

# Texture Synthesis

COS 526: Advanced Computer Graphics

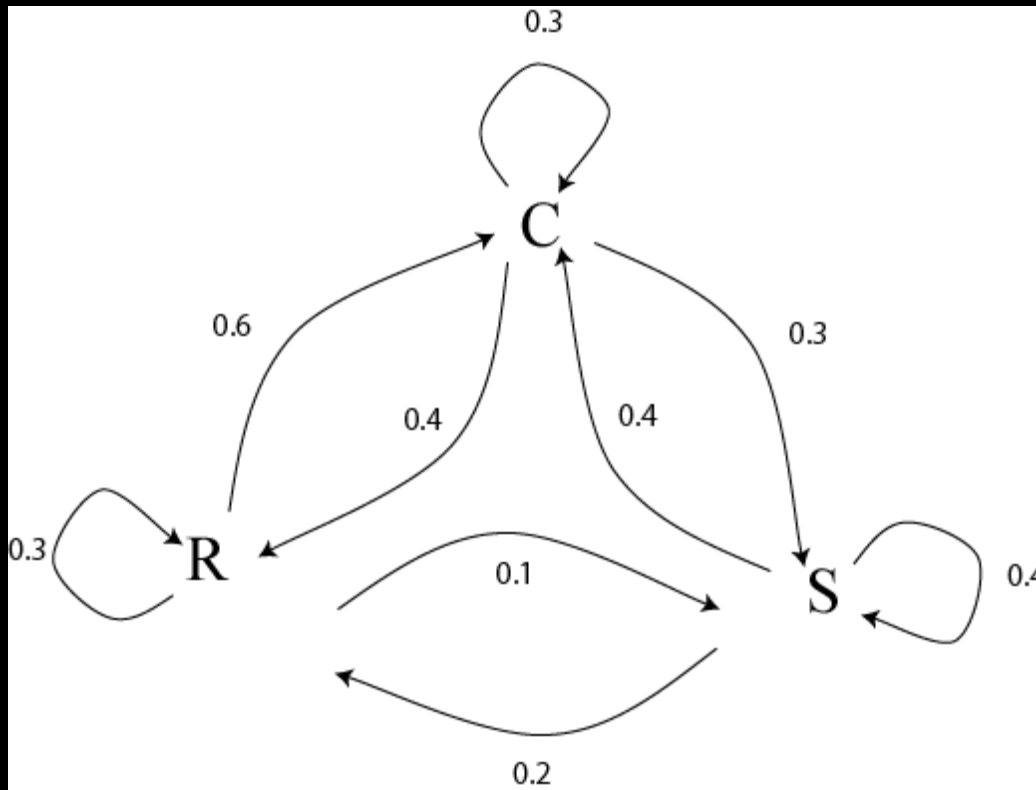


Slide credits: Alyosha Efros, Tom Funkhouser, Ravi Ramamoorthi.

# Weather Forecasting for Dummies

- Let's predict the weather:
  - Given today's weather only, we want to know tomorrow's
  - Suppose weather can only be {Sunny, Cloudy, Raining}
- Simple algorithm:
  - Over a long period of time, record:
    - How often S followed by R
    - How often S followed by S
    - Etc.
  - Compute percentages for each state:
    - $P(R|S)$ ,  $P(S|S)$ , etc.
  - Predict the state with highest probability!
  - It's a Markov Chain

# Markov Chain



$$\begin{pmatrix} 0.3 & 0.6 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

What if we know today's *and* yesterday's weather?

# Text Synthesis

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

**WE NEED TO EAT CAKE**

# Mark V. Shaney (Bell Labs)

- Results (using alt.singles corpus):
  - *“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”*

# Texture

- Texture depicts spatially repeating patterns
- Many natural phenomena are textures



radishes



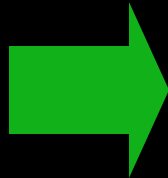
rocks



yogurt

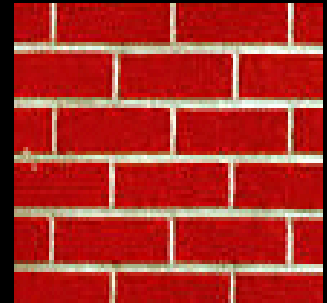
# Texture Synthesis

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

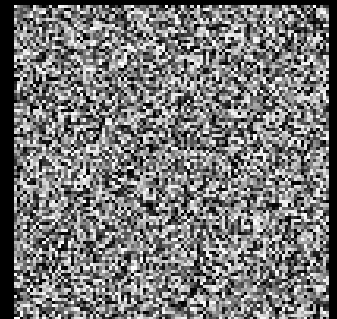


# The Challenge

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



**repeated**



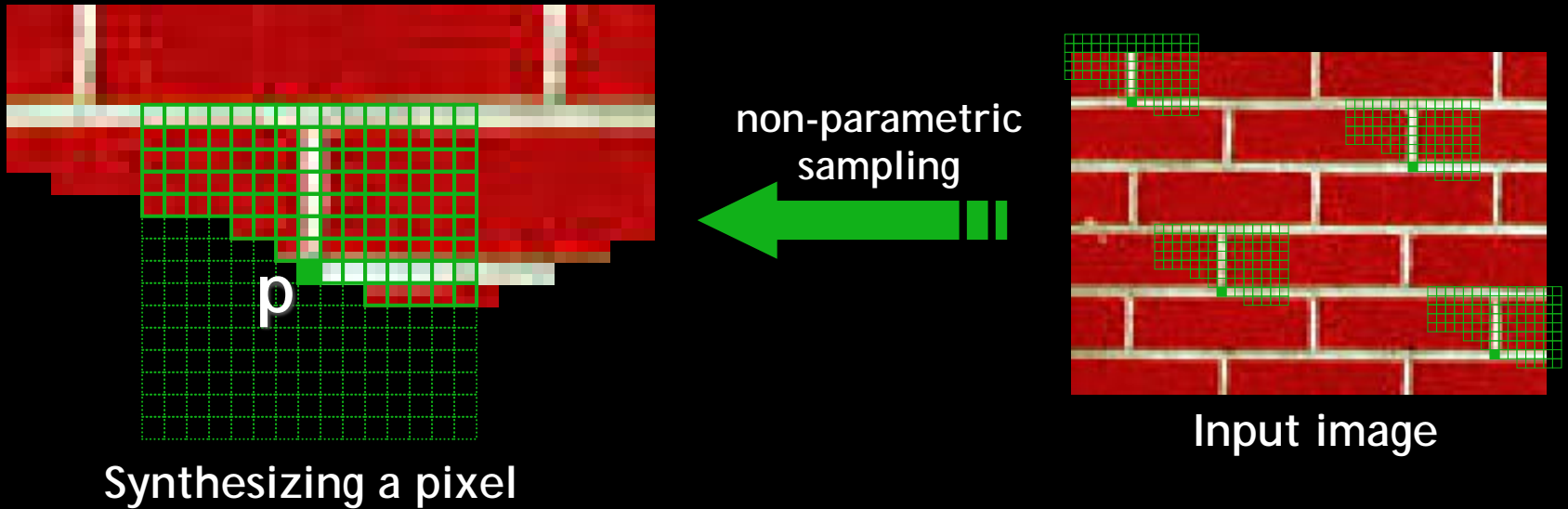
**stochastic**



**Both?**



# Efros & Leung Algorithm

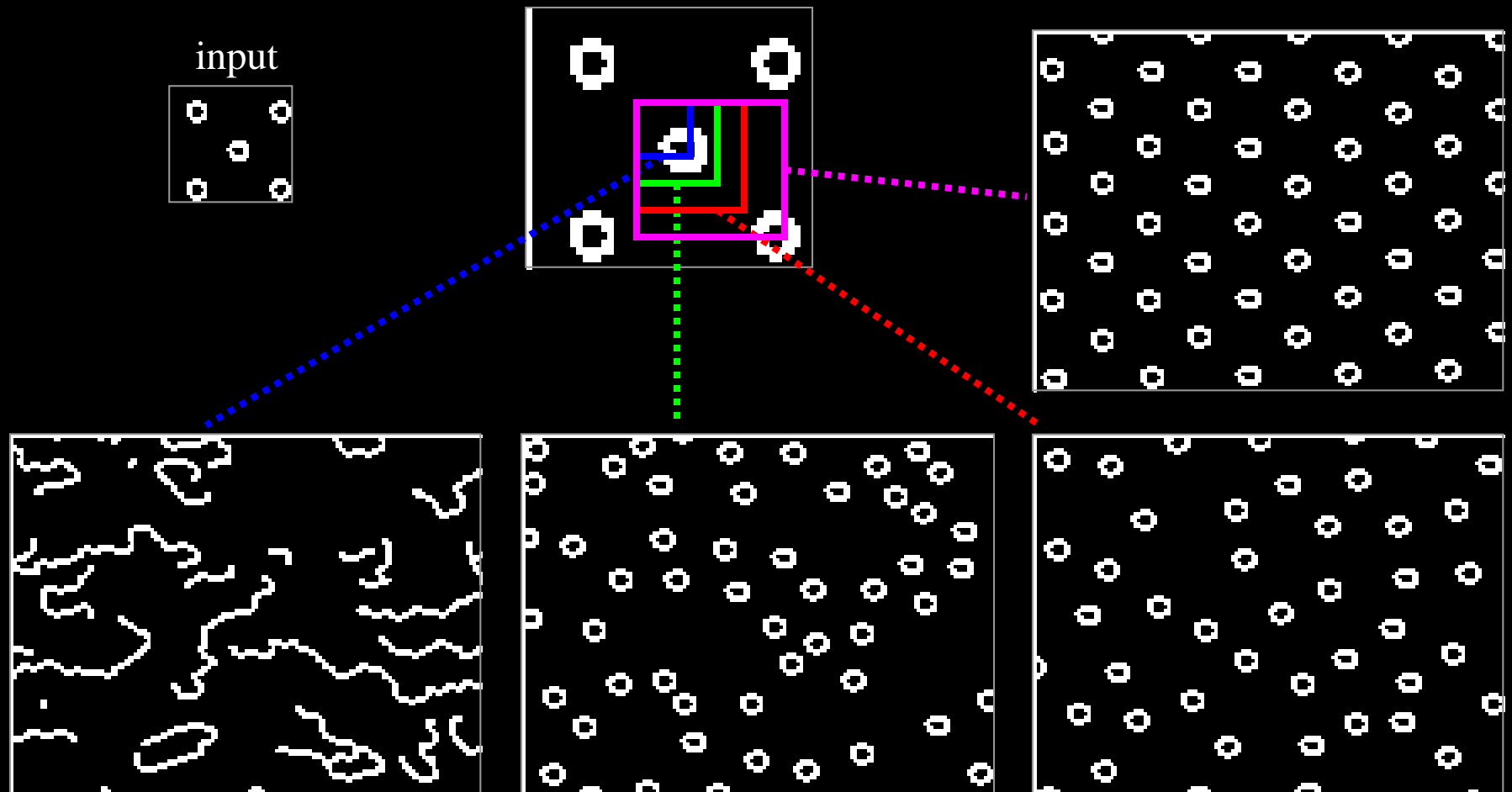


- Assuming Markov property, compute  $P(\mathbf{p}|\mathbf{N}(\mathbf{p}))$ 
  - Building explicit probability tables infeasible
  - Instead, we *search the input image* for all similar neighborhoods — that's our pdf for  $\mathbf{p}$
  - To sample from this pdf, just pick one match at random

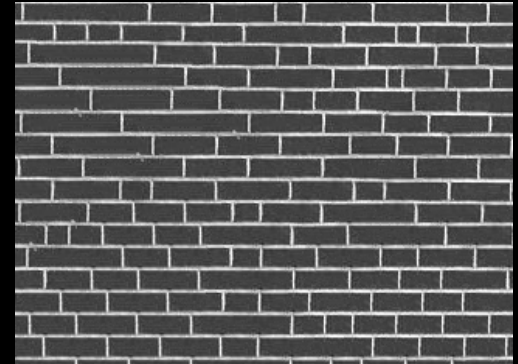
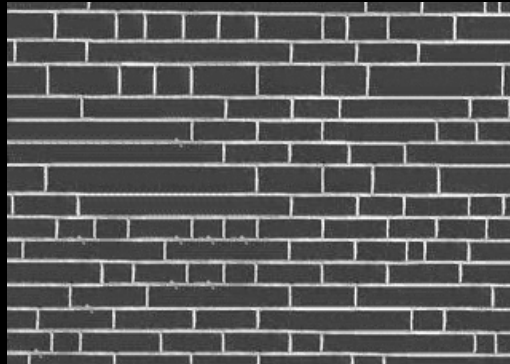
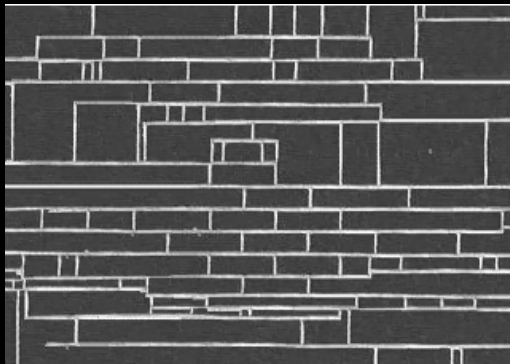
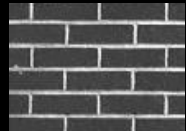
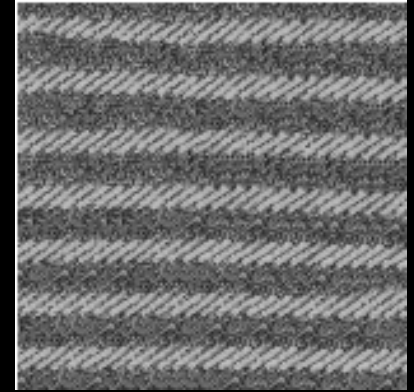
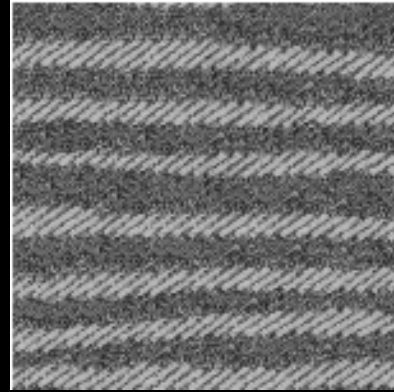
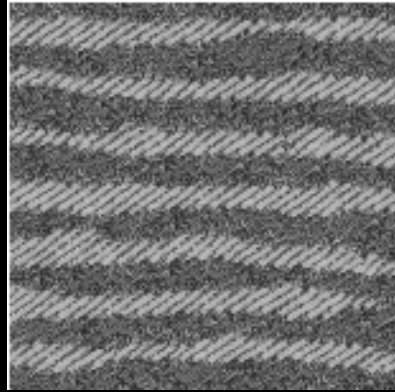
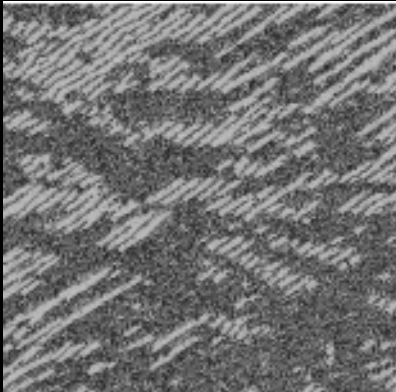
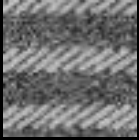
# Some Details

- 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

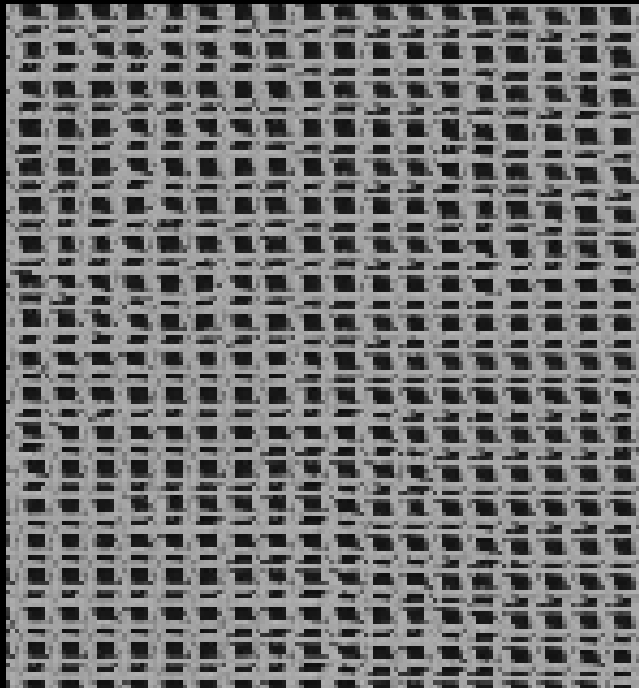
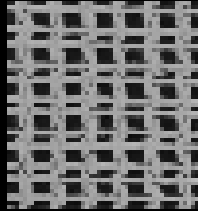


Increasing window size

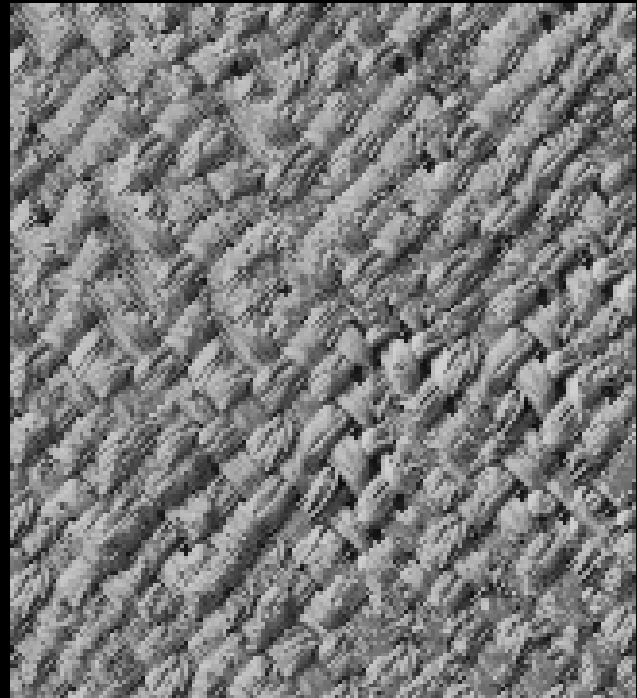


# Synthesis Results

french canvas



rafia weave

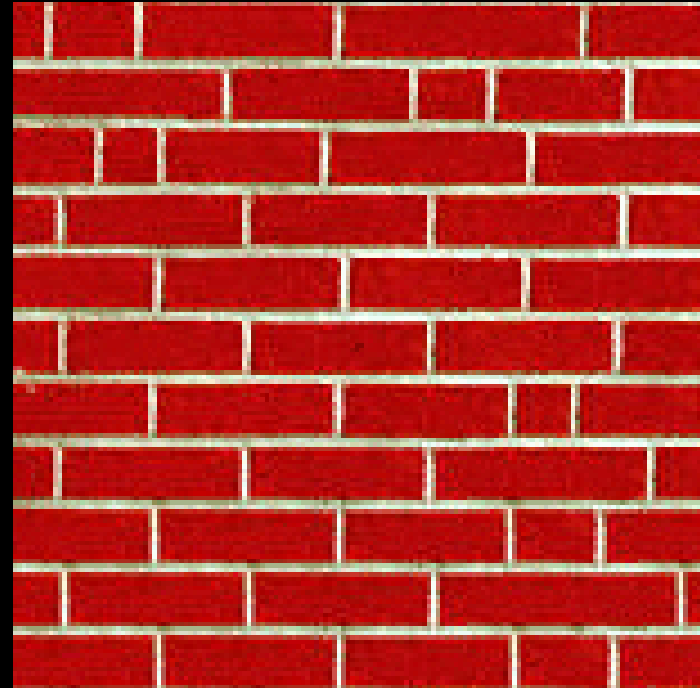
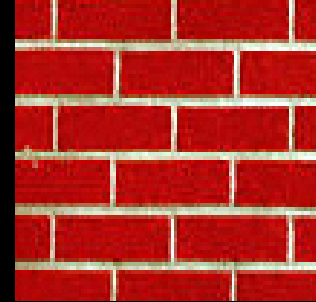


# More Results

white bread



brick wall

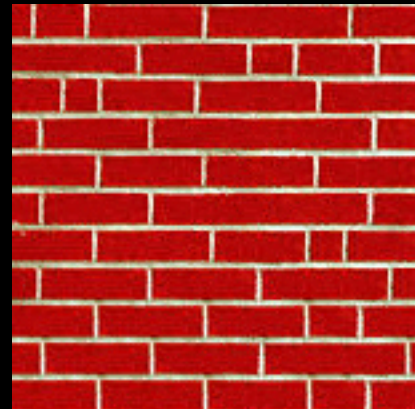
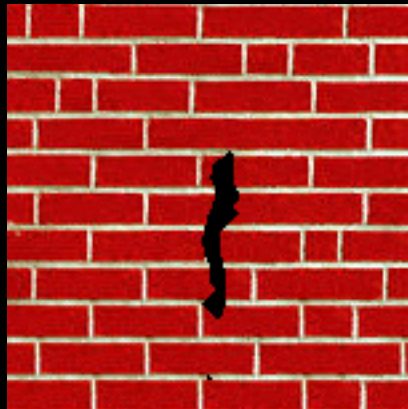


# Homage to Shannon

ing in the unsensational  
r Dick Gephardt was fai  
rful riff on the looming  
nly asked, "What's your  
tions?" A heartfelt sigh  
story about the emergen  
es against Clinton. "Boy  
g people about continuin  
hardt began, patiently obs  
e, that the legal system h  
g with this latest tanger

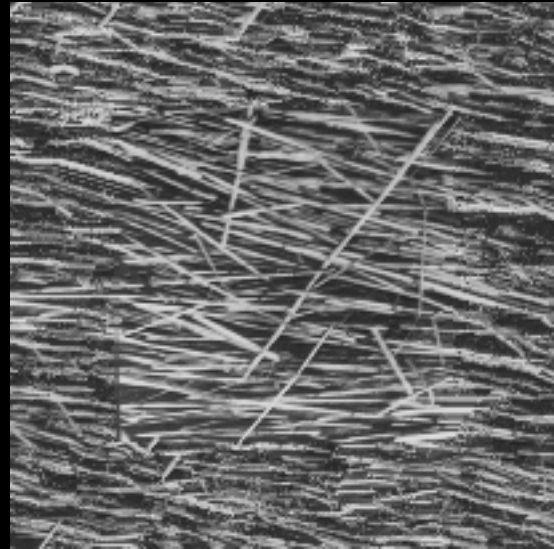
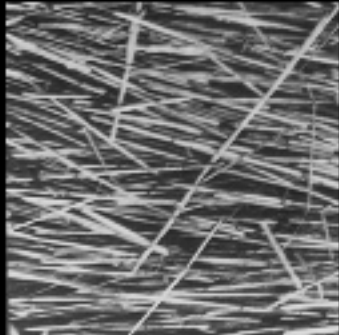
thaim, them. "Whnephartfe lartifelintomimer,  
fel ck Clirtioout omaim thartfelins, f out s anento  
the ry onst wartfe lck Gephtoomimeationl sigab  
Cllooufti Clinut Cil riff on, hat's y'odn, parut tly  
ons yontonsteht wasked, paim t sahe loo' riff on  
nskoneploourtfeas leil A nst Clit, "Wleontongal s  
k Clirtioouirtfepe.ong pme abegal fartfenstemem  
tiensteneltorydt telemephminsverdt was agemer  
ff ons artientont Cling peme as,rtfe atih,"Boui s  
nal s fartfelt sig pedrtihdt ske abounutie aboutioo  
tfaonewwas yow aboronthardt thatins fain, ped,  
ains, them, pabout wasy arfuit couitly d, l n A h  
ble emthrdngbooreme agas fa bontinsyst Clinut  
ory about continst Clipeouinst Cloke agatiff out C  
stome zinemen tly ardt beoraboul n, thenly as t C  
cons faimeme Diontont wat coutlyohgans as fan  
ien, phrtfaul, "Wbaut cout congagal comininga  
mifmst Cliiy abon'al coountha.emungairt tfoun  
The loocrystan loontieph, intly on, tieoplegatick C  
aul fatieeontly atie Dioniomt wal s f t begae ener  
nthahgat's epenhhmas fan, "intchthory abons

# Hole Filling





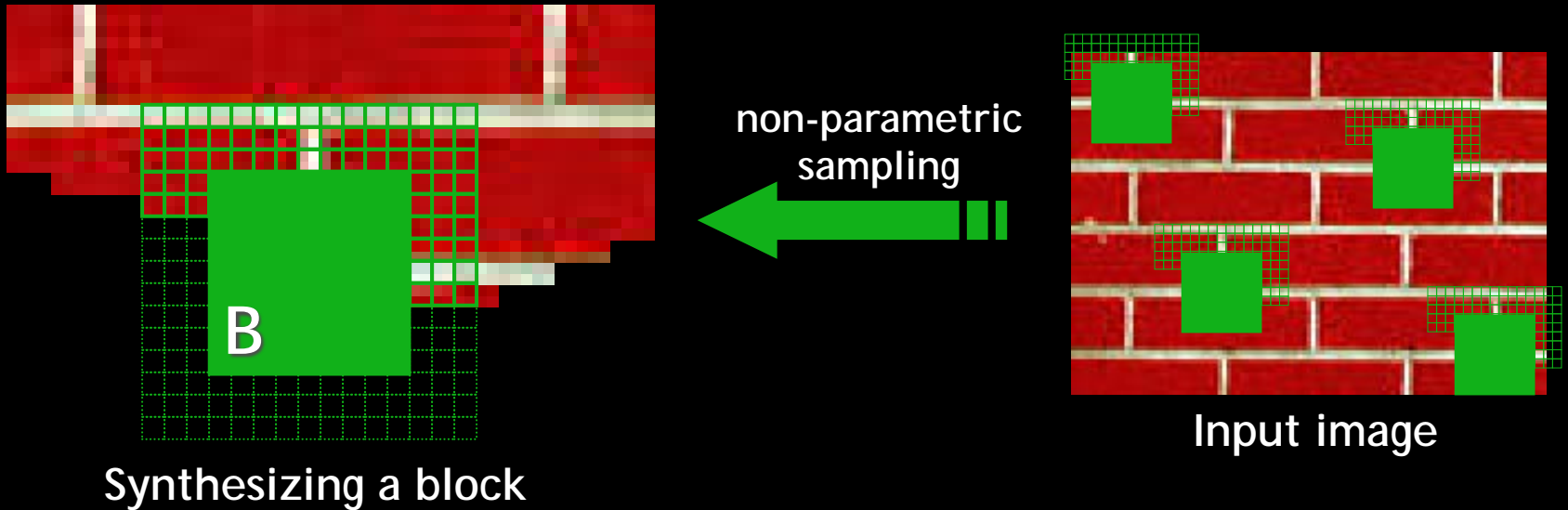
# Extrapolation



# Summary

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

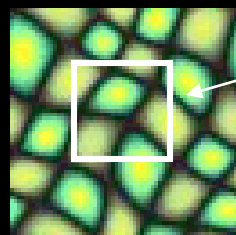
# Image Quilting [Efros & Freeman]



- Observation: neighbor pixels are highly correlated

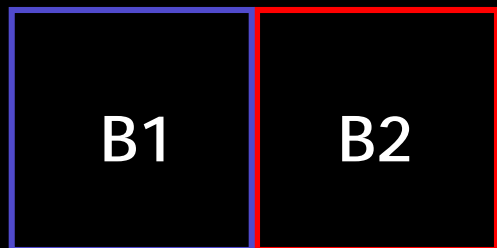
Idea: unit of synthesis = block

- Exactly the same but now we want  $P(B|N(B))$
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

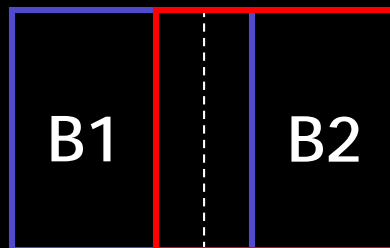


block

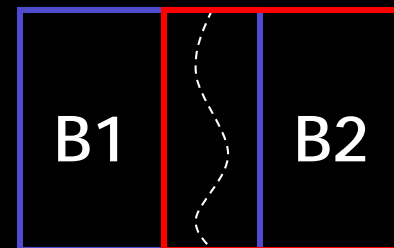
Input texture



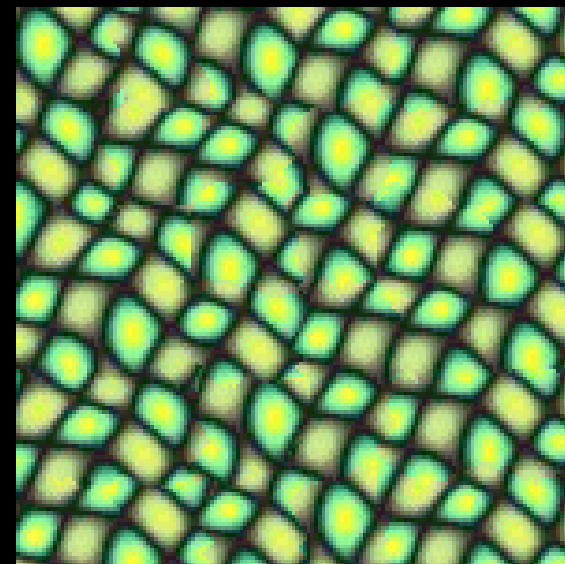
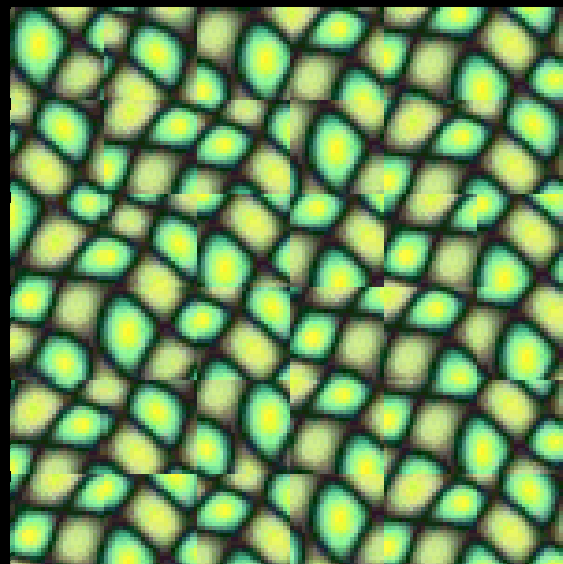
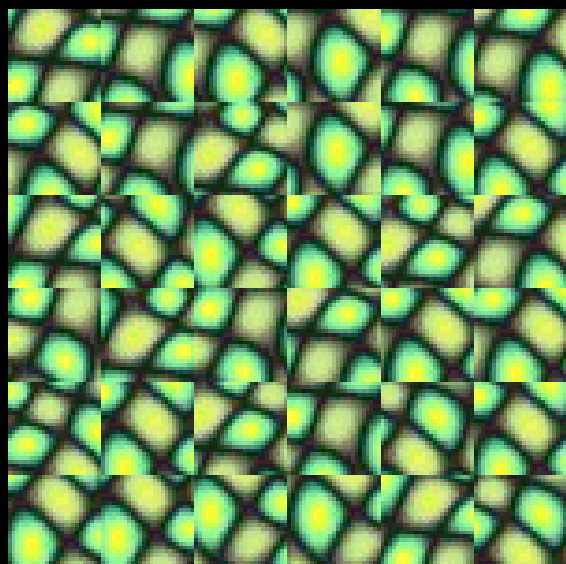
Random placement  
of blocks



Neighboring blocks  
constrained by overlap

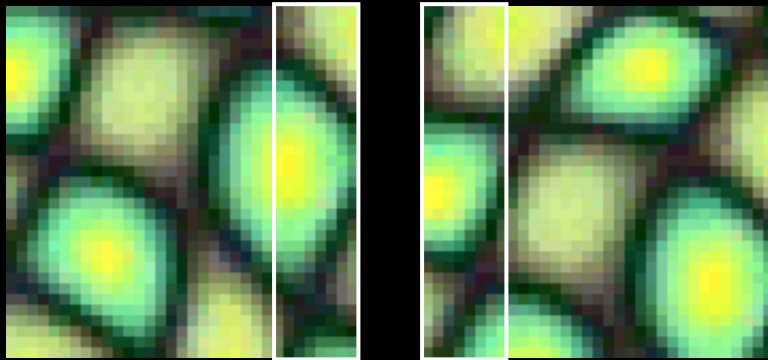


Minimal error  
boundary cut

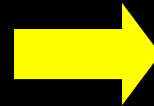
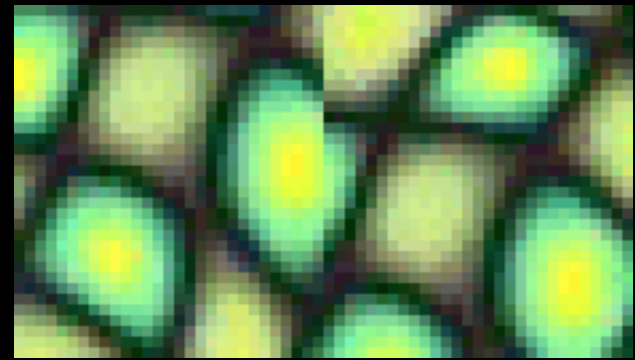


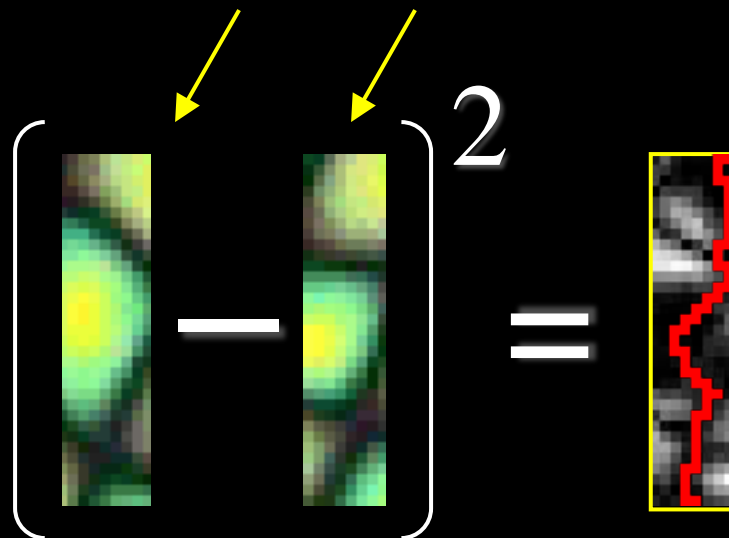
# Minimal error boundary

overlapping blocks

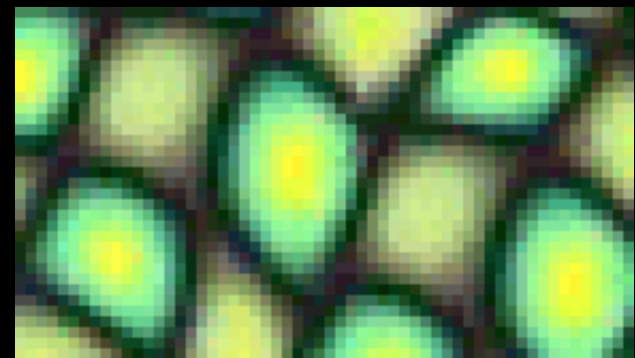


vertical boundary




$$\left[ \begin{array}{c} \text{block 1} \\ - \\ \text{block 2} \end{array} \right]^2 = \text{boundary map}$$
The diagram shows two overlapping blocks of the cell image. A yellow arrow points from the top-left corner of the first block to the first block in the subtraction. Another yellow arrow points from the top-right corner of the second block to the second block in the subtraction. The result of the subtraction is a grayscale image with a red line indicating the boundary. The red line is a jagged, step-like line that follows the boundary of the cells.

overlap error

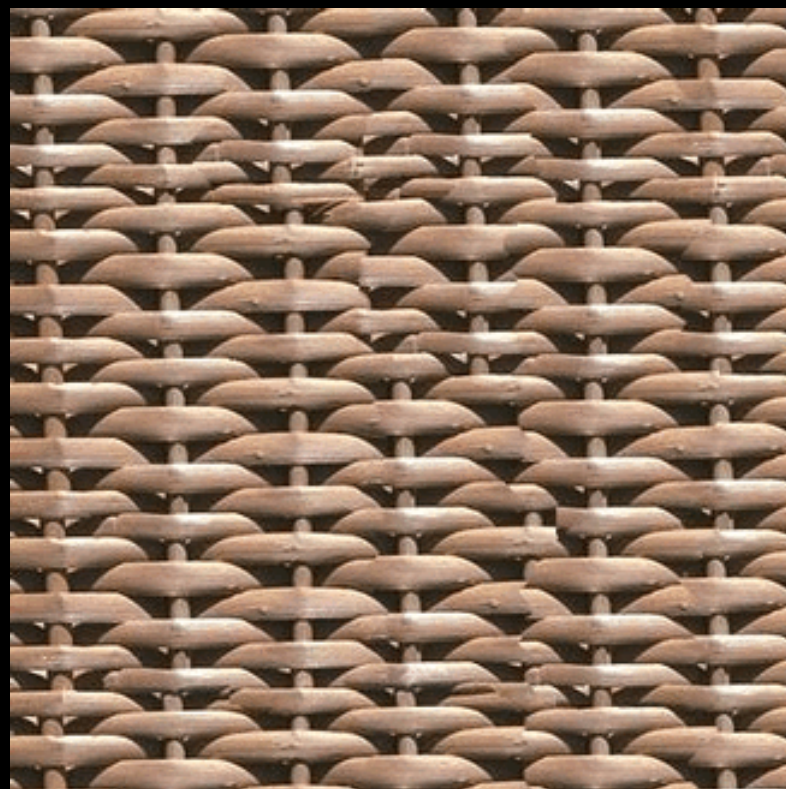
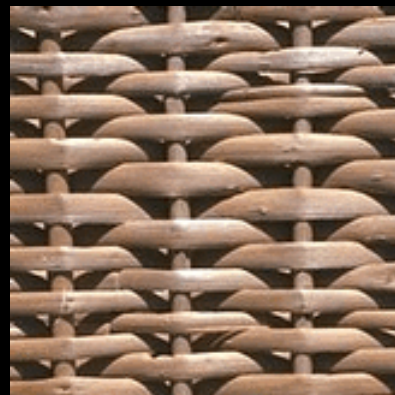


min. error boundary

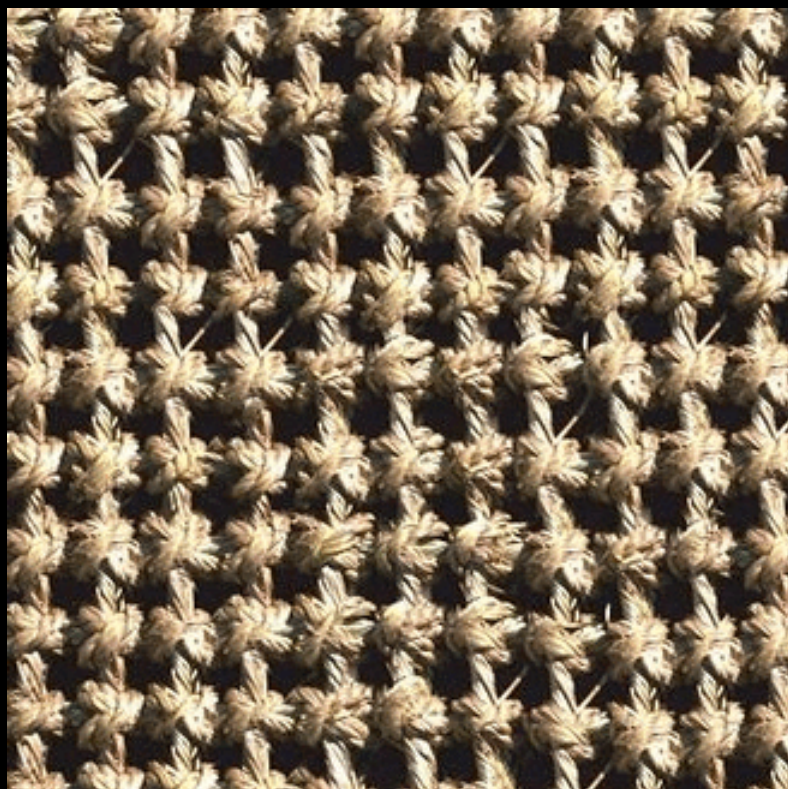
# Our Philosophy

- The “Corrupt Professor’s Algorithm”:
  - Plagiarize as much of the source image as you can
  - Then try to cover up the evidence
- Rationale:
  - Texture blocks are by definition correct samples of texture so problem only connecting them together

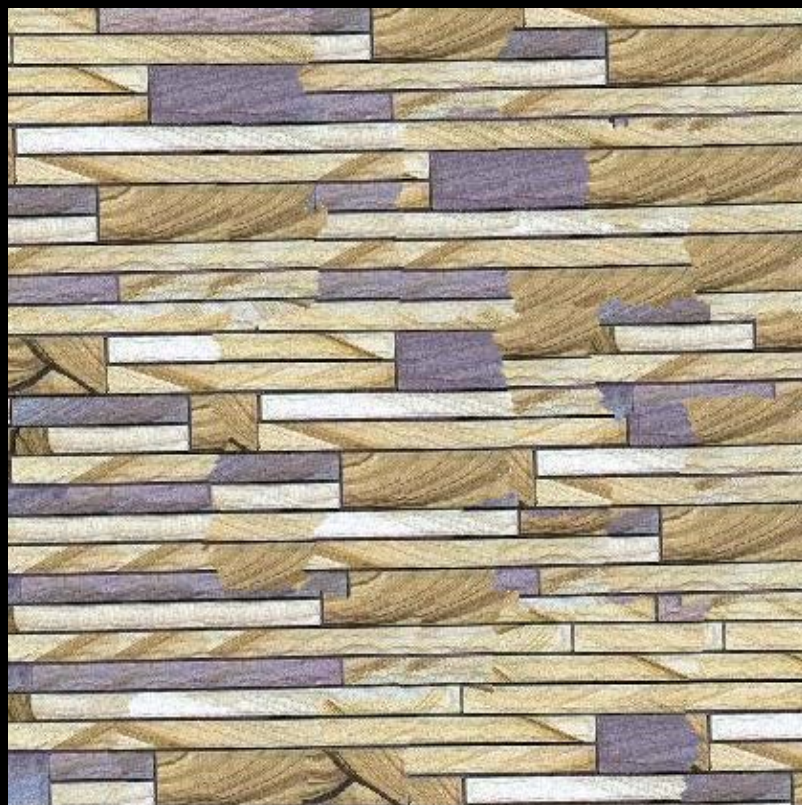










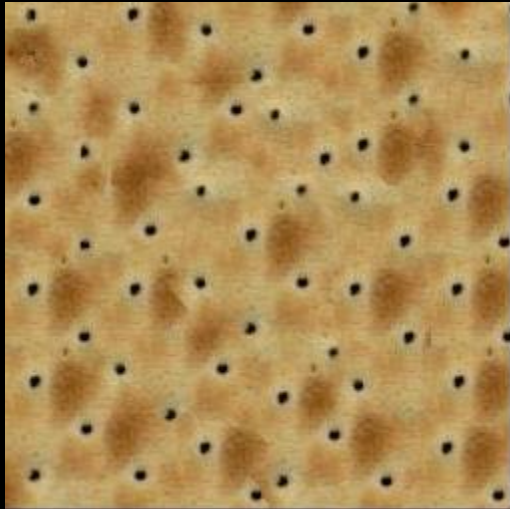
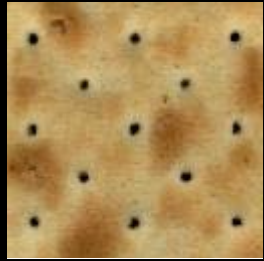










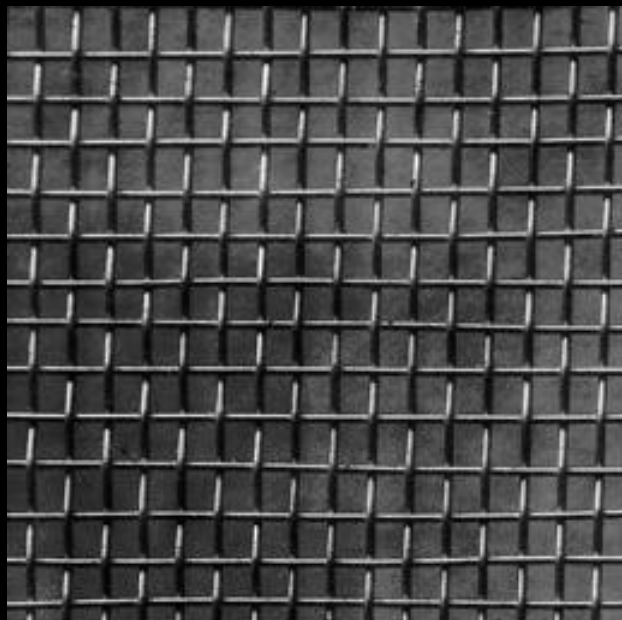




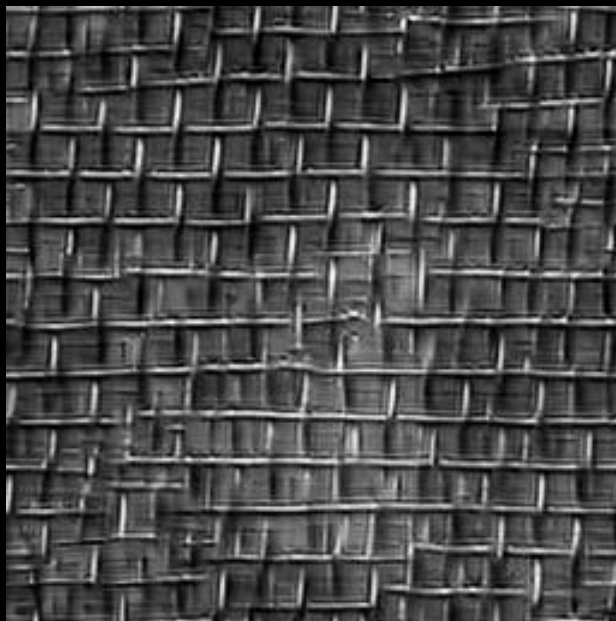


# Failures (Chernobyl Harvest)

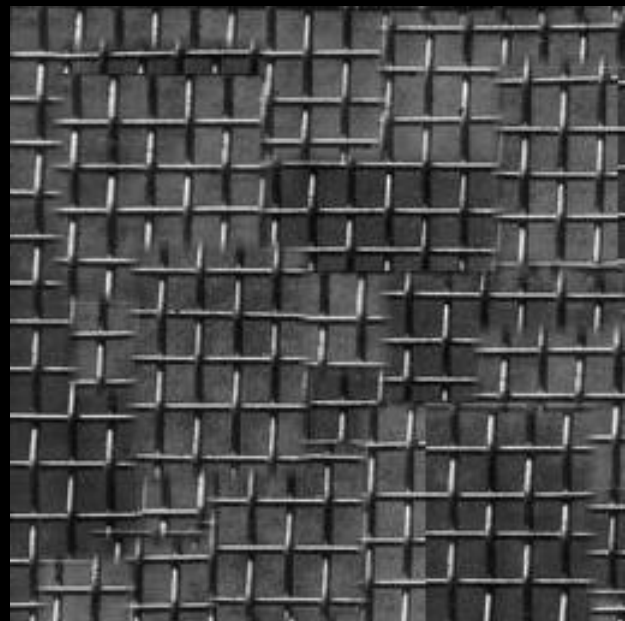




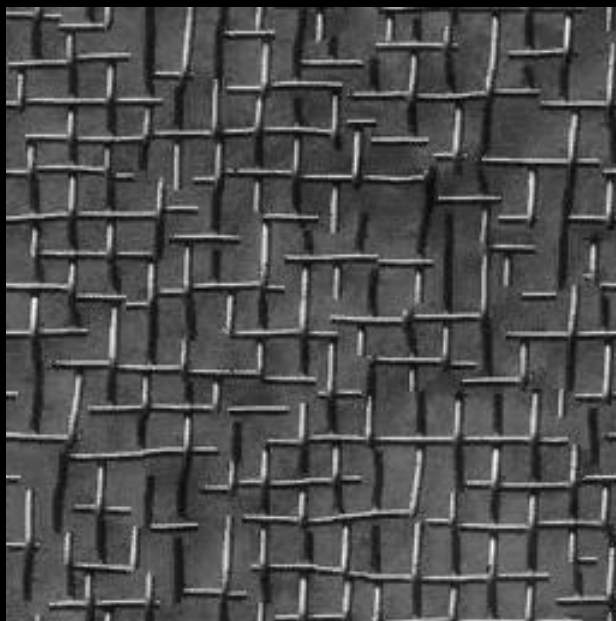
input image



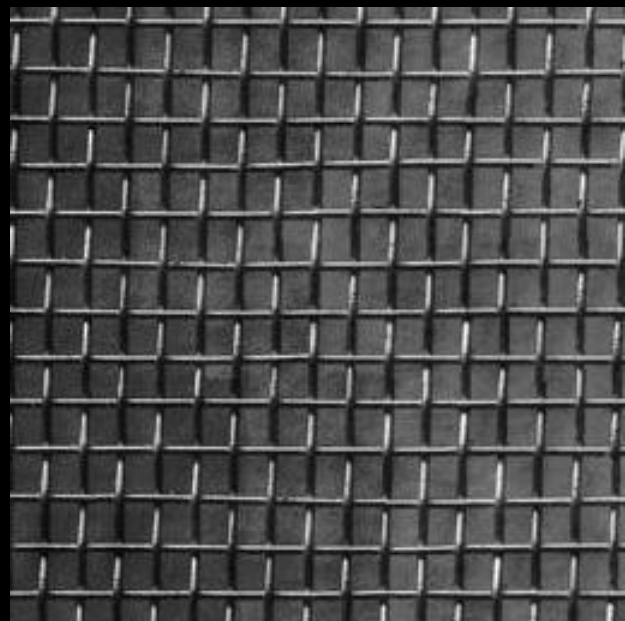
Portilla & Simoncelli



Xu, Guo & Shum

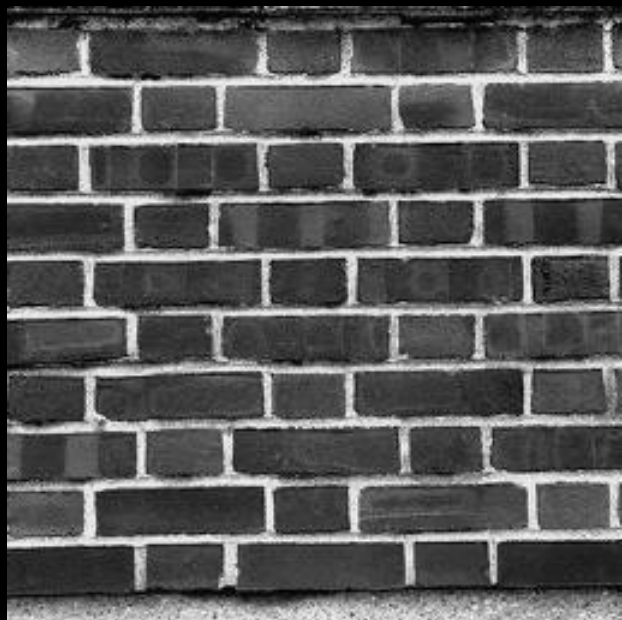


Wei & Levoy



Our algorithm

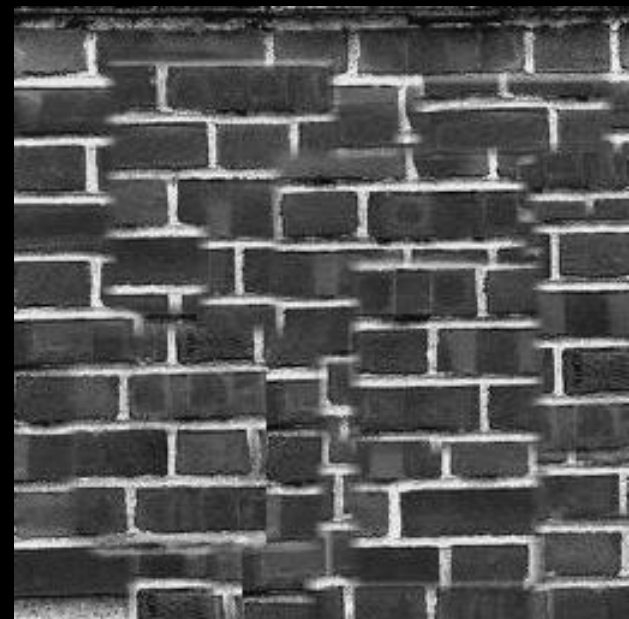




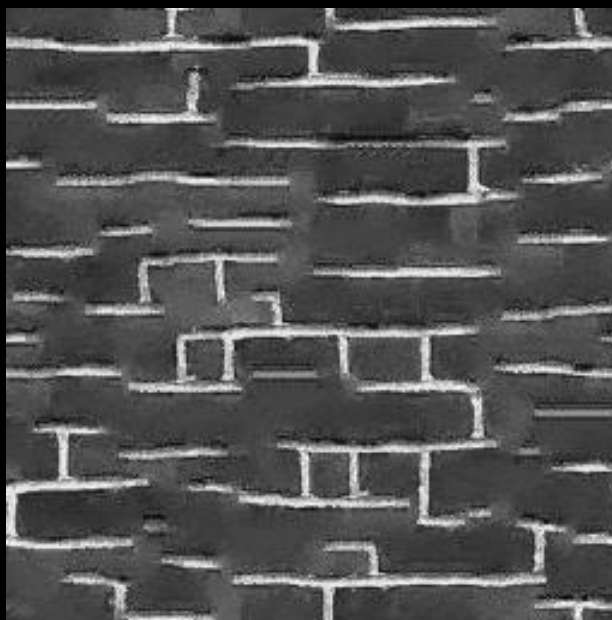
input image



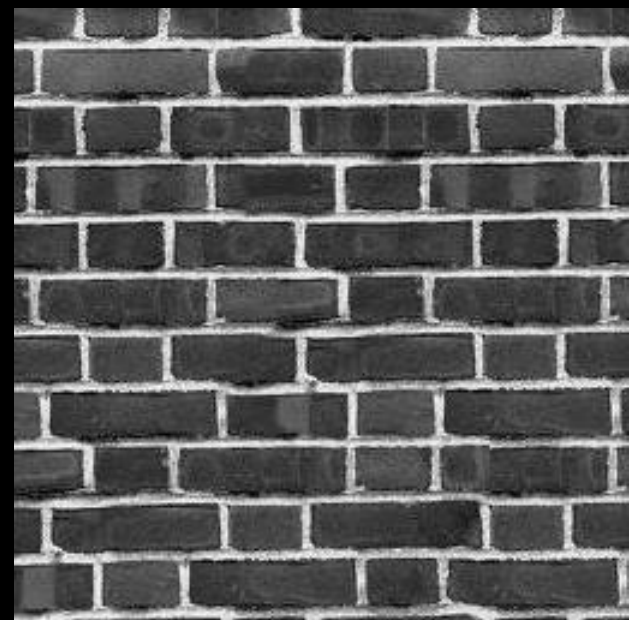
Portilla & Simoncelli



Xu, Guo & Shum



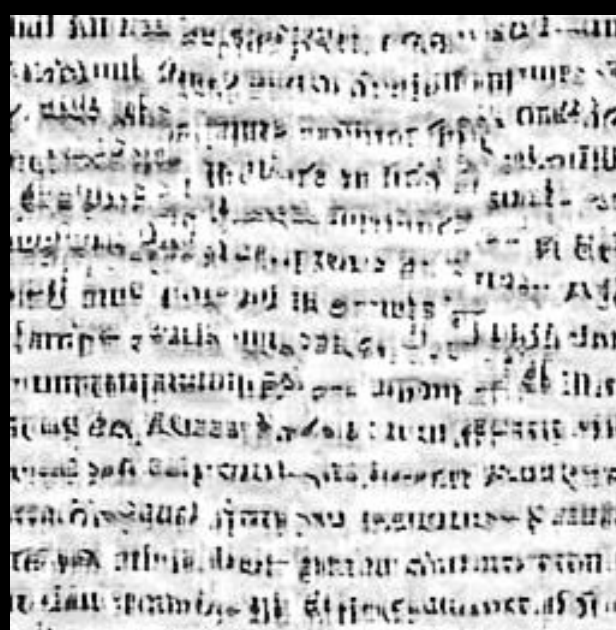
Wei & Levoy



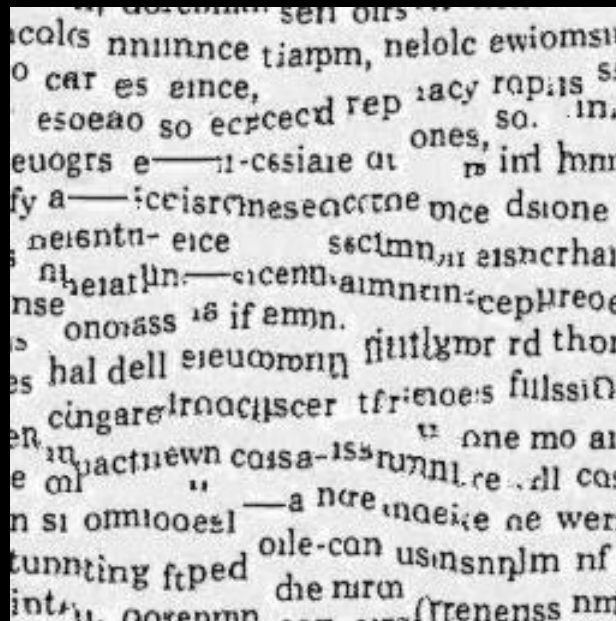
Our algorithm

...id of a visual cortical neuron—the in  
describing the response of that neuro  
ht as a function of position—is perhap  
functional description of that neuron.  
seek a single conceptual and mathem  
describe the wealth of simple-cell recep  
ad neurophysiologically<sup>1-3</sup> and inferred  
especially if such a framework has the  
it helps us to understand the functio  
eeper way. Whereas no generic mo  
ussians (DOG), difference of offset C  
rivative of a Gaussian, higher derivati  
function, and so on—can be expecte  
imple-cell receptive field, we noneth

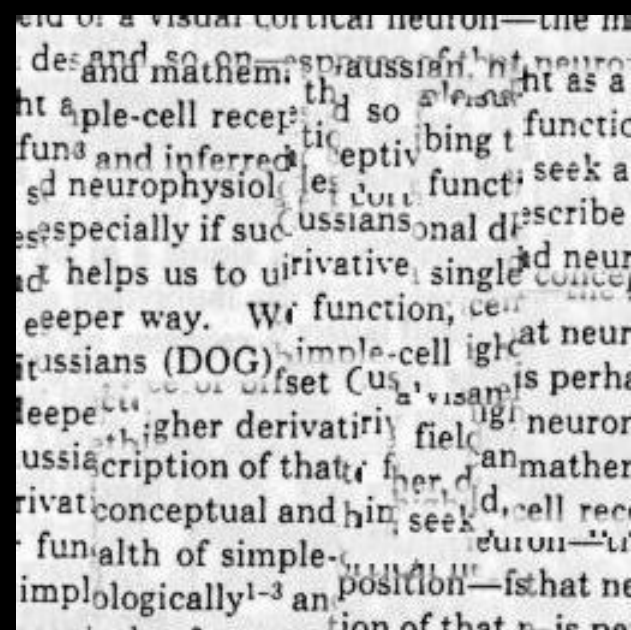
input image



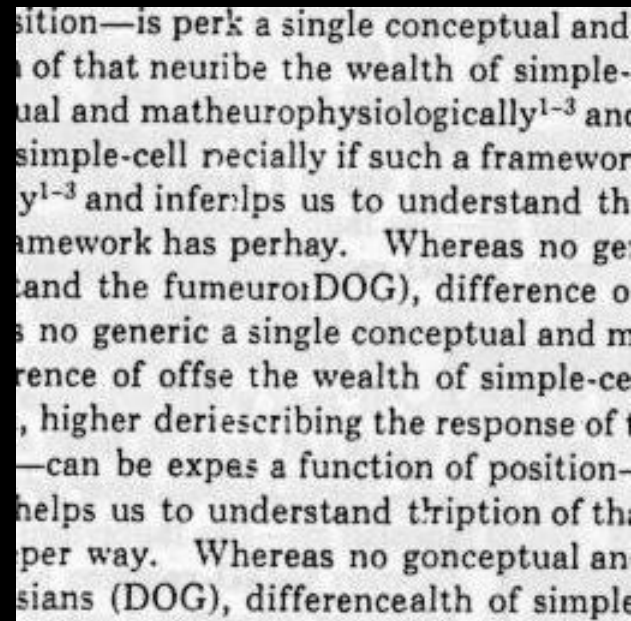
Portilla & Simoncelli



Wei & Levoy



Xu, Guo & Shum



Our algorithm



# Political Texture Synthesis!

## Bush campaign digitally altered TV ad

President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.



Original photograph

# Fill Order



- In what order should we fill the pixels?

# Fill Order



- In what order should we fill the pixels?
  - choose pixels that have more neighbors filled
  - choose pixels that are continuations of lines/curves/edges

# Application: Texture Transfer

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



# Texture Transfer



Constraint

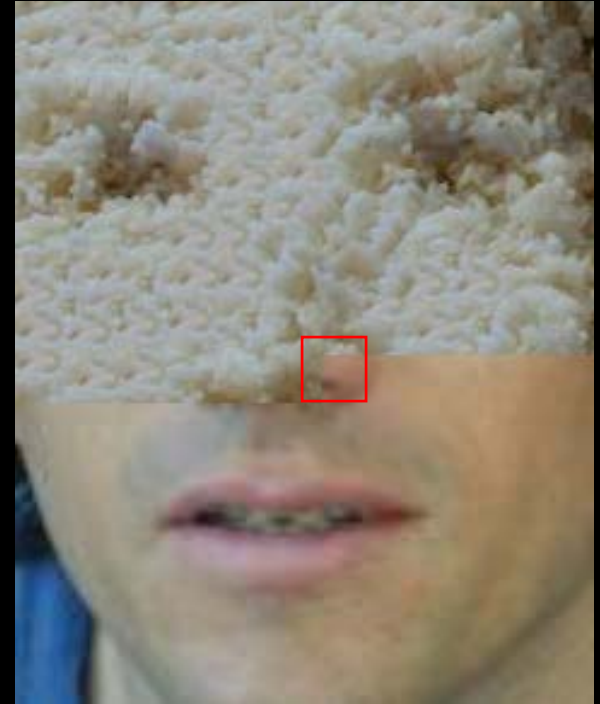


Texture sample



# Texture Transfer

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

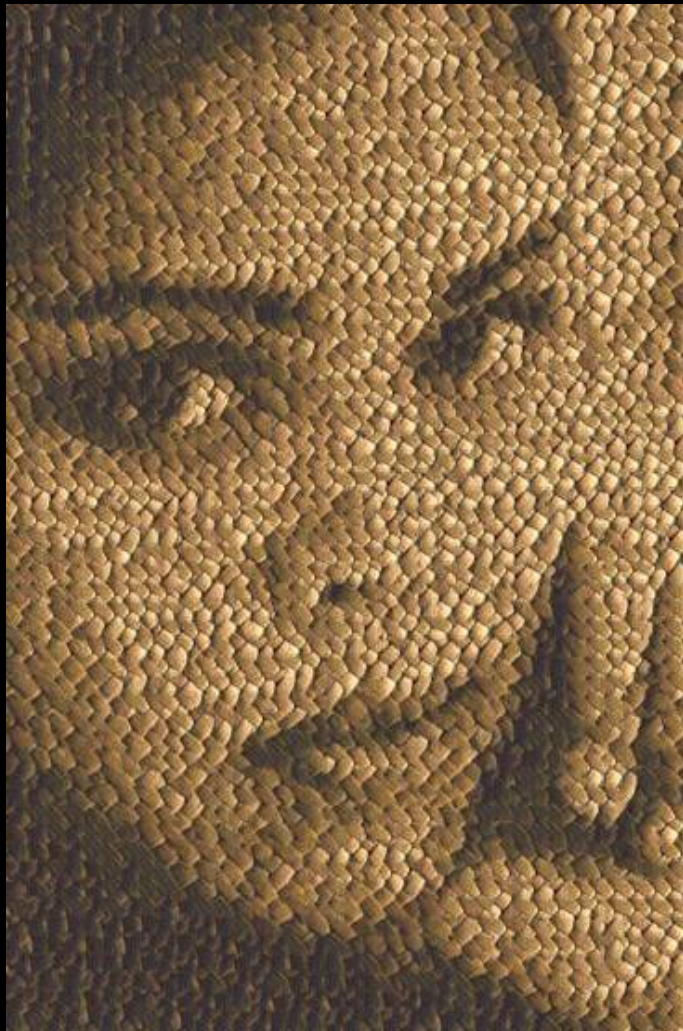


Same as texture synthesis, except an additional constraint:

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









# Image Analogies

Aaron Hertzmann<sup>1,2</sup>

Chuck Jacobs<sup>2</sup>

Nuria Oliver<sup>2</sup>

Brian Curless<sup>3</sup>

David Salesin<sup>2,3</sup>

<sup>1</sup>**New York University**

<sup>2</sup>**Microsoft Research**

<sup>3</sup>**University of Washington**

# Image Analogies



A



A'



B



B'



# Blur Filter



**Unfiltered source ( $A$ )**



**Filtered source ( $A'$ )**



**Unfiltered target ( $B$ )**



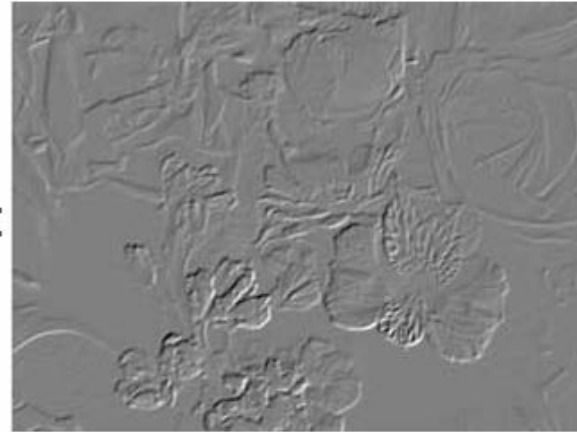
**Filtered target ( $B'$ )**



# Edge Filter



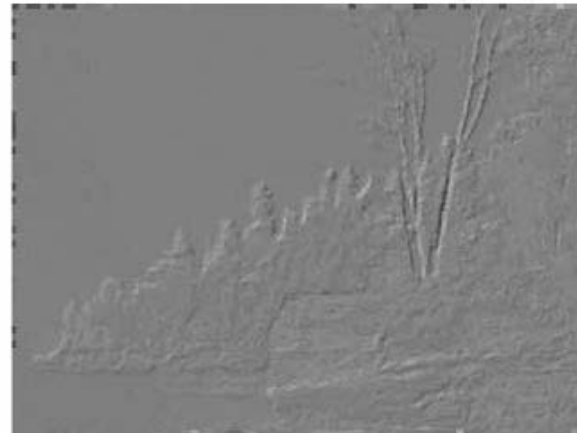
Unfiltered source ( $A$ )



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 ( $B'$ )



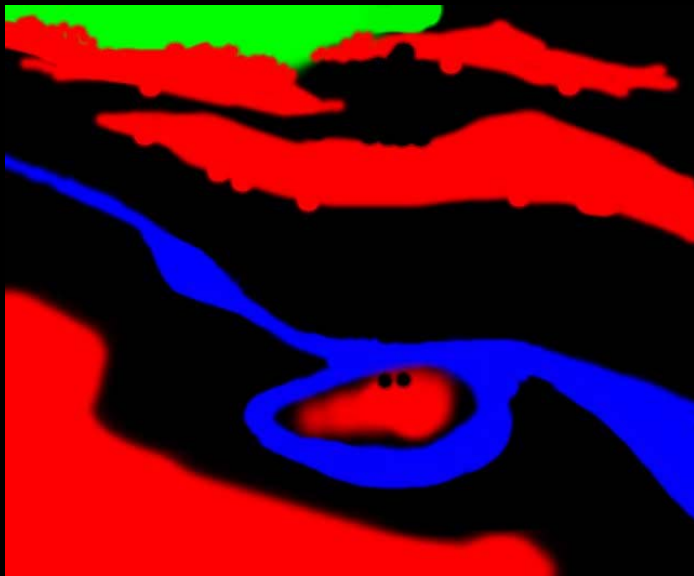
# Texture-by-numbers



A



A'



B



B'

# Super-resolution



A



A'

# Super-resolution (result!)



B



B'

# Video Textures

Arno Schödl

Richard Szeliski

David Salesin

Irfan Essa

Microsoft Research, Georgia Tech

# Our approach

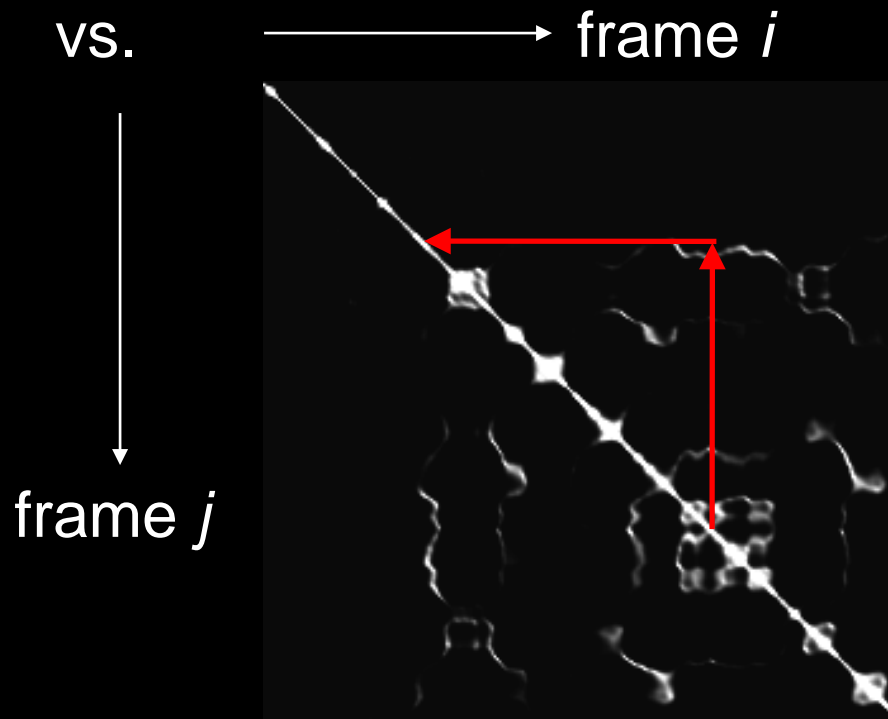


- How do we find good transitions?



# Finding good transitions

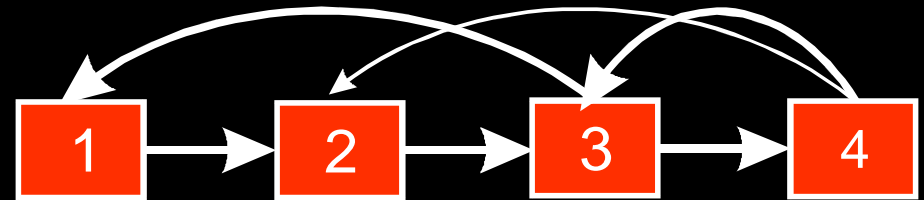
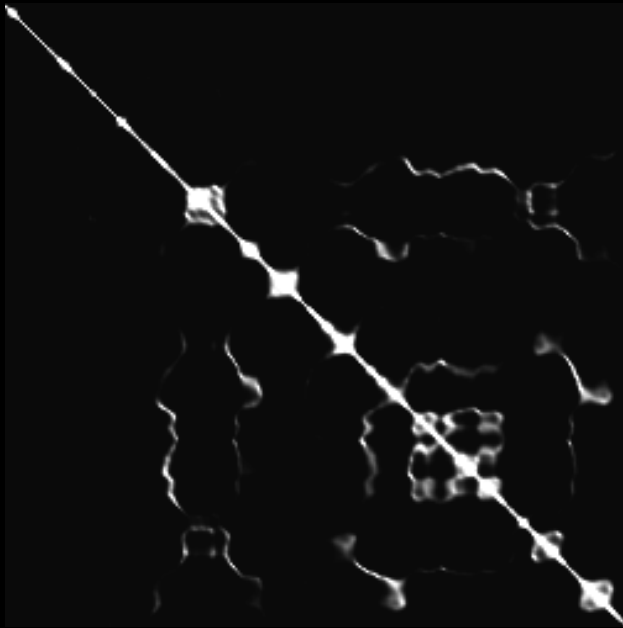
- Compute  $L_2$  distance  $D_{i,j}$  between all frames



Similar frames make good transitions



# Markov chain representation



Similar frames make good transitions

# Example

