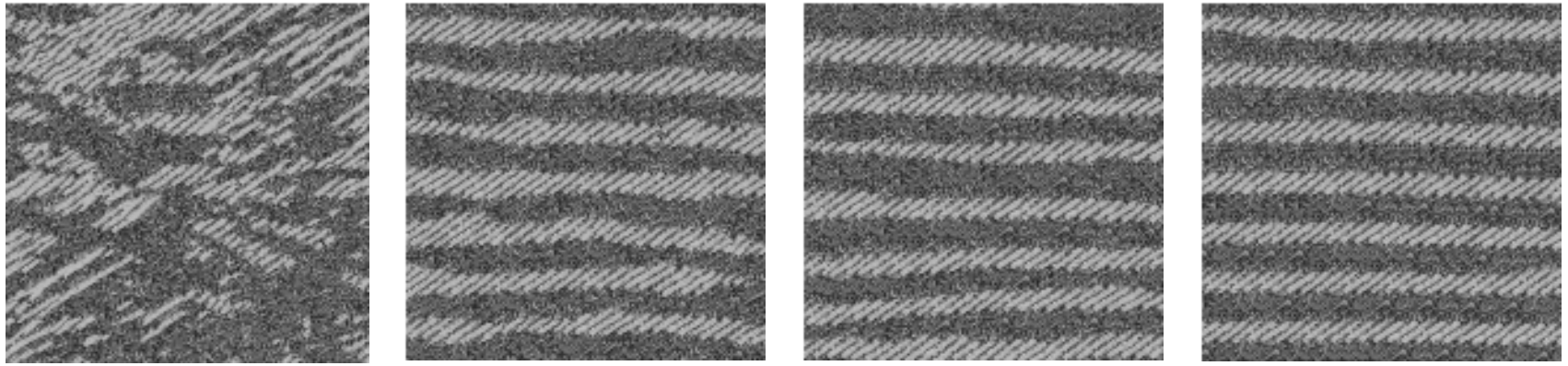
“I spent an interesting evening recently with a grain of salt.”

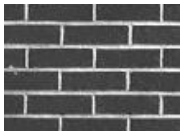
Size of Neighborhood Window

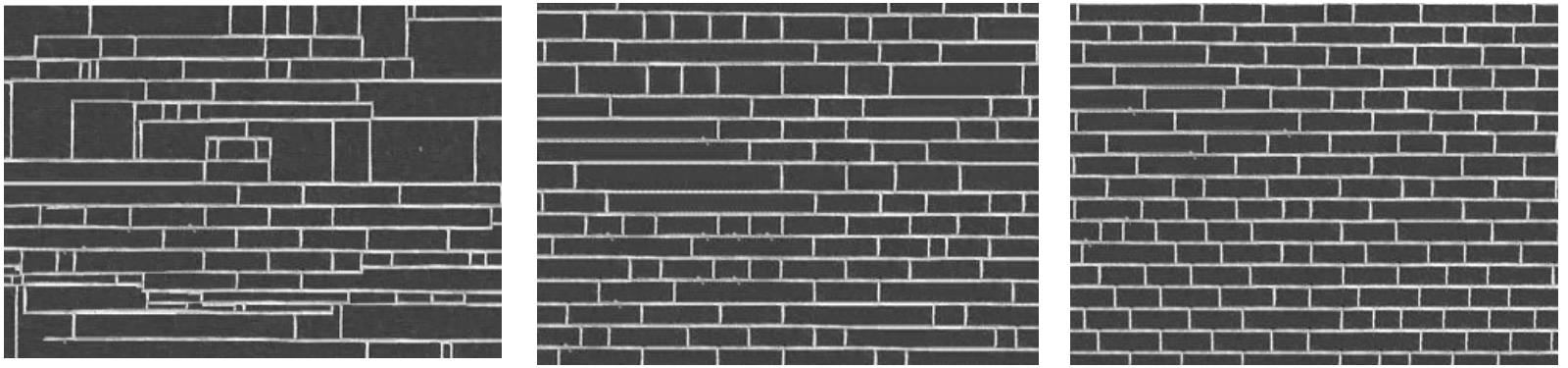


Varying Window Size





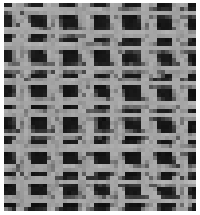


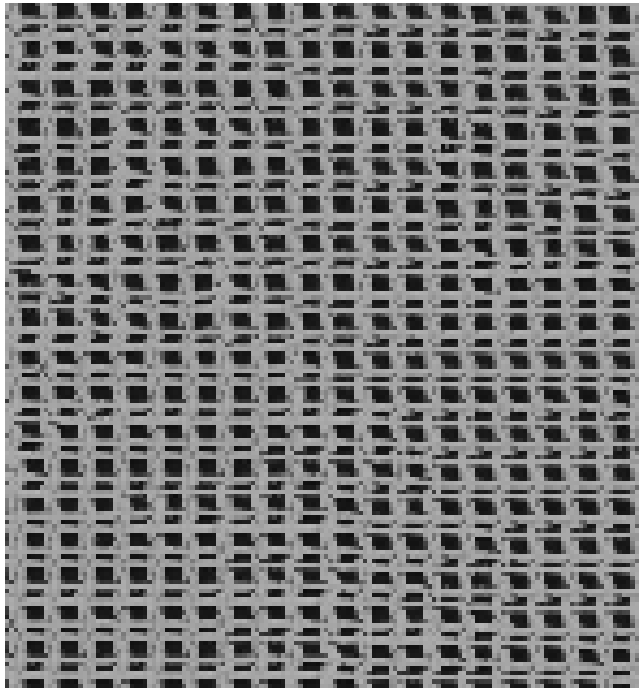


Increasing window size

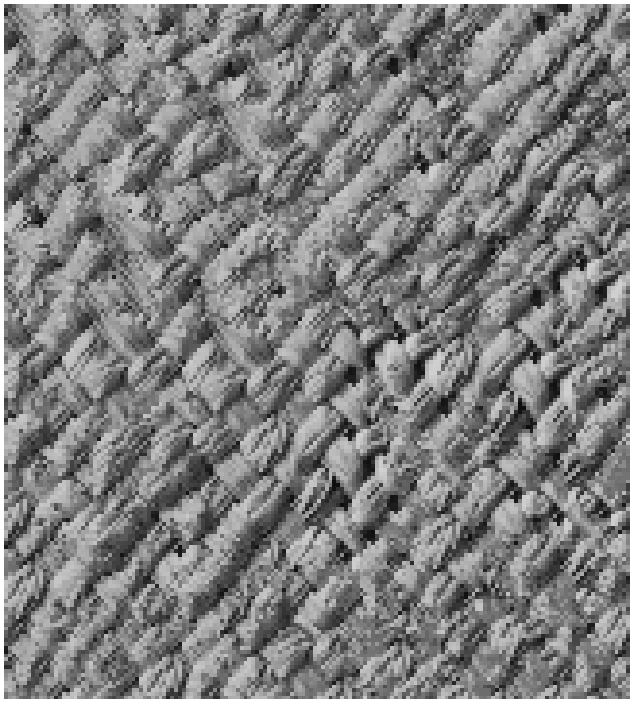
Synthesis Results

french canvas



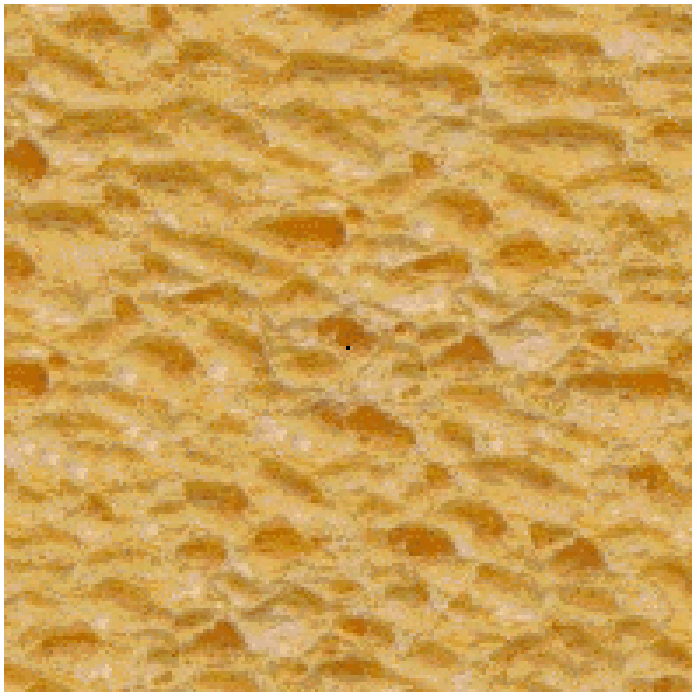


rafia weave

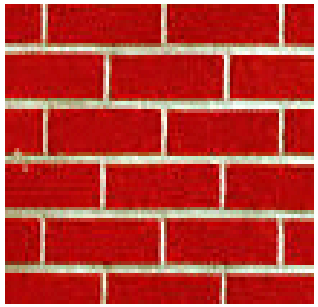


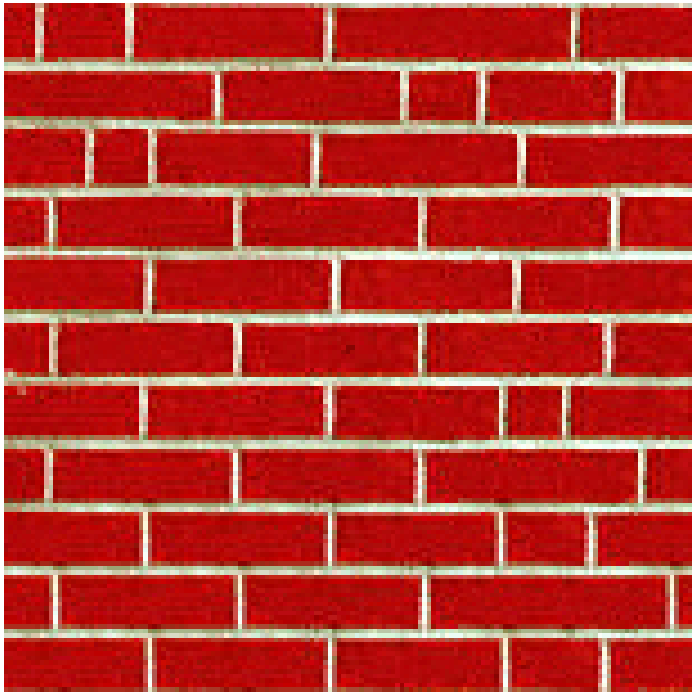
More Results

white bread

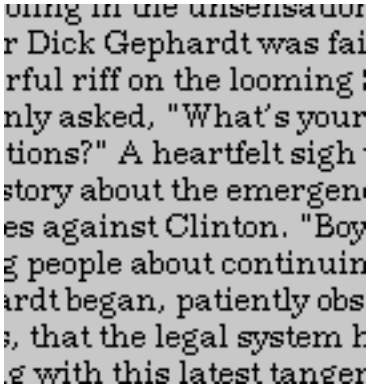


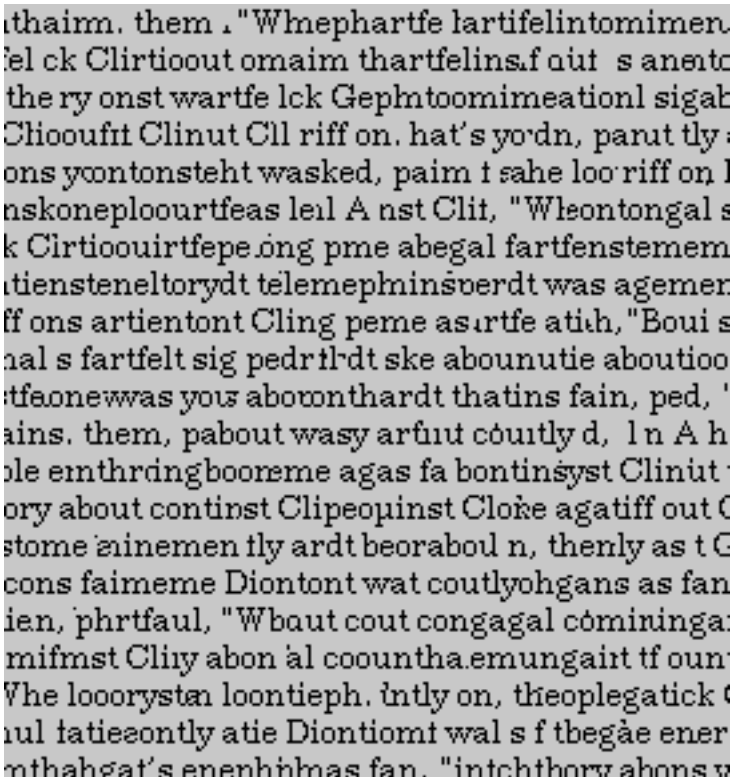
brick wall



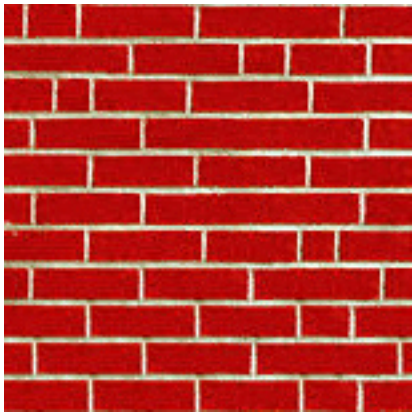


Homage to Shannon

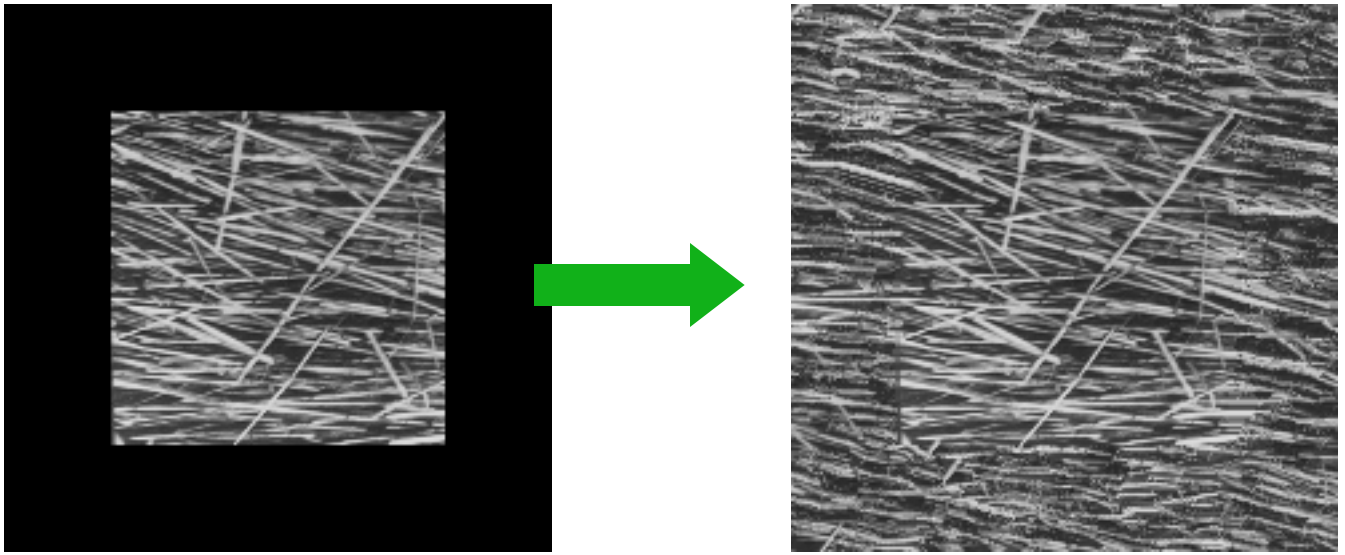


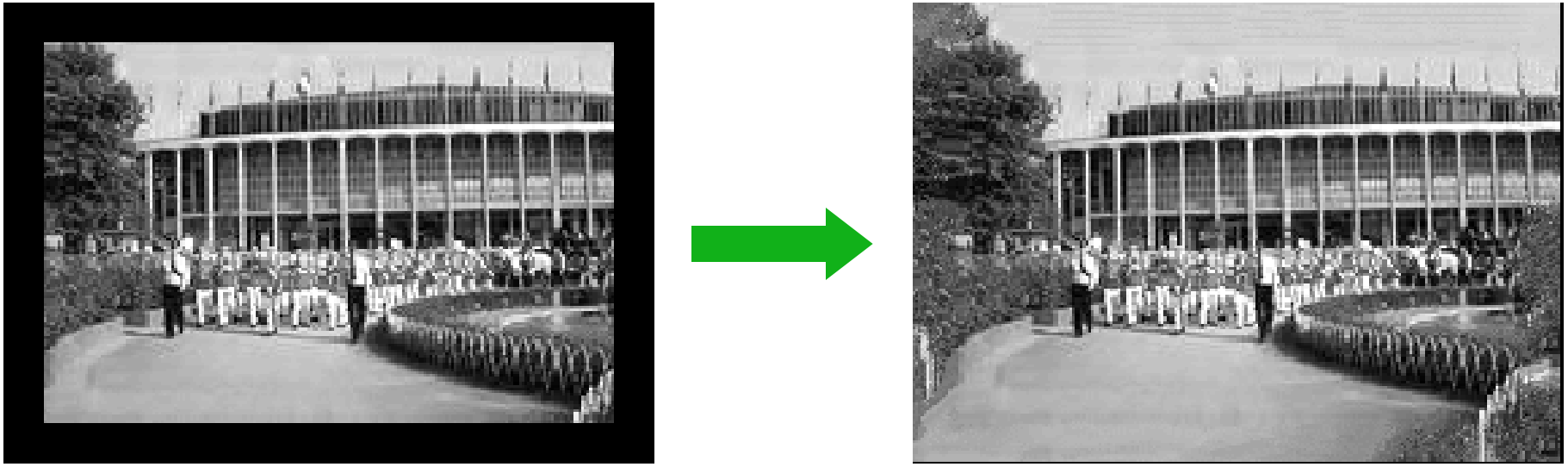


Hole Filling



Extrapolation

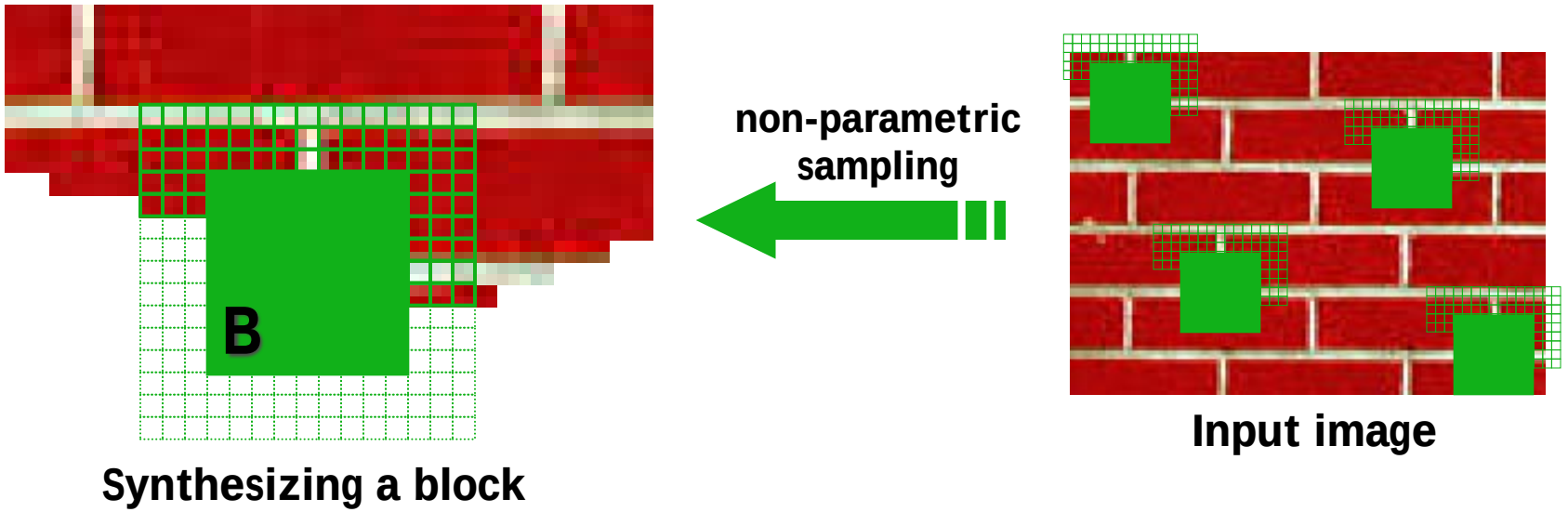




Summary

- The Efros & Leung texture synthesis algorithm
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow

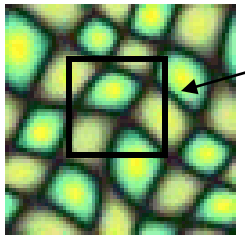
Image Quilting [Efros & Freeman 2001]



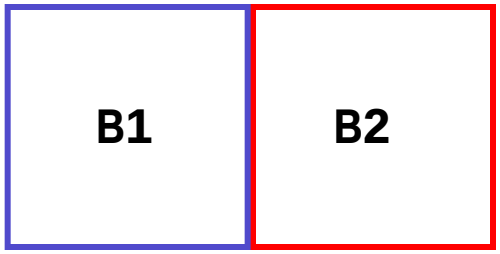
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

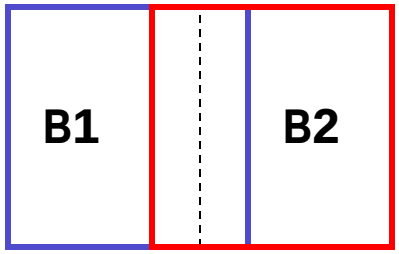
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once



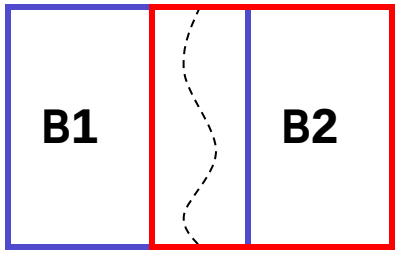
Input texture



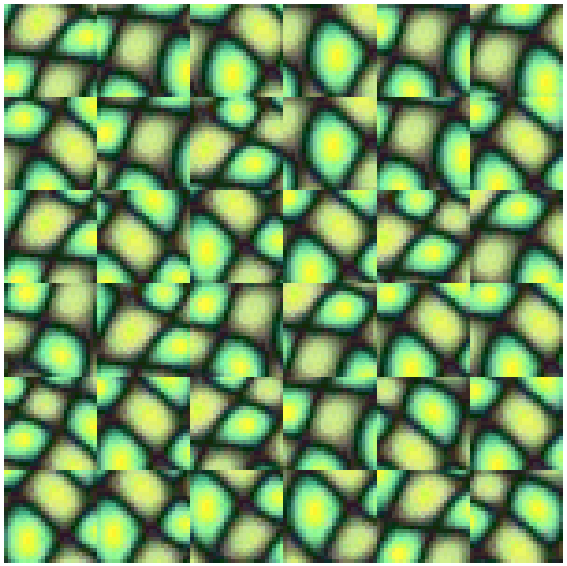
Random placement of blocks

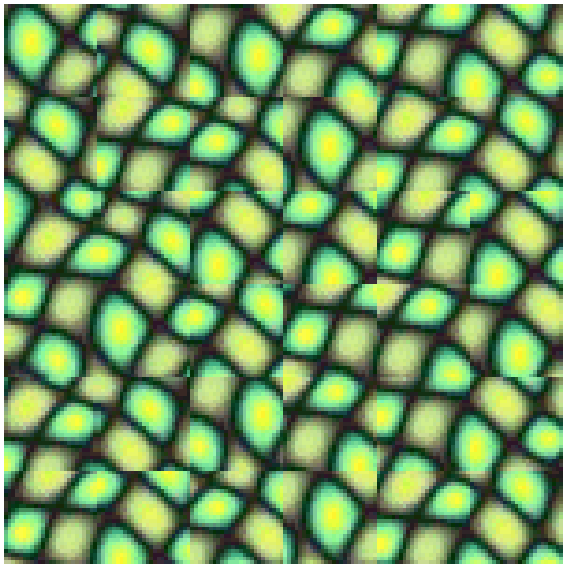


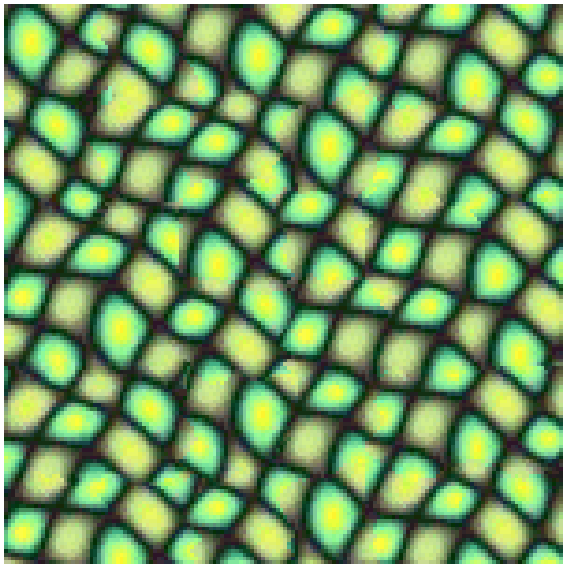
Neighboring blocks constrained by overlap



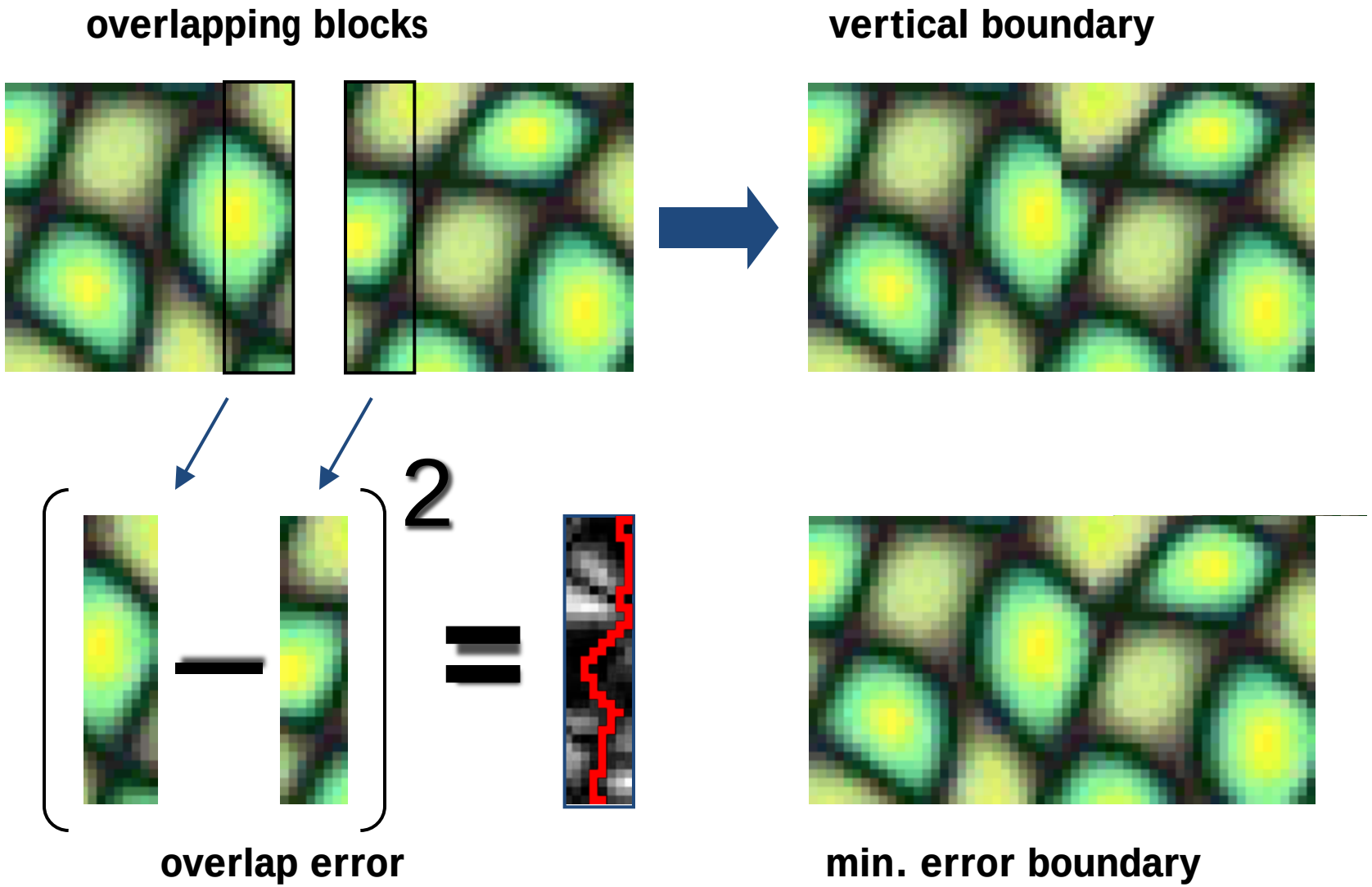
Minimal error boundary cut

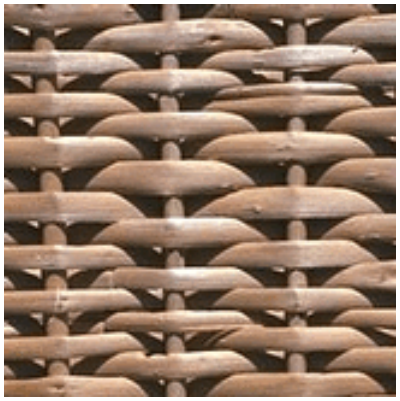


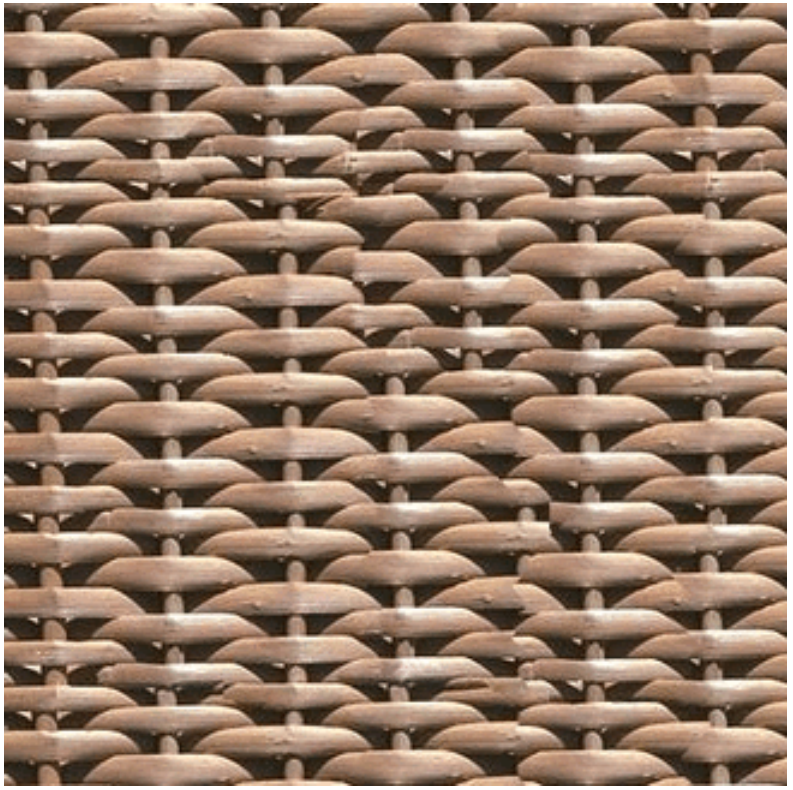


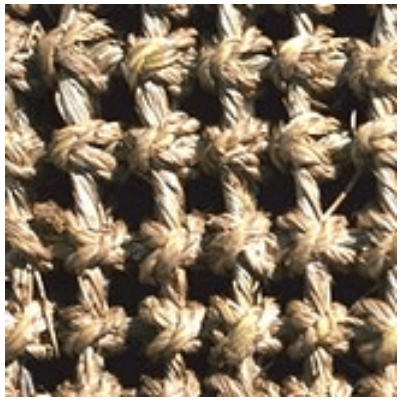


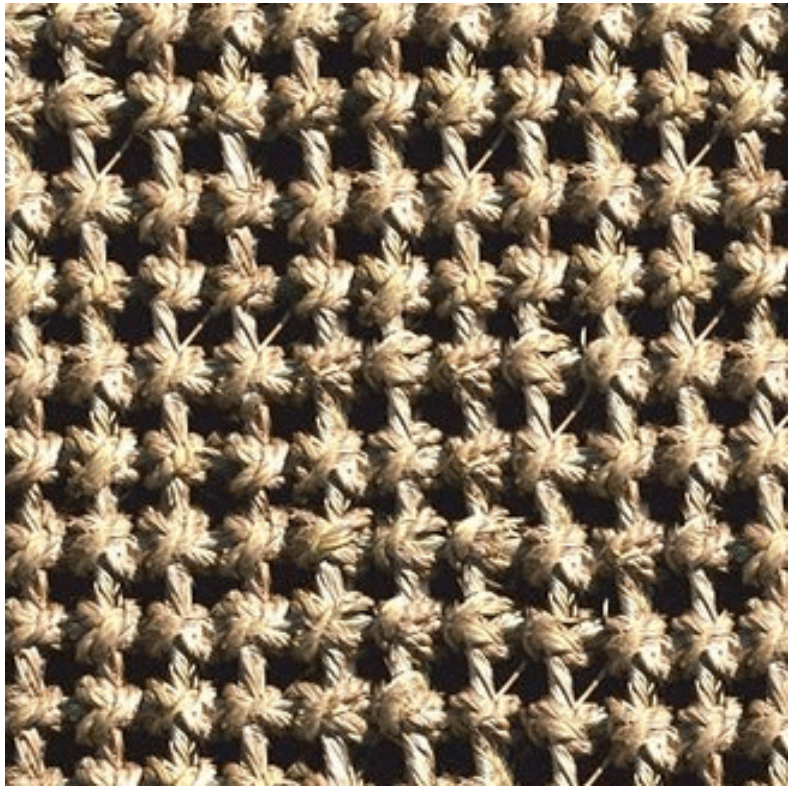
Minimal error boundary



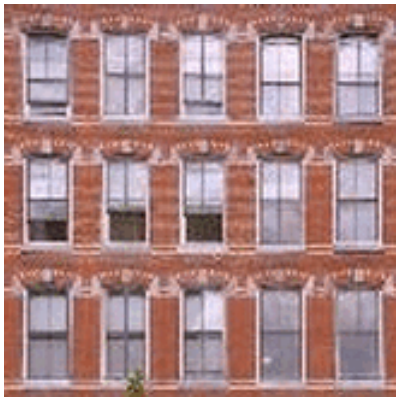




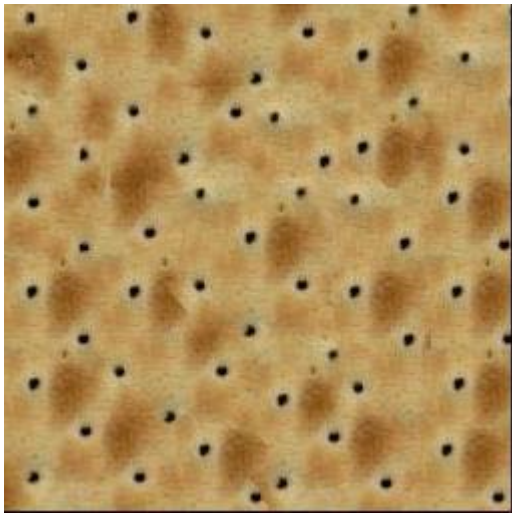


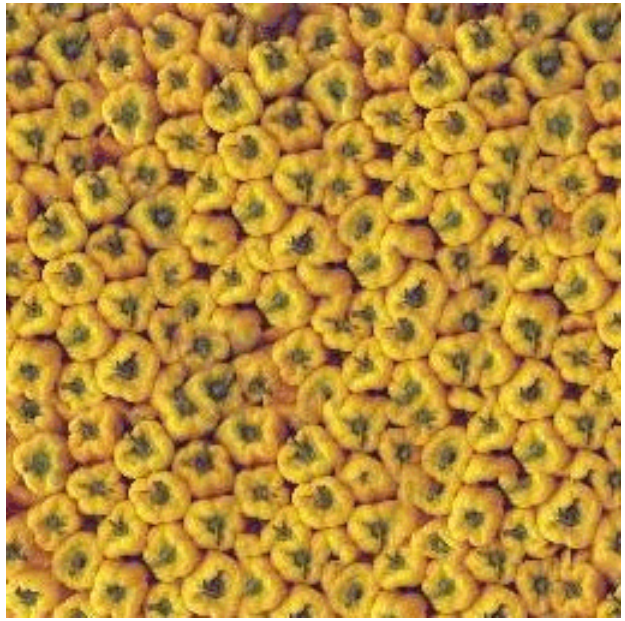


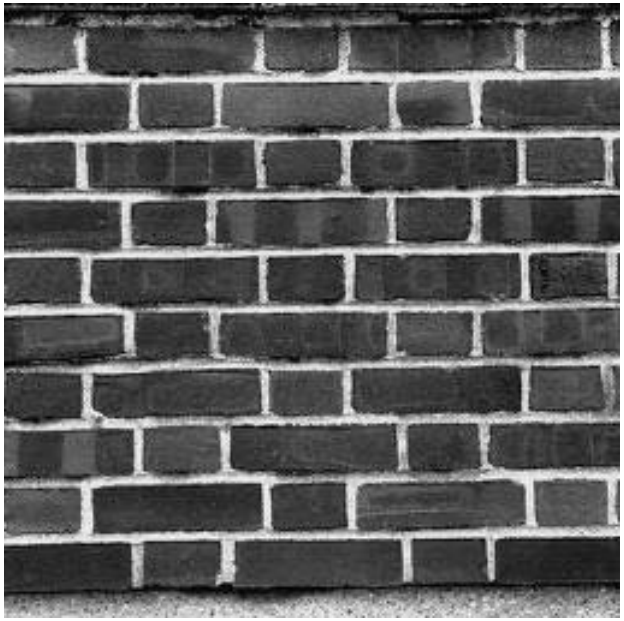
College Inn
Chicken Broth



College Inn
Chicken Broth

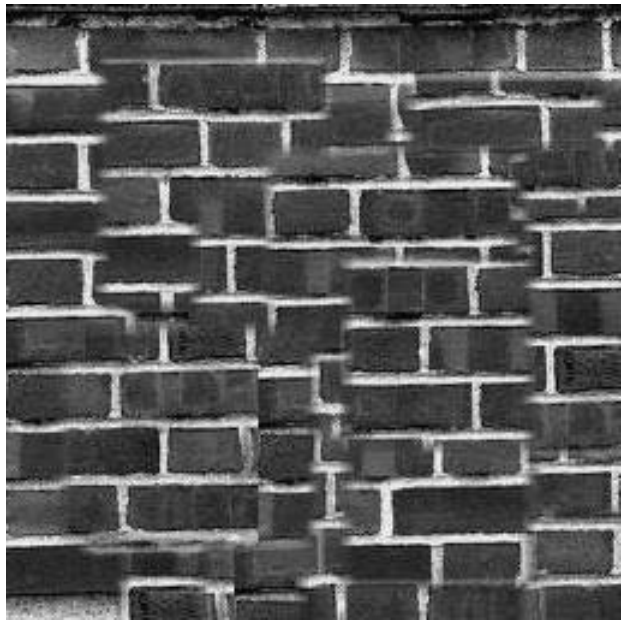




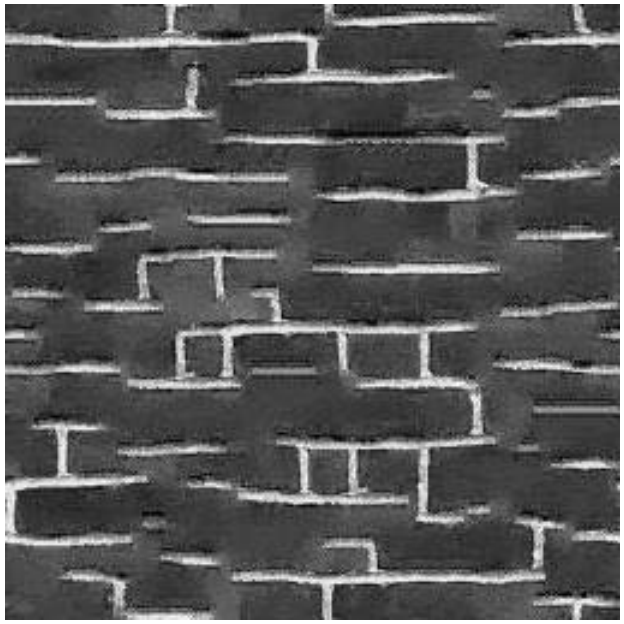


input image

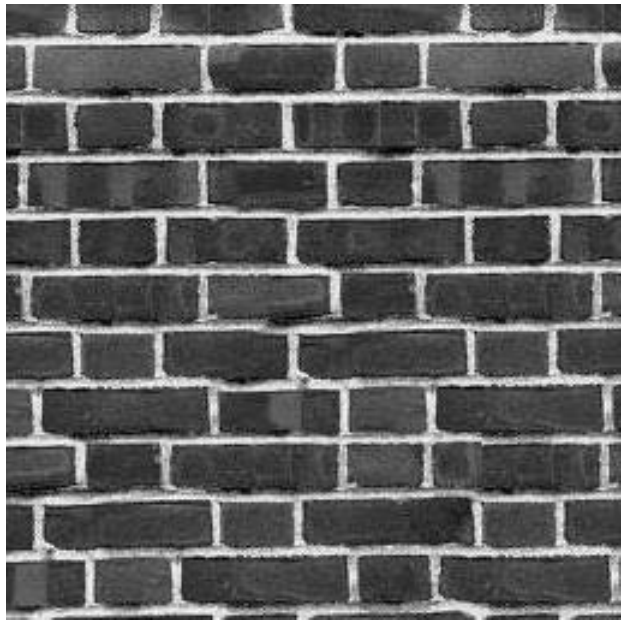
Portilla & Simoncelli



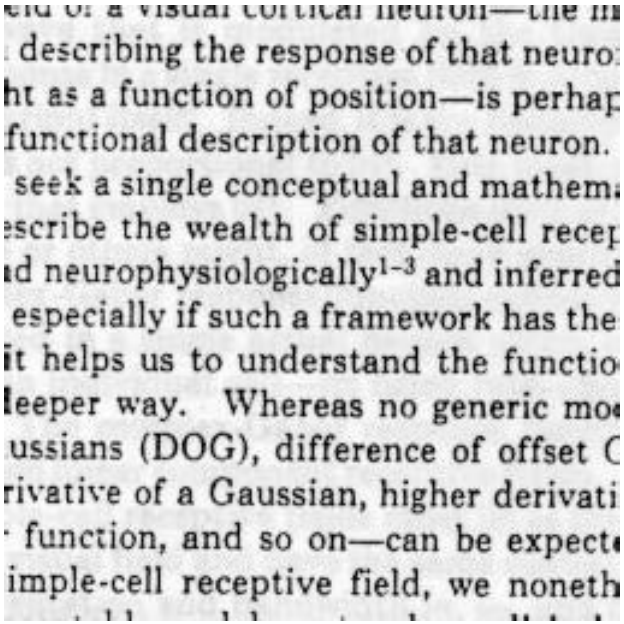
Xu, Guo & Shum



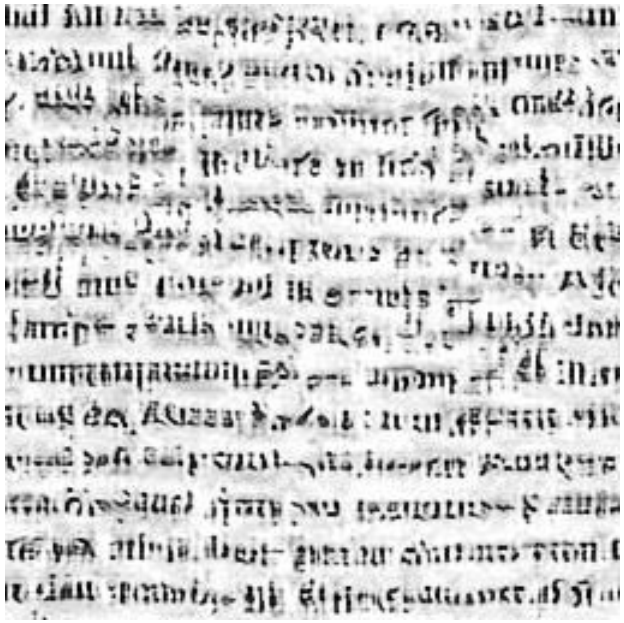
Wei & Levoy



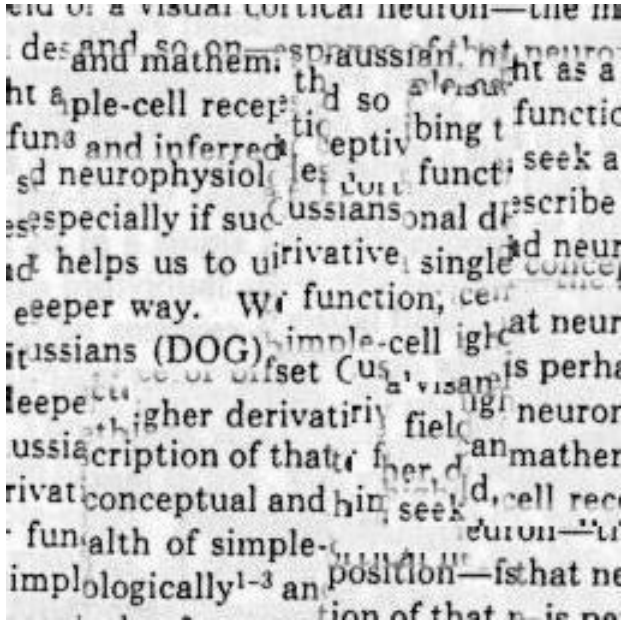
Quilting



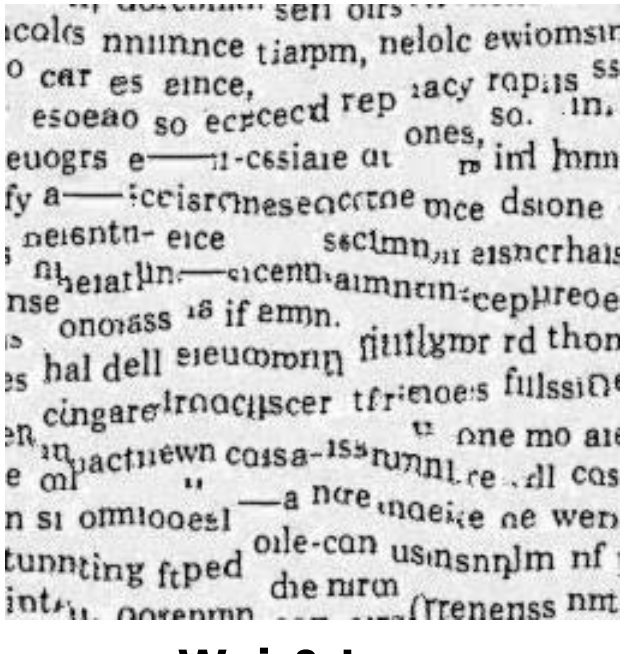
input image



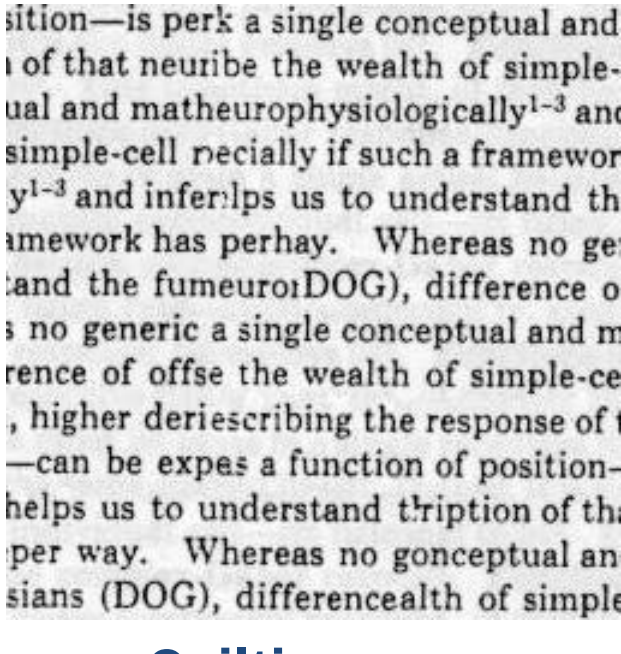
Portilla & Simoncelli



Xu, Guo & Shum

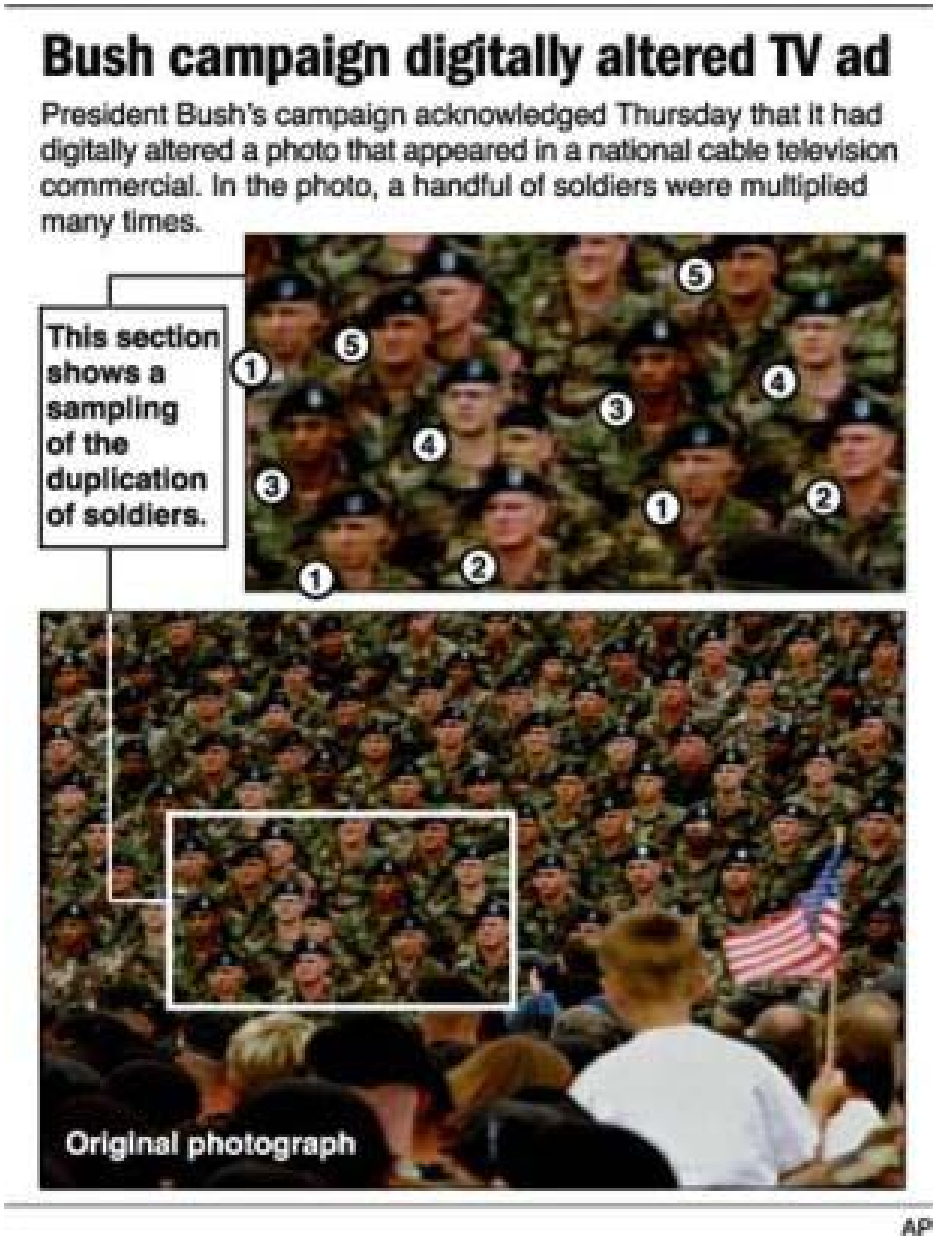


Wei & Levoy



Quilting

Political Texture Synthesis!



In-painting natural scenes

Criminisi, Perez, and Toyama. “Object Removal by Exemplar-based Inpainting,” Proc. CVPR, 2003.

Key idea: Filling order matters

In-painting Result

Image with Hole

Raster-Scan Order

Onion-Peel (Concentric Layers)

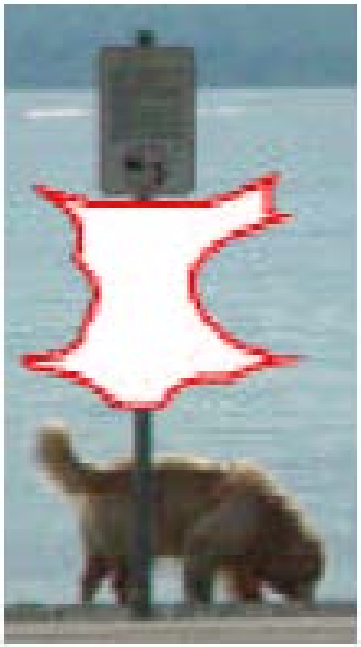
Gradient-Sensitive Order

Criminisi, Perez, and Toyama. "Object Removal by Exemplar-based Inpainting," Proc. CVPR, 2003.

Filling order

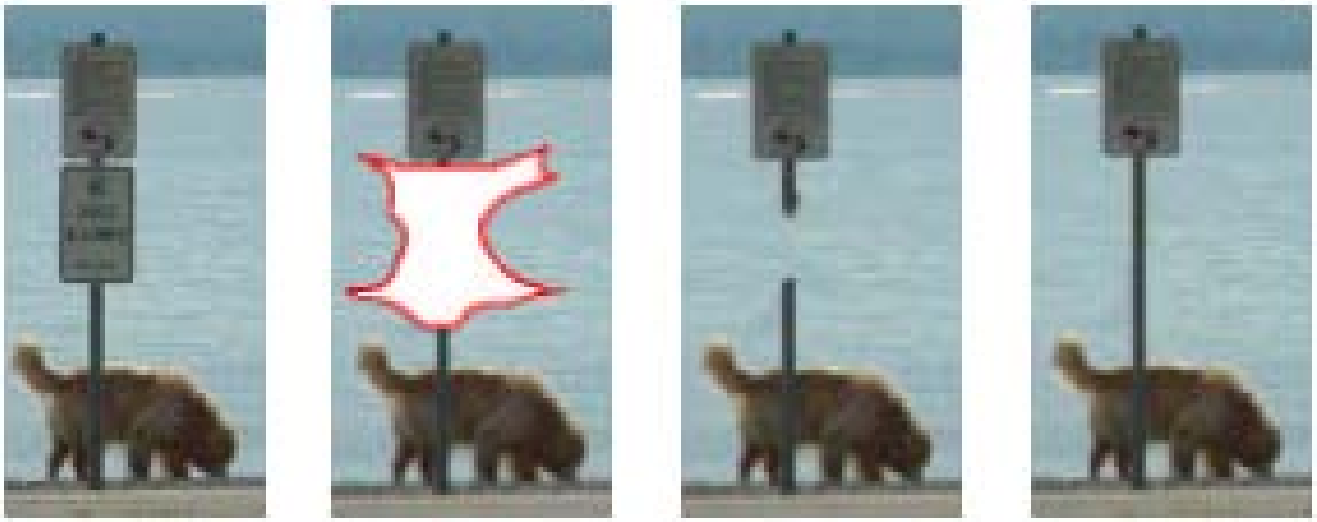
Fill a pixel that:

1. Is surrounded by other known pixels
2. Is a continuation of a strong gradient or edge



Criminisi, Perez, and Toyama. “Object Removal by Exemplar-based Inpainting,” Proc. CVPR, 2003.

Comparison



Original With Hole Onion-Ring Fill Criminisi

Criminisi, Perez, and Toyama. “Object Removal by Exemplar-based Inpainting,” Proc. CVPR, 2003.

Comparison

a

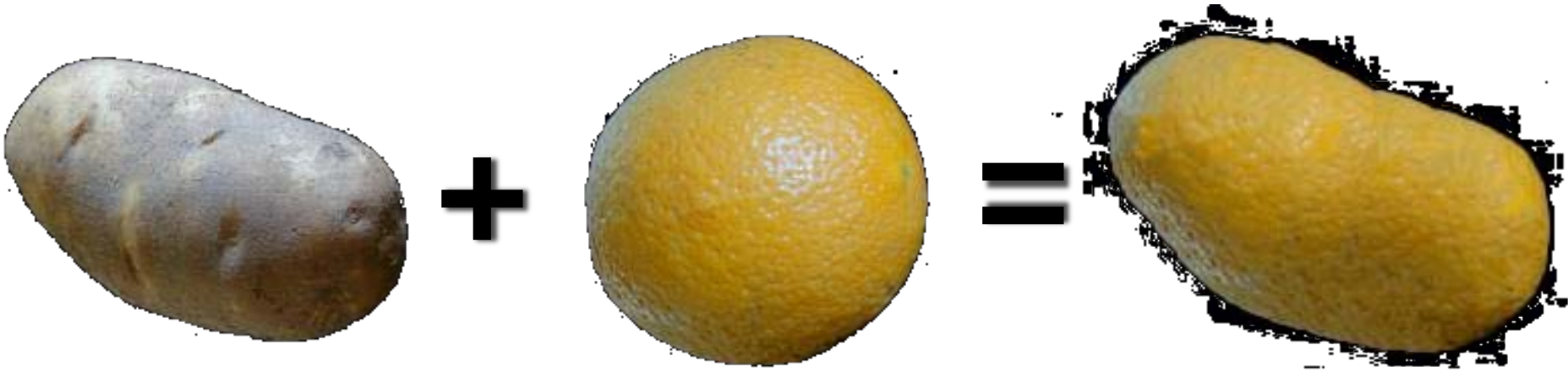
b

Concentric Layers

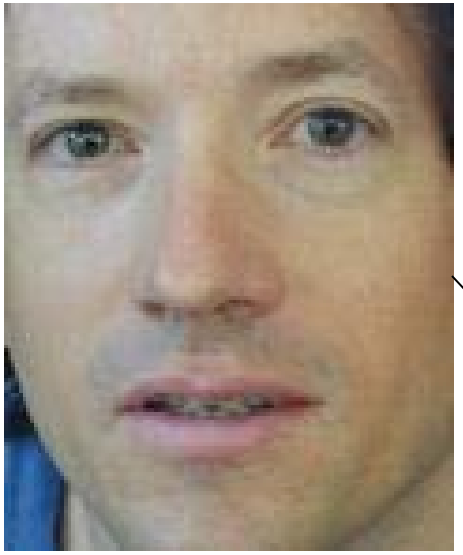
Gradient Sensitive

Texture Transfer

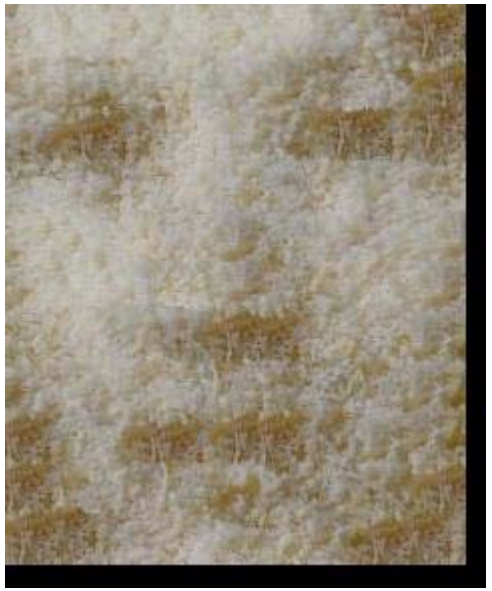
- Try to explain one object with bits and pieces of another object:



Texture Transfer



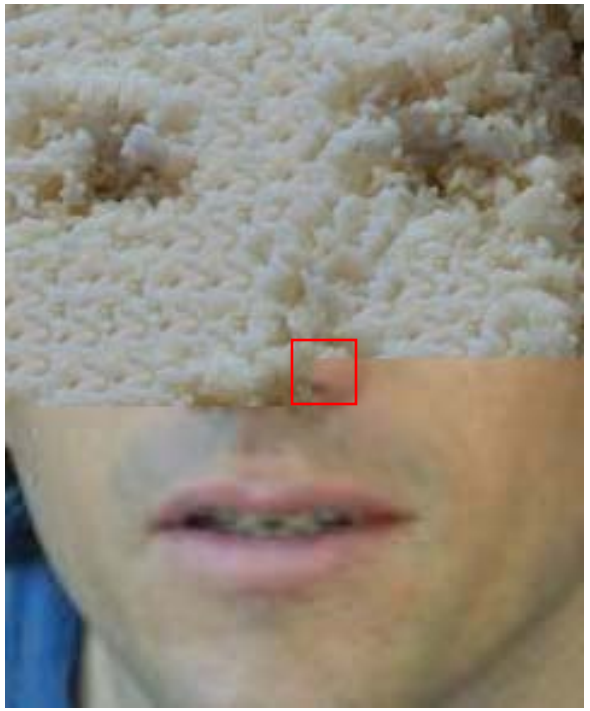
Constraint



Texture sample

Texture Transfer

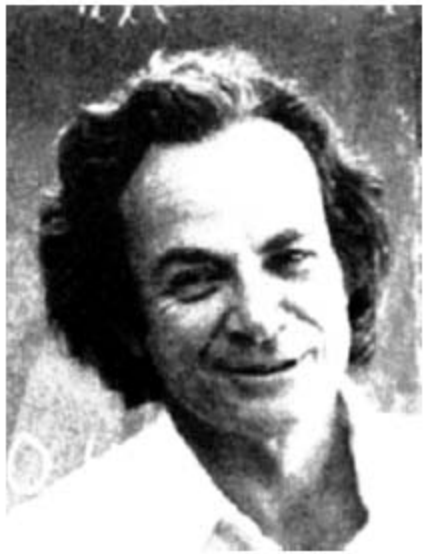
Take the texture from one image and “paint” it onto another object



Same as texture synthesis, except an additional constraint:

1. Consistency of texture
2. Similarity to the image being “explained”

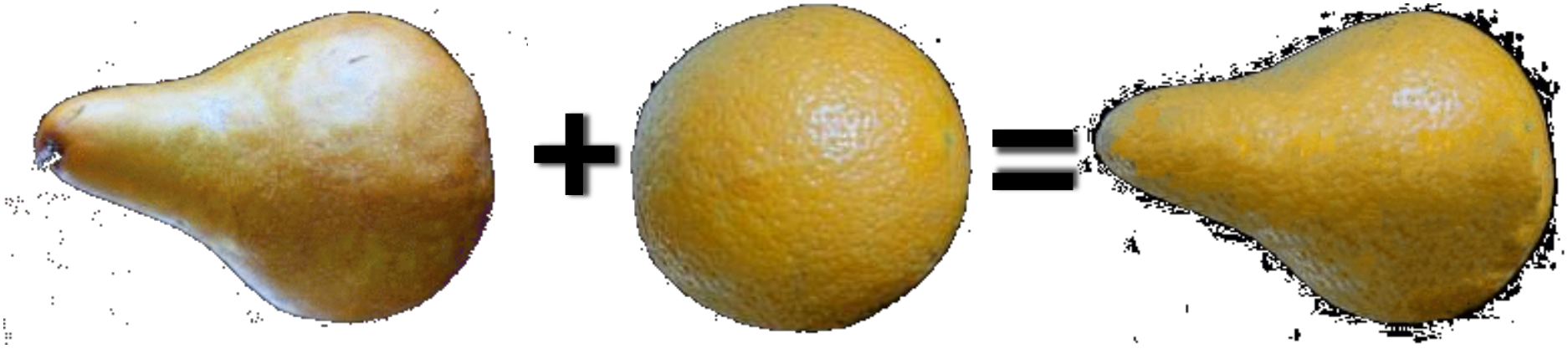
source texture

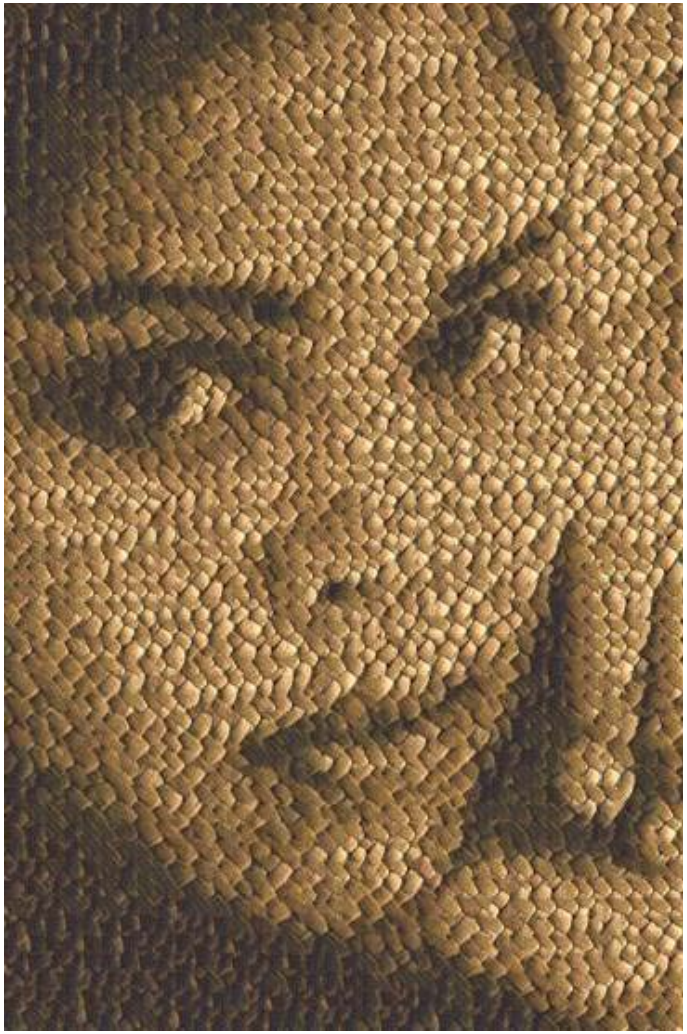


target image

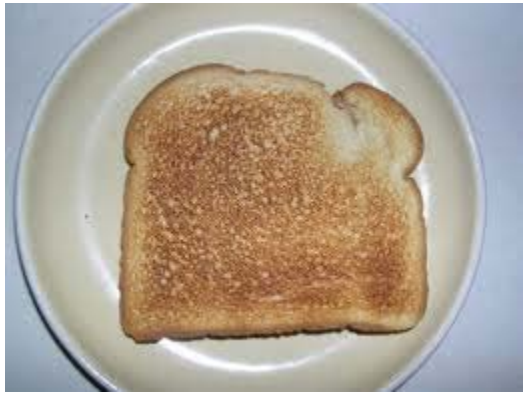
correspondence maps

texture transfer result





Make your own sacred toast



+

?

Makes a good final project

- Some combination of texture synthesis, hole-filling, texture transfer

- Or do it for extra points (e.g., for grad students)
  - Efros Leung 1999 synthesis (40 pts) OR Efros Freeman 2001 (60 pts) or Criminisi et al. 2004 hole-filling (60 pts)
  - Texture transfer (40 pts)

Related idea: Image Analogies

A

::

A’

B

::

B’

Image Analogies, Hertzmann et al. SG 2001

Blur Filter

Unfiltered source (A)

Filtered source (A')

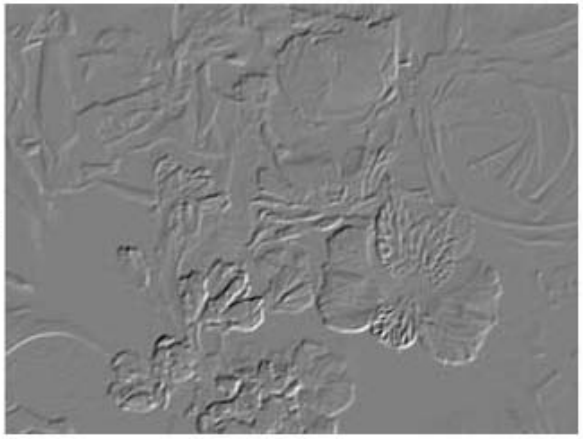
Unfiltered target (B)

Filtered target (B')

Edge Filter

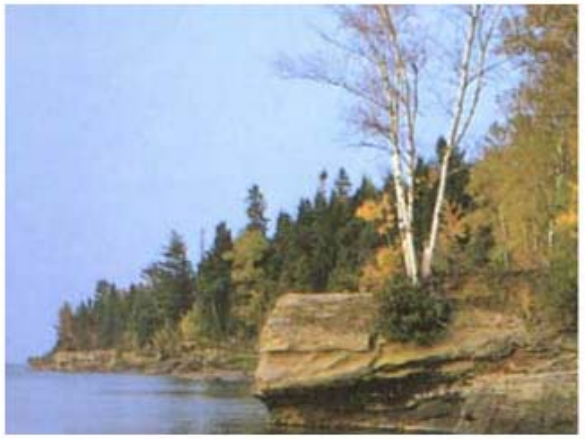
Unfiltered source (*A*)

:

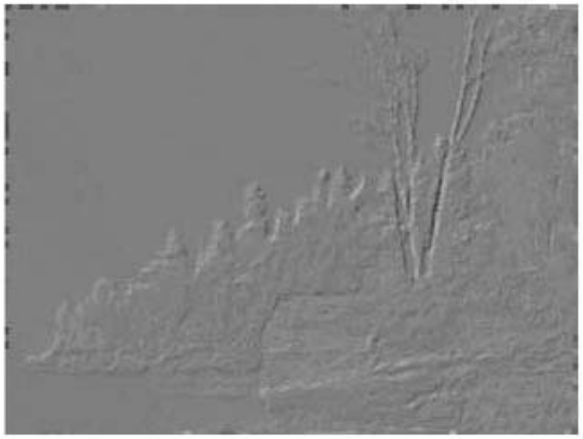


Filtered source (*A'*)

::



Unfiltered target (*B*)



Filtered target (*B'*)

Artistic Filters

A

A’

B

B’

Colorization

Unfiltered source (*A*)

:

Filtered source (*A'*)

::

Unfiltered target (*B*)

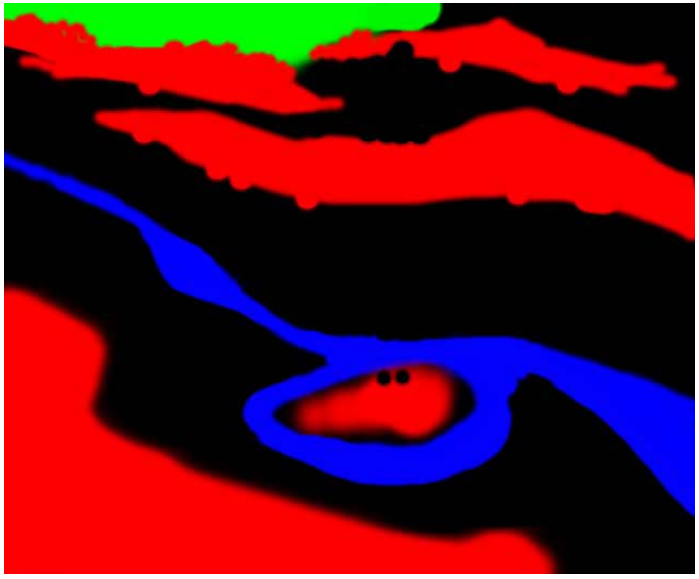
:

Filtered target (*B'*)

Texture-by-numbers

A

A’



B

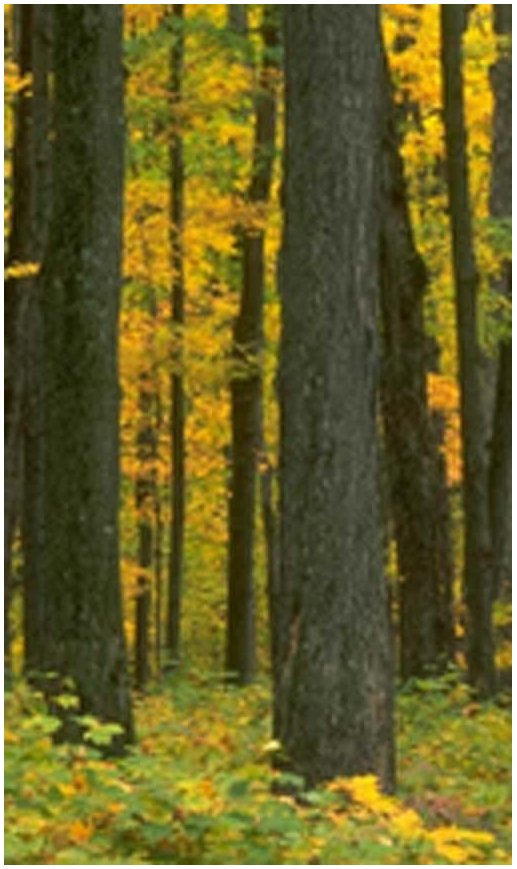
B’

Super-resolution

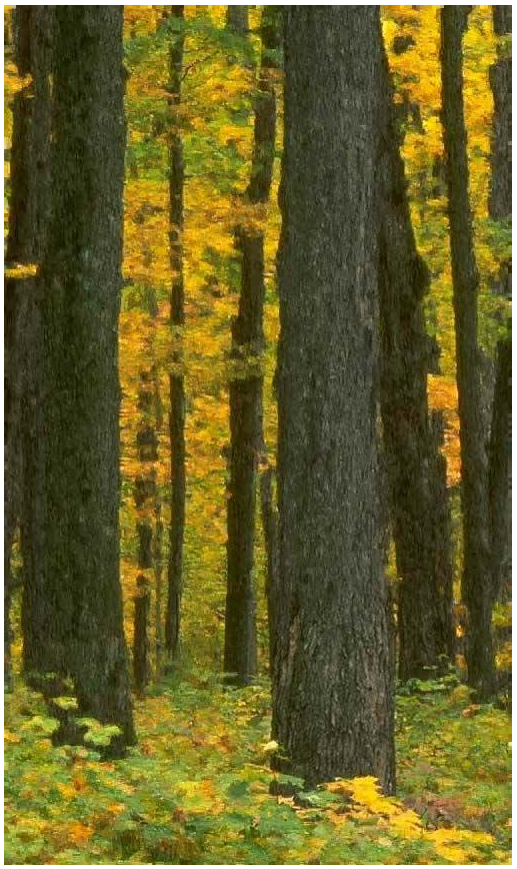
A

A’

Super-resolution (result!)

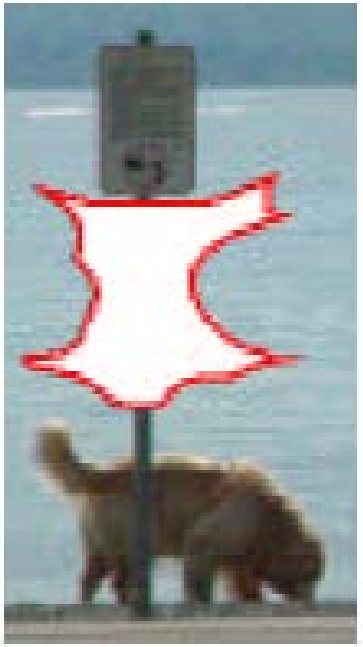


B



B’

Things to remember



- Texture synthesis and hole-filling can be thought of as a form of probabilistic hallucination

- Simple, similarity-based matching is a powerful tool
  - Synthesis
  - Hole-filling
  - Transfer
  - Artistic filtering
  - Super-resolution
  - Recognition, etc.

- Key is usually how to define similarity

Next class

- Blending and gradient-domain methods