

Texture Synthesis and Hole-Filling



Computational Photography
Derek Hoiem, University of Illinois

Project 1

- Will have links up today
- Grading done in next day or two

Next section: The digital canvas



Cutting and pasting objects,
filling holes, and blending



Image warping and object
morphing

Today's Class

- Texture synthesis and hole-filling



Texture

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



radishes



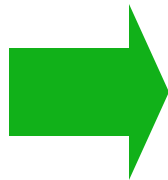
rocks



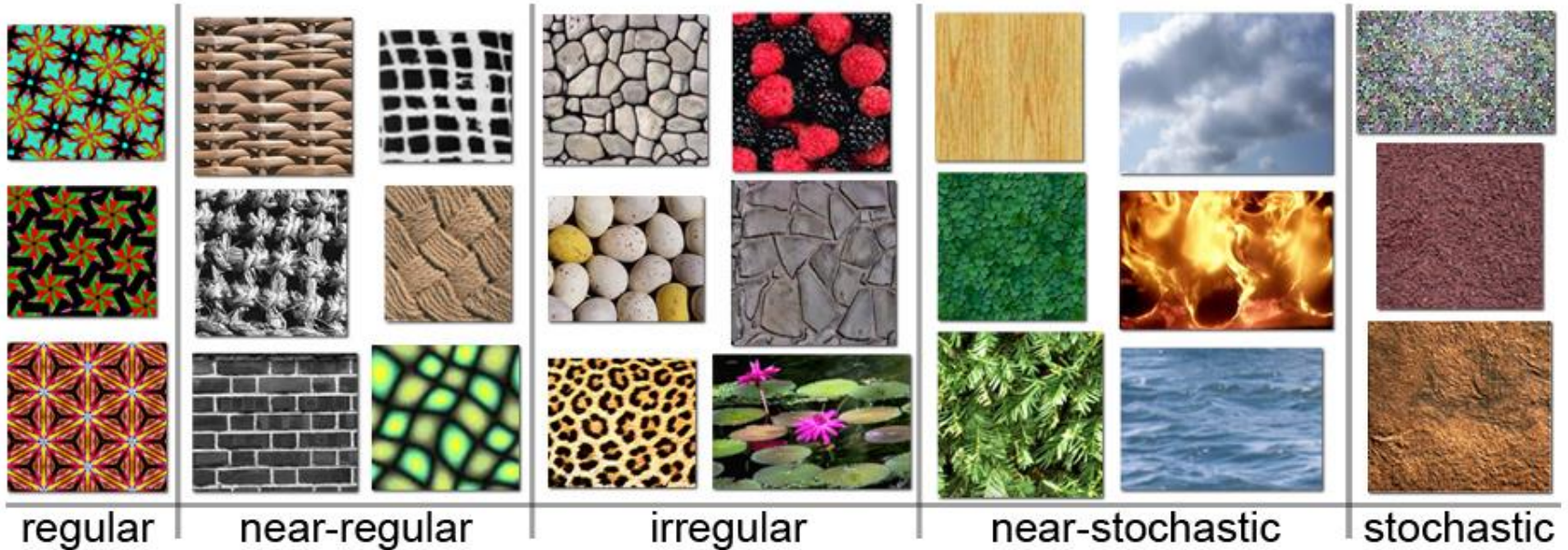
yogurt

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

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



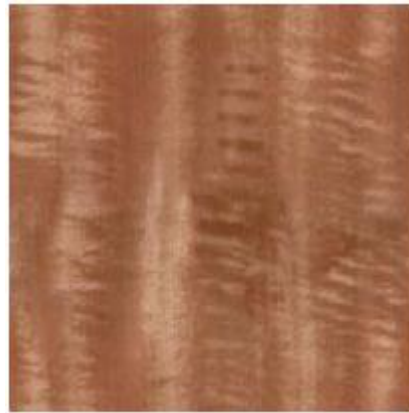
- D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95*.
- E. P. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *ICIP 1998*.

One idea: Build Probability Distributions

But it (usually) doesn't work

- Probability distributions are hard to model well

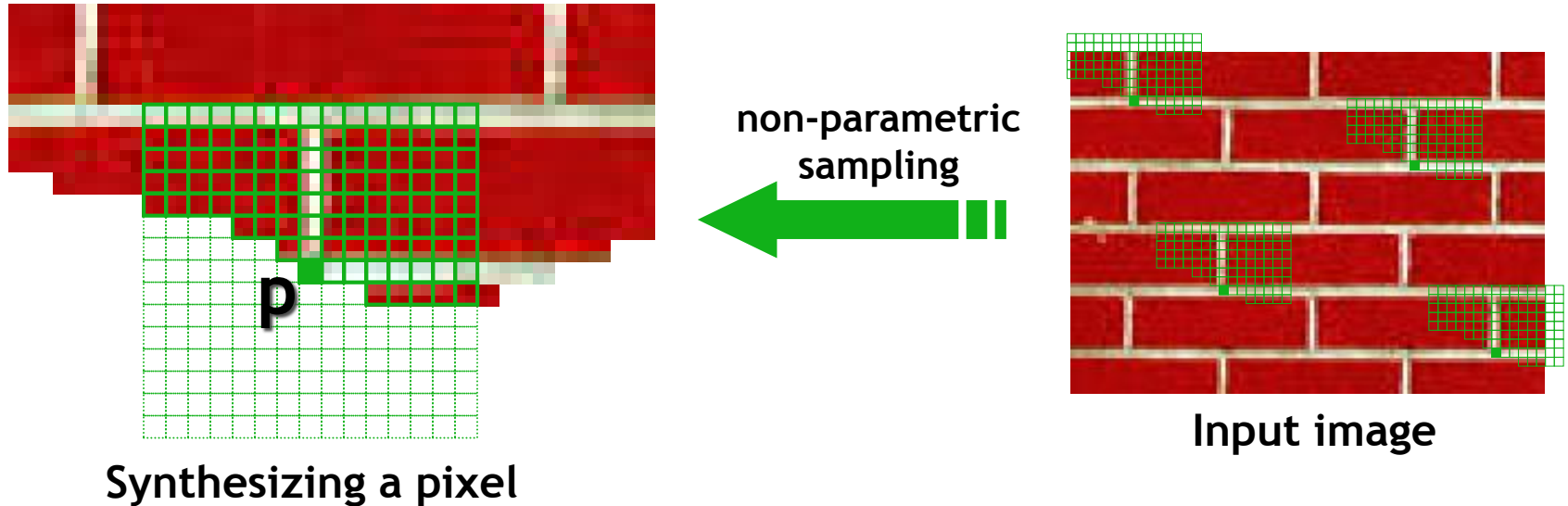
Input



Synthesized



Another idea: Sample from the image



- Assuming Markov property, compute $P(\mathbf{p} | N(\mathbf{p}))$
 - 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

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

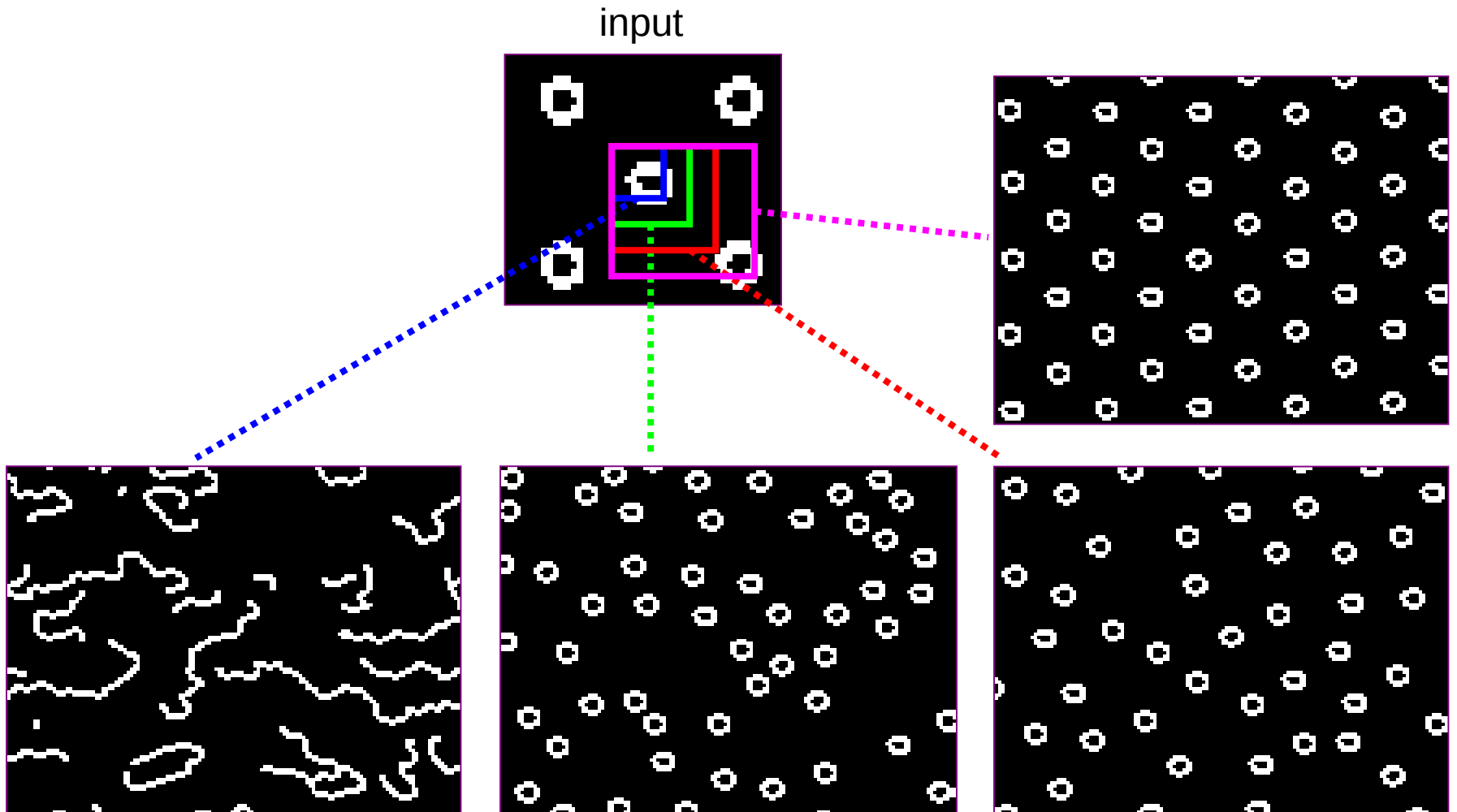
“I spent an interesting evening recently
with a grain of salt.”

(example from fake single.net user [Mark V Shaney](#))

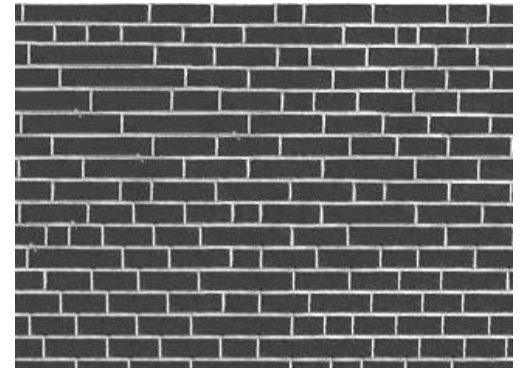
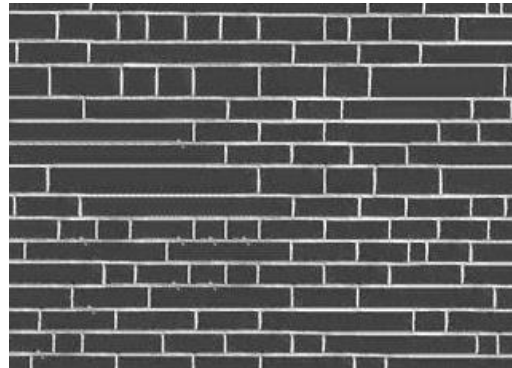
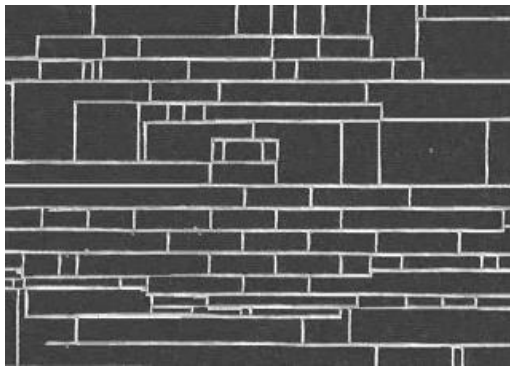
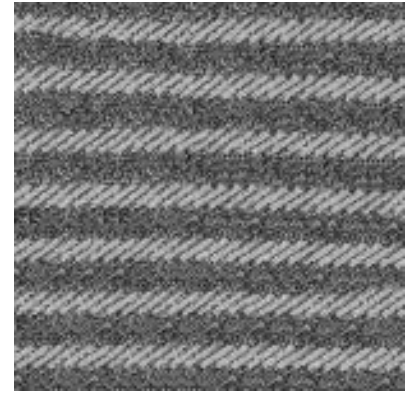
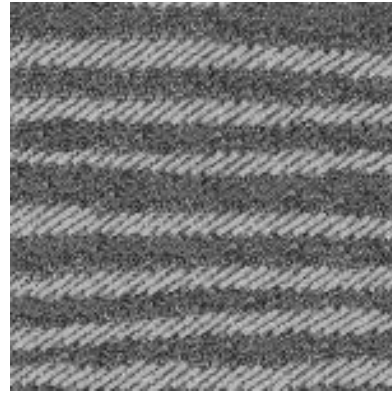
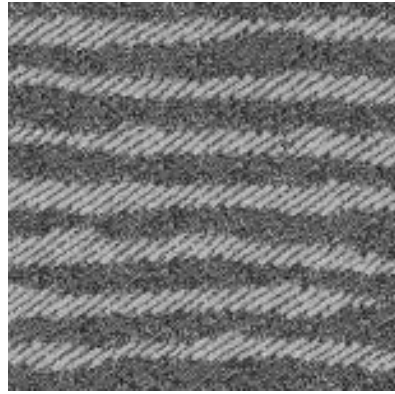
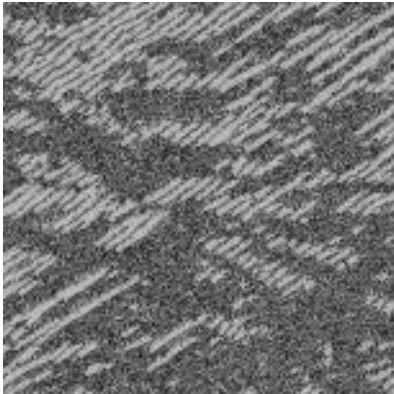
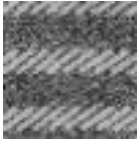
Details

- 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
- How big should the patches be?

Size of Neighborhood Window



Varying Window Size



Increasing window size

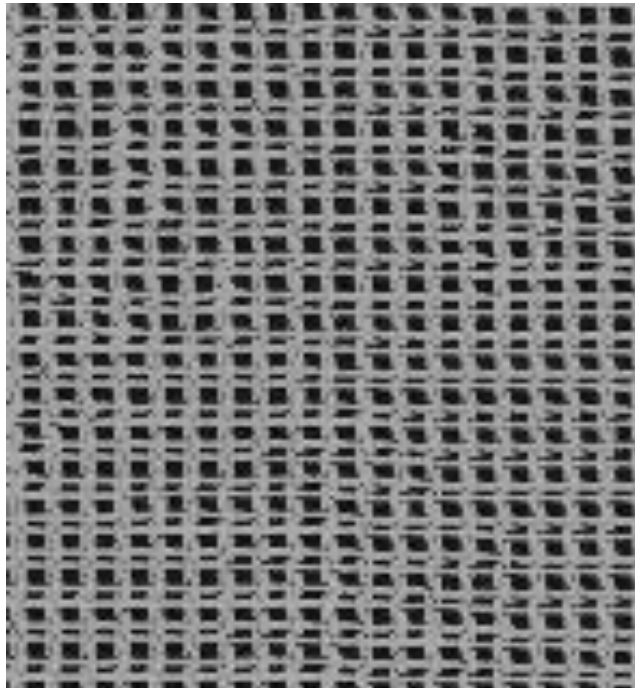
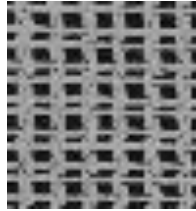


Texture synthesis algorithm

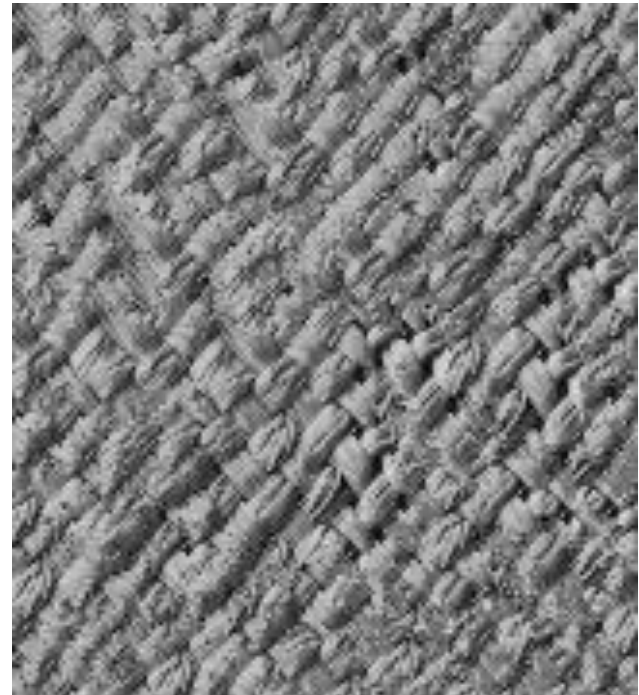
- While image not filled
 1. Get unfilled pixels with filled neighbors, sorted by number of filled neighbors
 2. For each pixel, get top N matches based on visible neighbors
 - Patch Distance: Gaussian-weighted SSD
 3. Randomly select one of the matches and copy pixel from it

Synthesis Results

french canvas



rafia weave

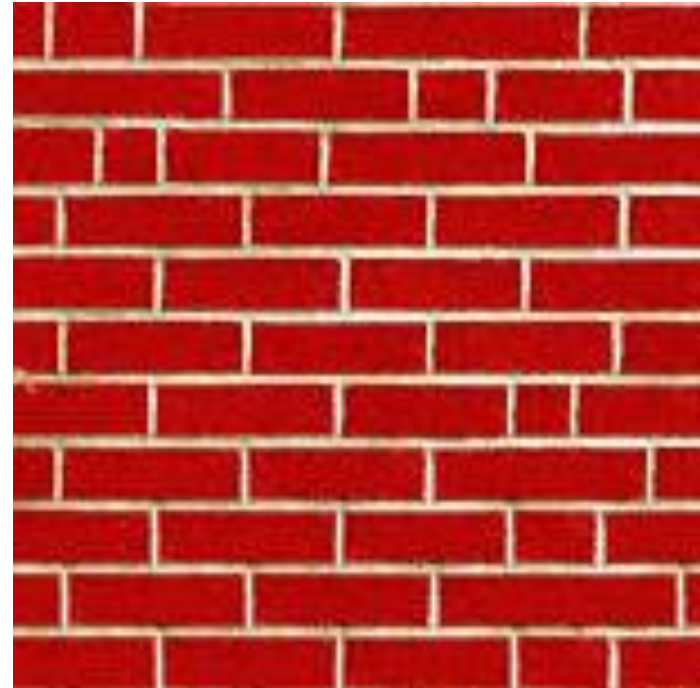
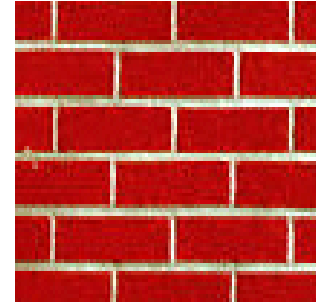


More Results

white bread



brick wall



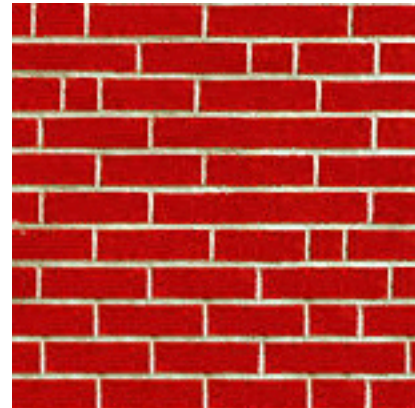
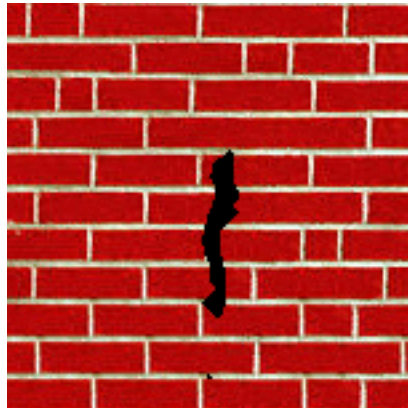
Homage to Shannon

oming in the unsensational
r Dick Gephardt was fair
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
ardt began, patiently obs
s, that the legal system k
g with this latest tanger

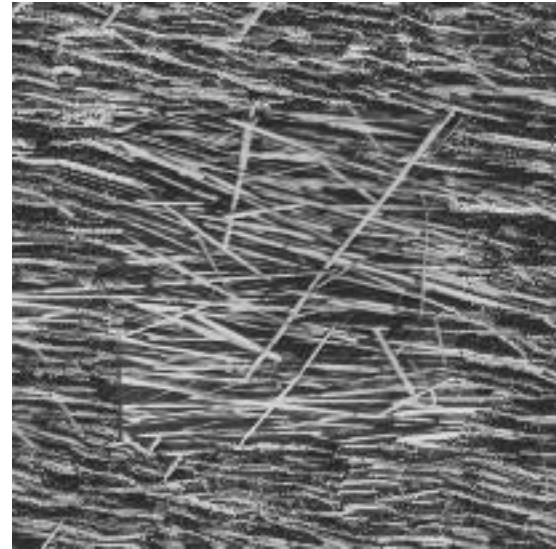
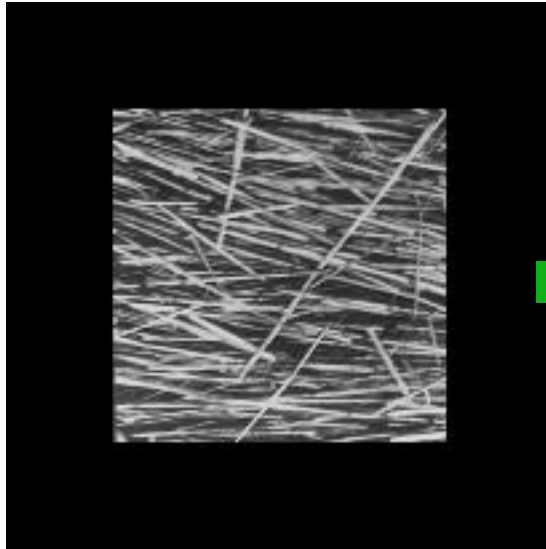


ithaim. them. "Whnephartfe lartifelintomimen
fel ck Clirtioout omaim thartfelins.f out s anetc
the ry onst wartfe lck Gephtoomimeationl sigak
Chiooufit Clinut Cll riff on. hat's yordn, parut tly
ons youtonsteht wasked, paim t sahe loo riff on l
nskoneploourtfeas leil A nst Clit, "Wleontongal s
k Clirtioouirtfepe.ong pme abegal fartfenstemem
tiensteneltorydt telemephminsverdt was agemer
ff ons artientont Cling peme as artfe atich, "Boui s
aal s fartfelt sig pedrtldt ske abounutie aboutioo
tfaonewwas yow abowonthardt thatins fain, ped, '
ains. them, pabout wasy arfint couitly d, l n A h
le emthrdngbooreme agas fa bontinsyst Clinut
ory about continst Clipeopinst 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 Clihy abon al coountha.emungait tfoun
Vhe loocrystan loontieph. intly on, tieoplegatick C
aul fatiecontly atie Diontiomt wal s f tbegae ener
nthahsat's enenhhhoas fan. "intchthorv abonsu

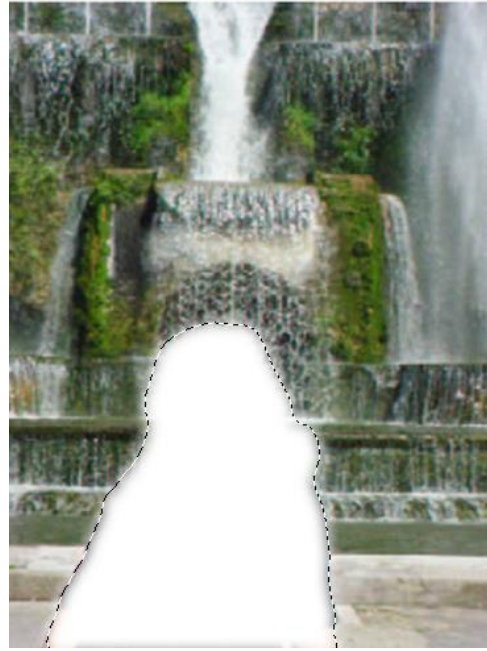
Hole Filling



Extrapolation



In-painting natural scenes



Key idea: Filling order matters

In-painting Result



Image with Hole



Raster-Scan Order



Onion-Peel
(Concentric Layers)



Gradient-Sensitive
Order

Filling order

Fill a pixel that:

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



Comparison



Original



With Hole



Onion-Ring Fill



Criminisi

Comparison



a



b



Concentric Layers

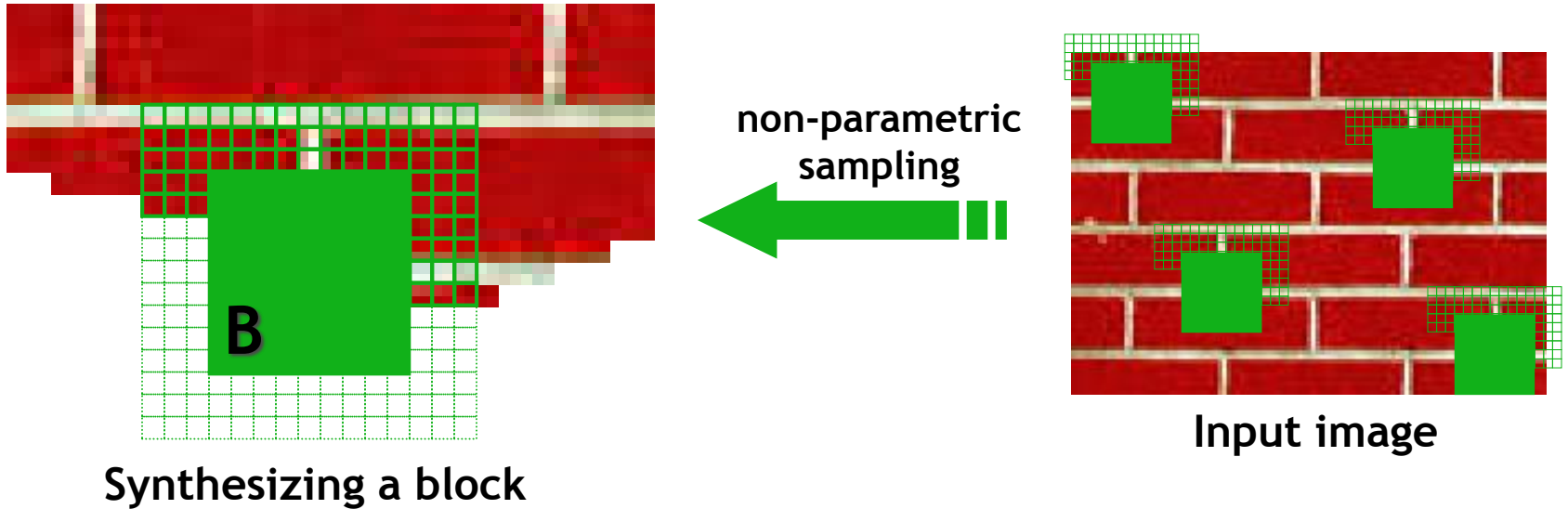


Gradient Sensitive

Summary

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

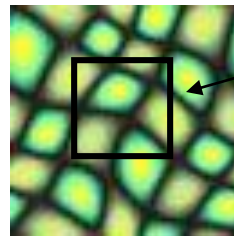
Image Quilting [Efros & Freeman 2001]



- Observation: neighbor pixels are highly correlated

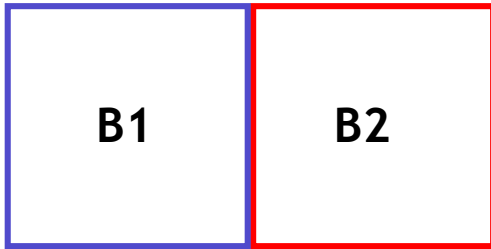
Idea: unit of synthesis = block

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

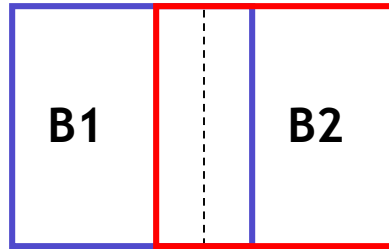


block

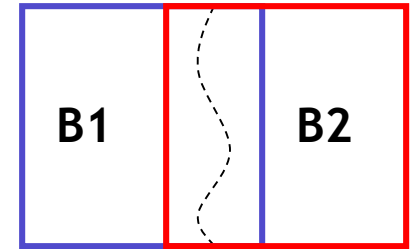
Input texture



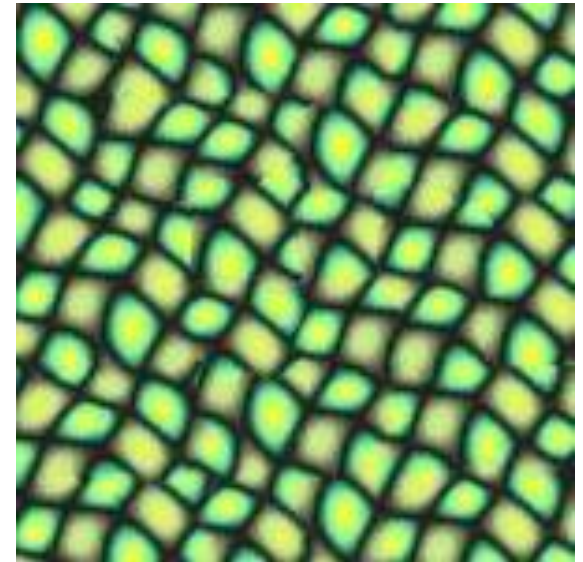
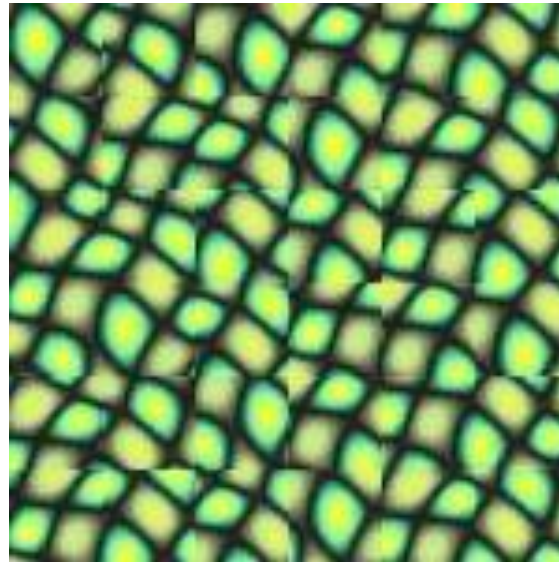
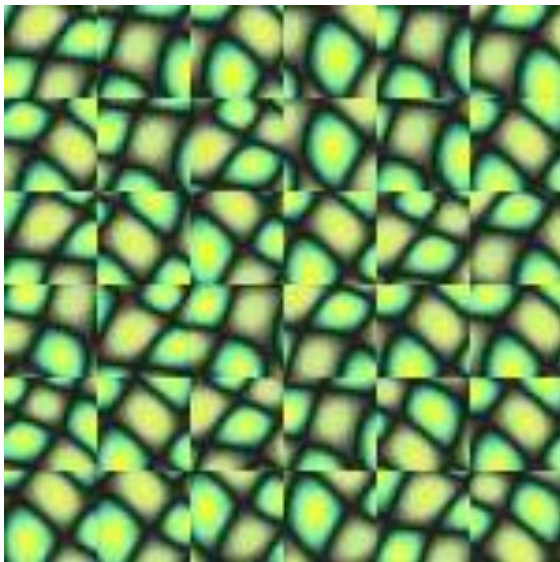
Random placement
of blocks



Neighboring blocks
constrained by overlap

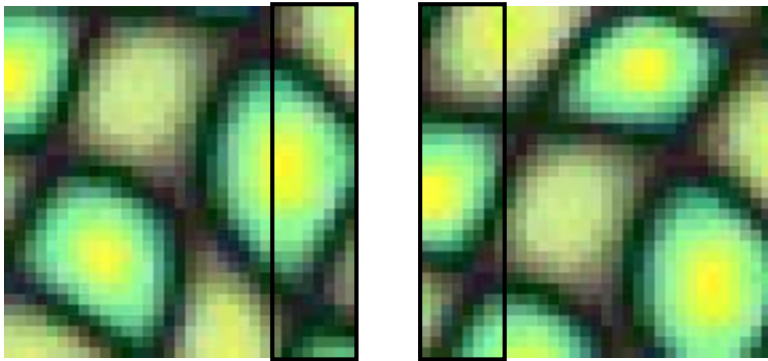


Minimal error
boundary cut

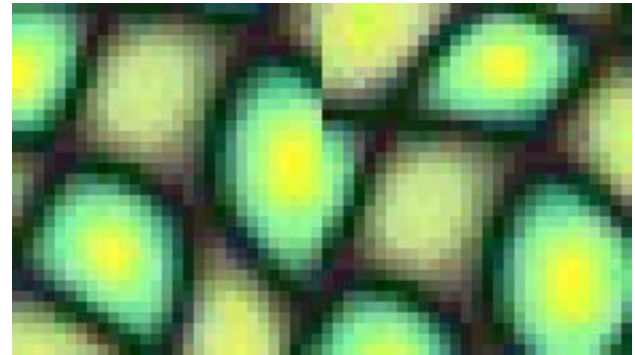


Minimal error boundary

overlapping blocks

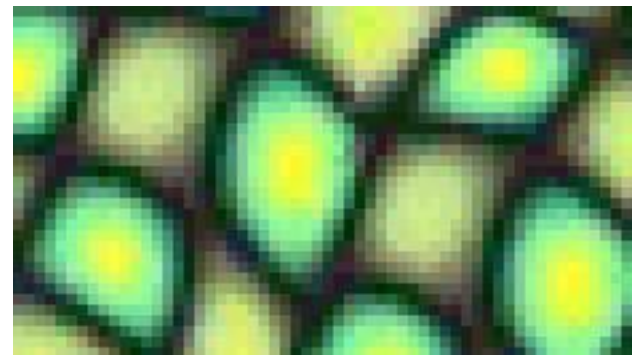


vertical boundary



A diagram illustrating the calculation of the overlap error. It shows two vertical blocks of the cell image, one shifted relative to the other. A large right square bracket groups them, with a minus sign between them, followed by a superscript 2. An equals sign follows, leading to a vertical strip of the image. A red line is drawn along the boundary of this strip, representing the error.

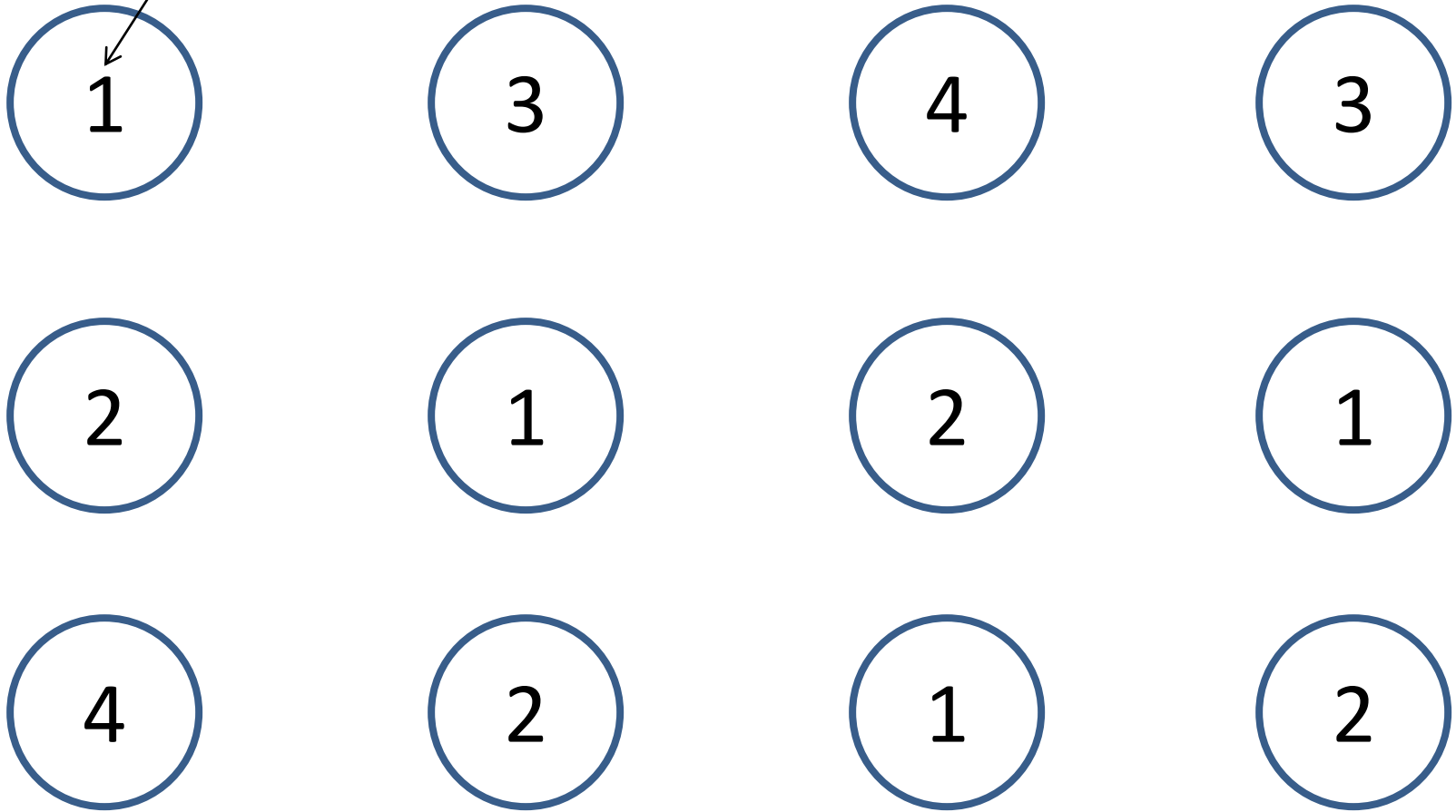
overlap error



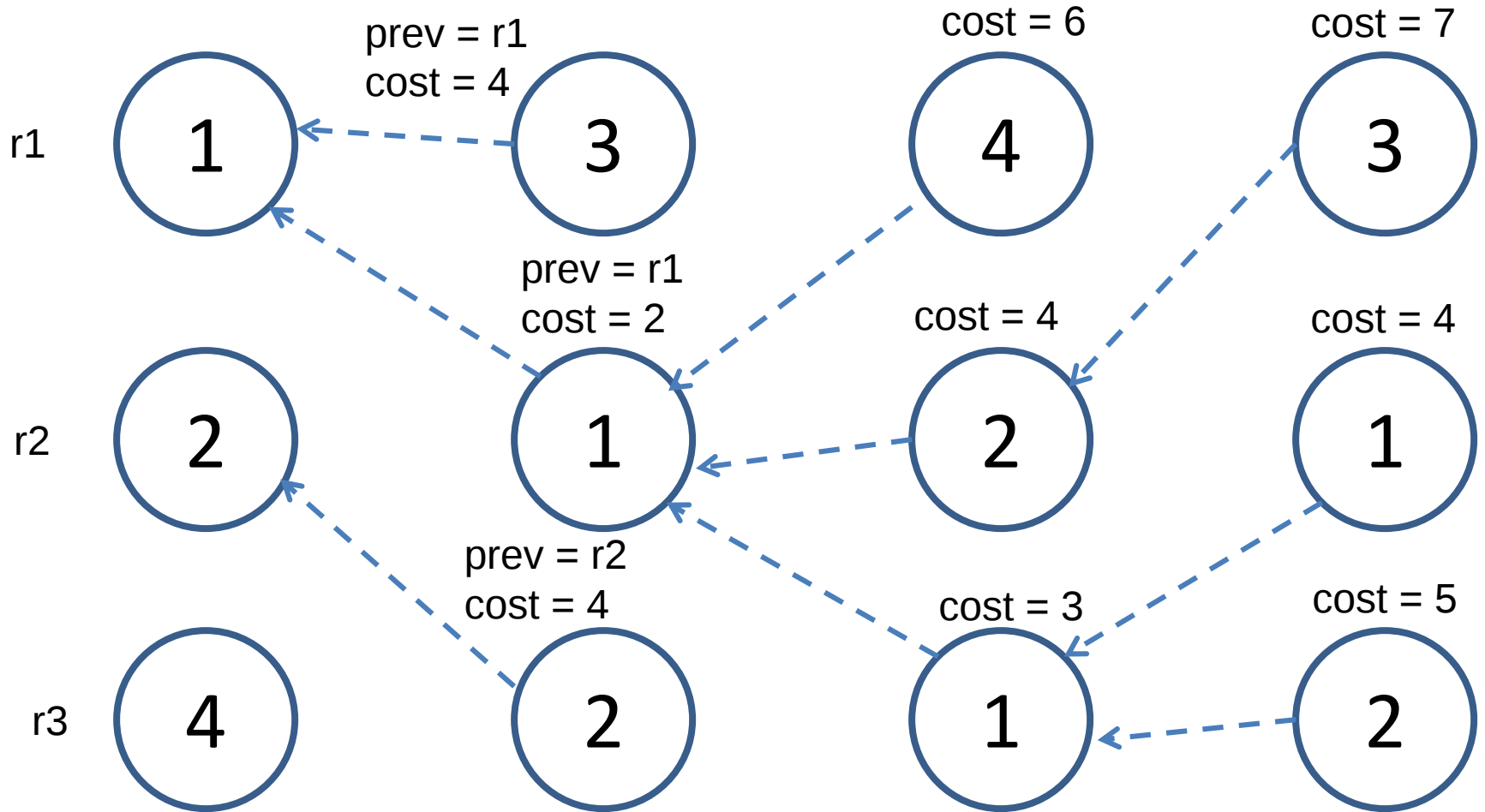
min. error boundary

Solving for Minimum Cut Path

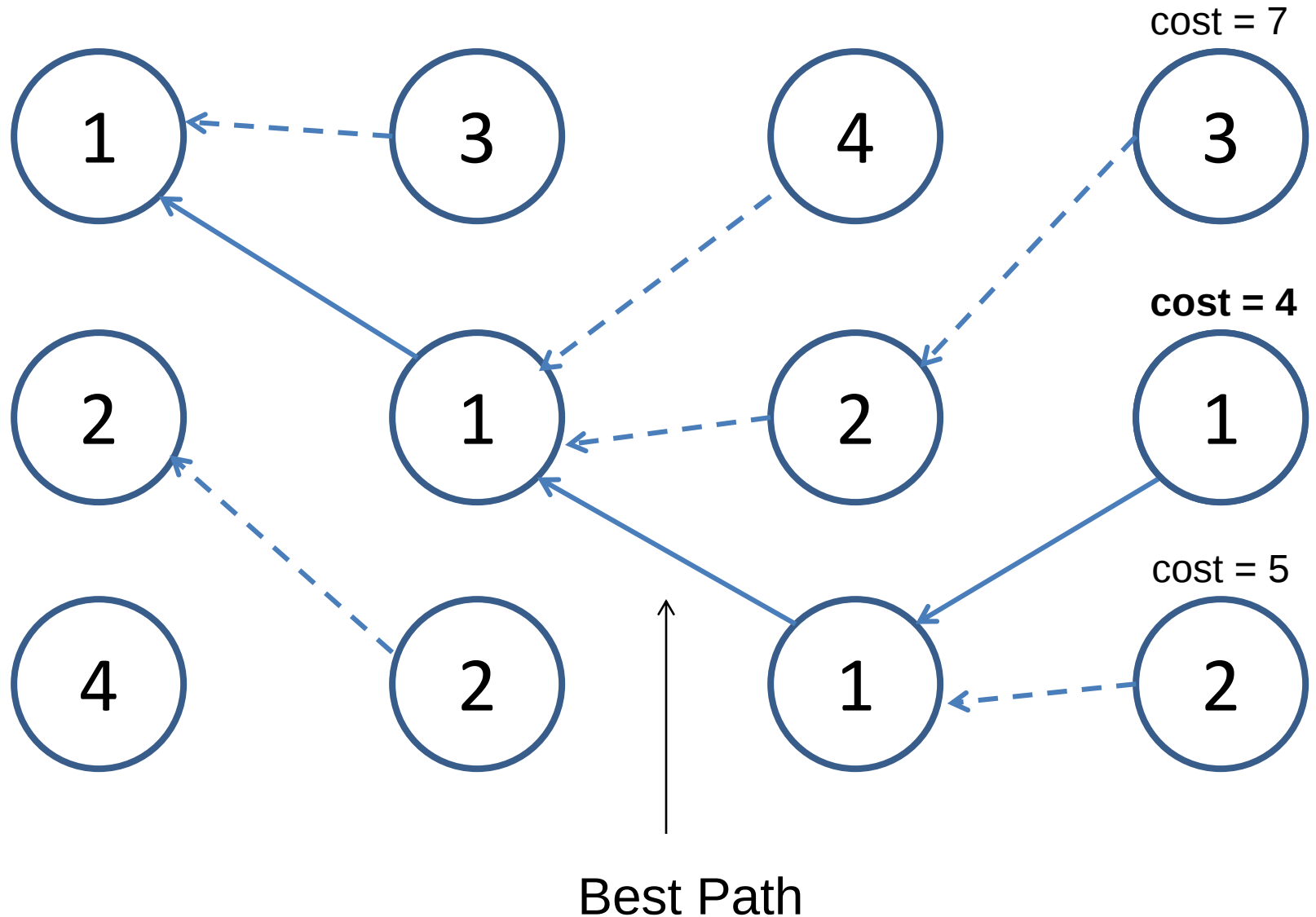
Cost of a cut through this pixel



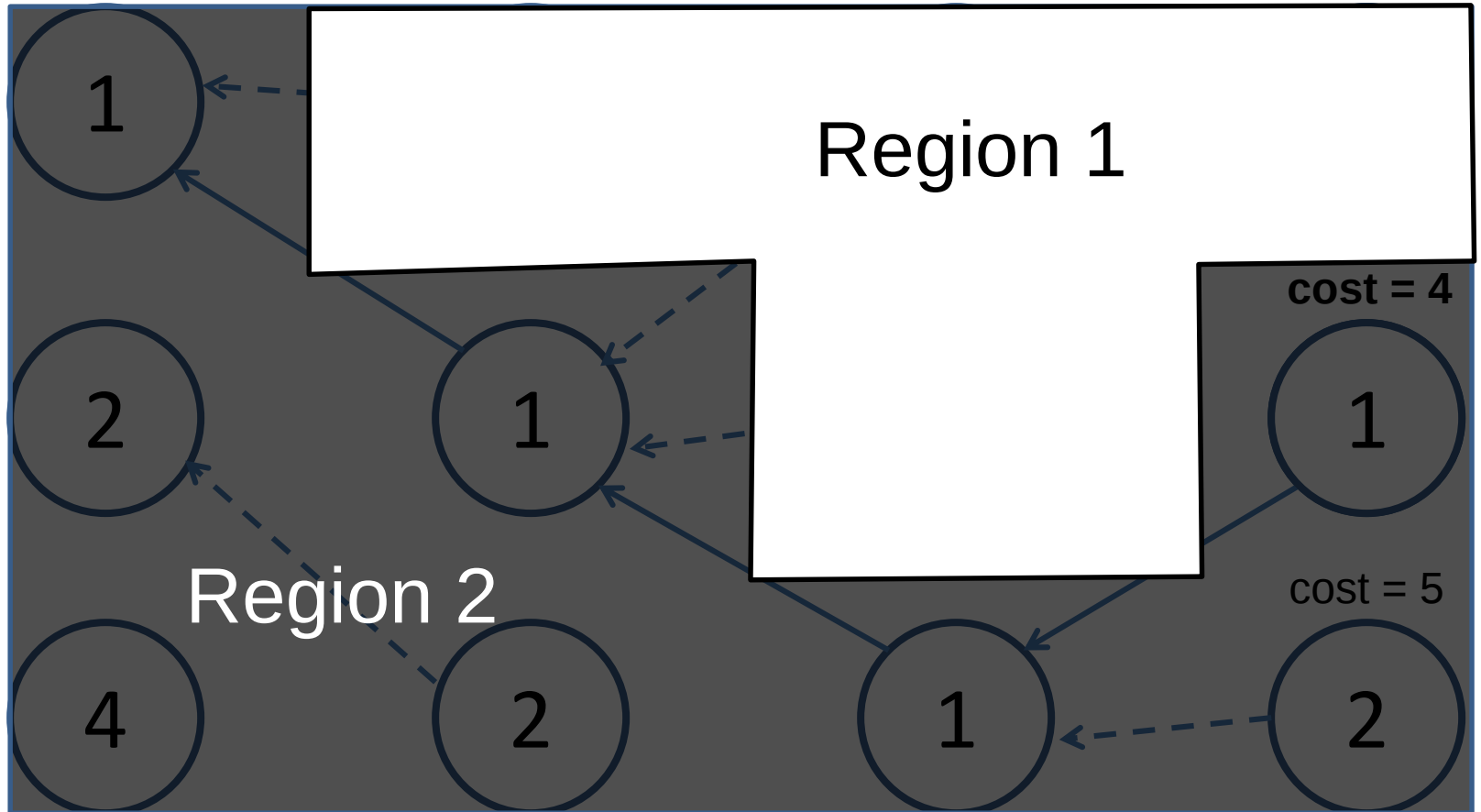
Solving for Minimum Cut Path



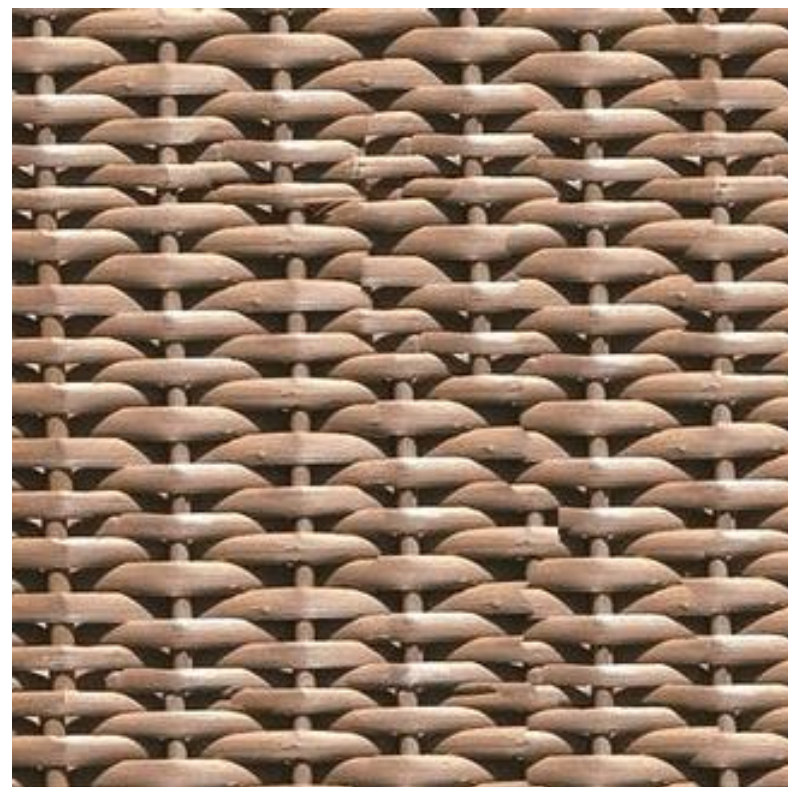
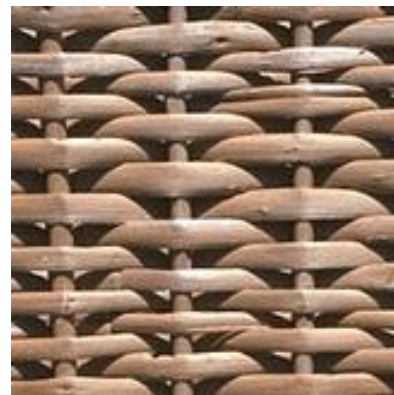
Solving for Minimum Cut Path

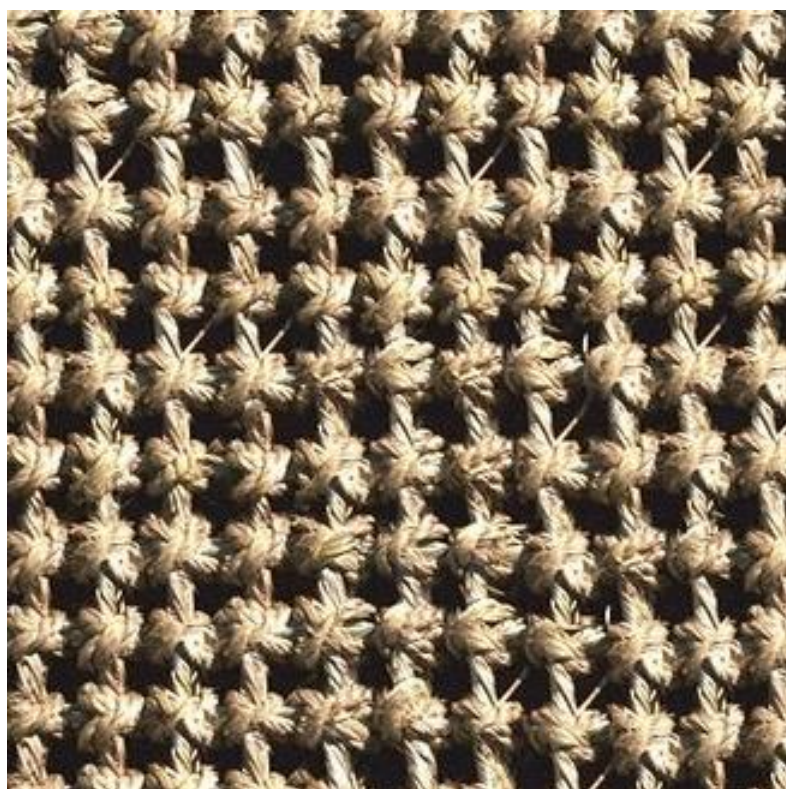


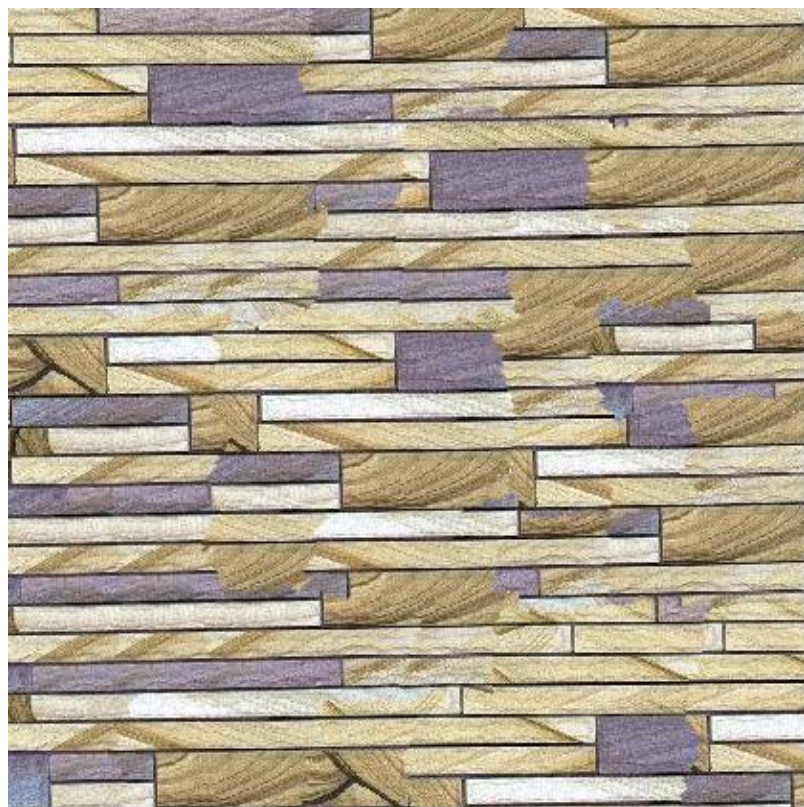
Solving for Minimum Cut Path



Mask Based on Best Path

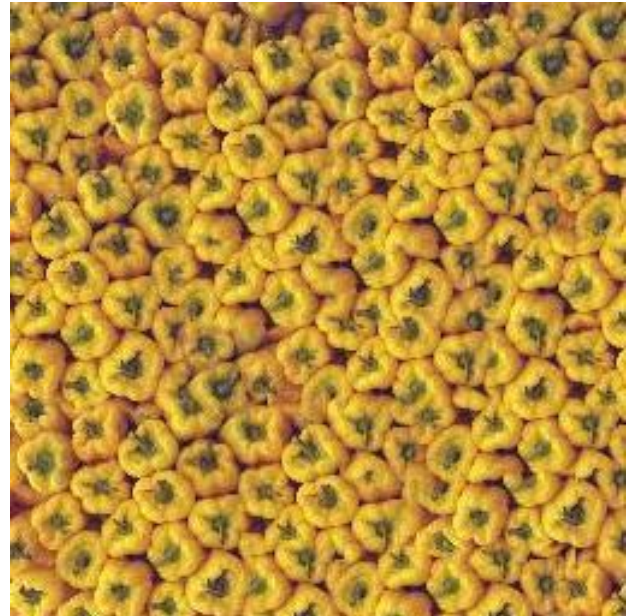
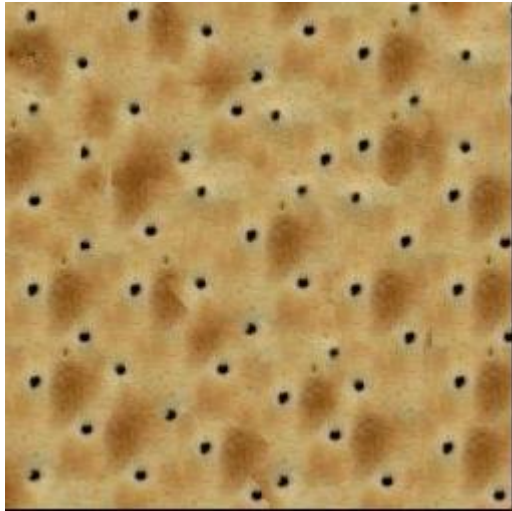
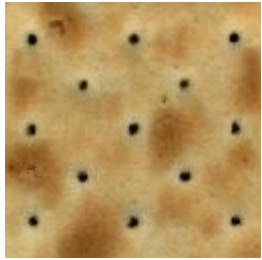


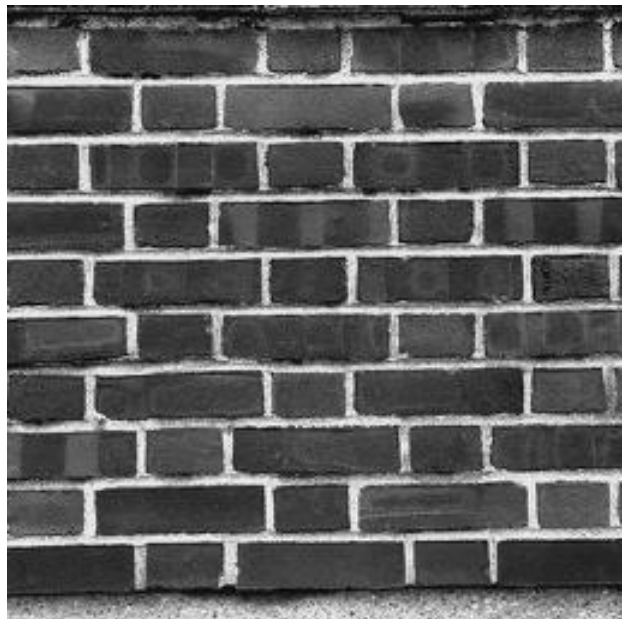








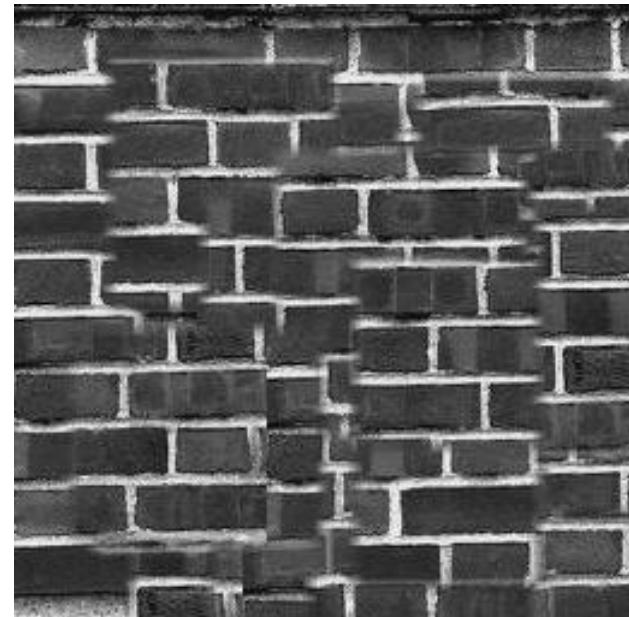




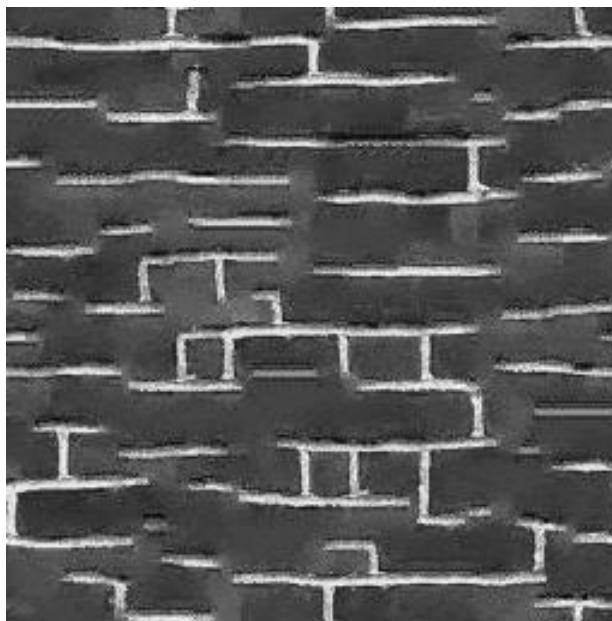
input image



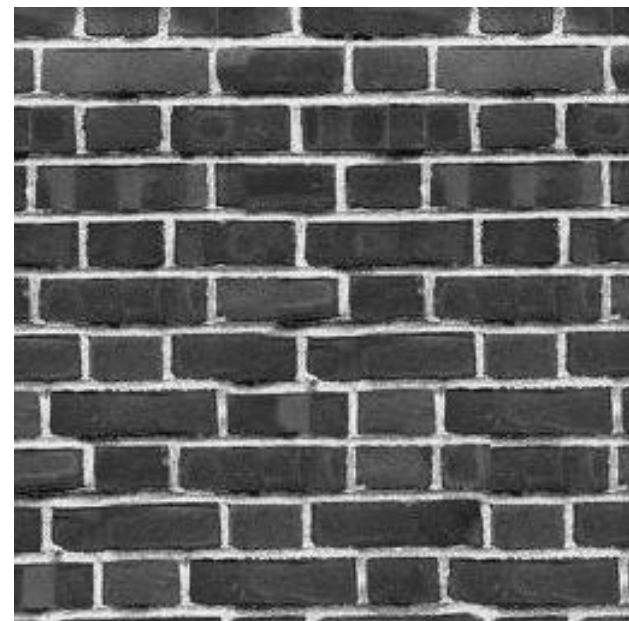
Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy



Quilting

... 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
and neurophysiologically¹⁻³ 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
simple-cell receptive field, we noneth

input image

... 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
and neurophysiologically¹⁻³ 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
simple-cell receptive field, we noneth

Portilla & Simoncelli

... 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
and neurophysiologically¹⁻³ 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
simple-cell receptive field, we noneth

Wei & Levoy

... 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
and neurophysiologically¹⁻³ 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
simple-cell receptive field, we noneth

Xu, Guo & Shum

... 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
and neurophysiologically¹⁻³ 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
simple-cell receptive field, we noneth

Quilting

Political Texture Synthesis!

Bush campaign digitally altered TV ad

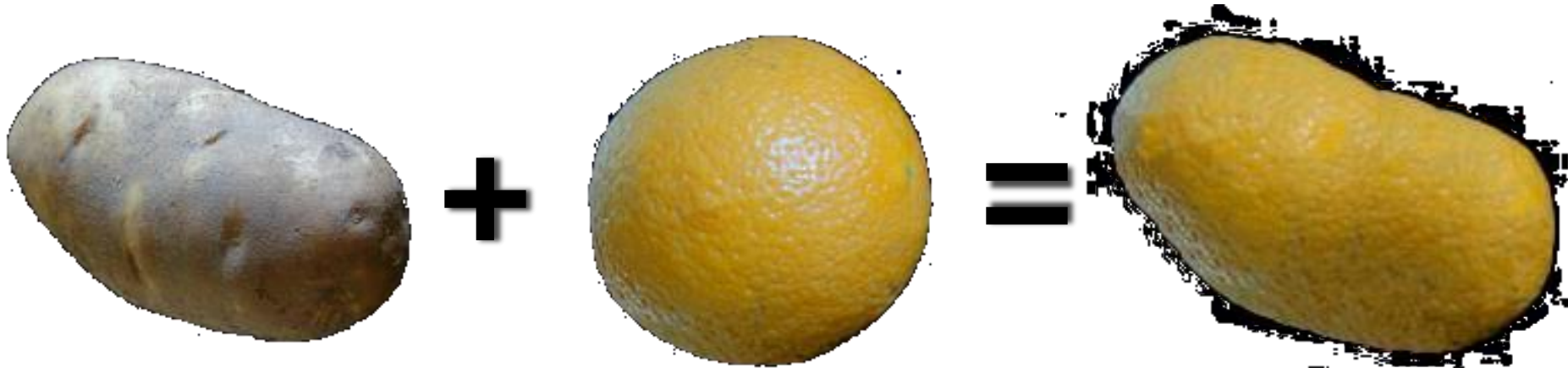
President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.



Texture Transfer

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



Texture Transfer



Constraint

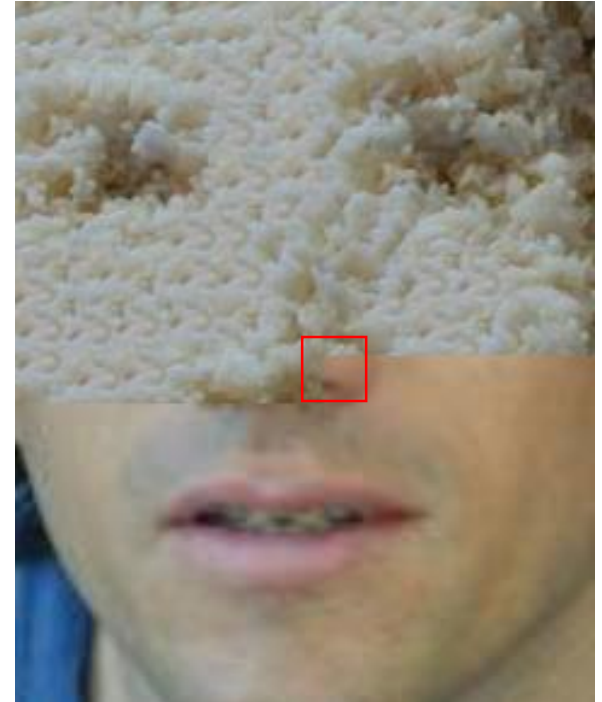


Texture sample



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. Patches from texture should correspond to patches from constraint in some way. Typical example: blur luminance, use SSD for distance



source texture



target image

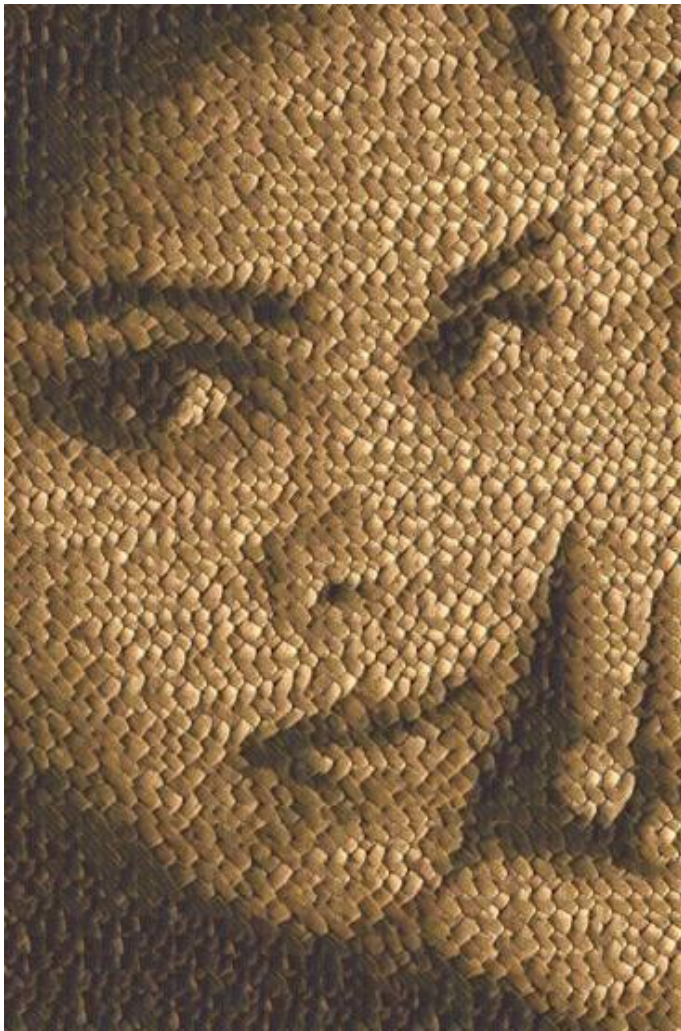


correspondence maps

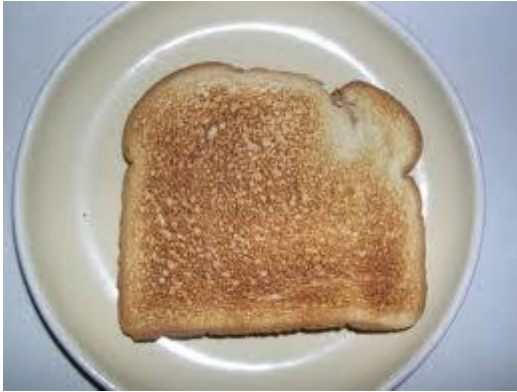


texture transfer result





Making sacred toast



+



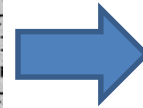
?



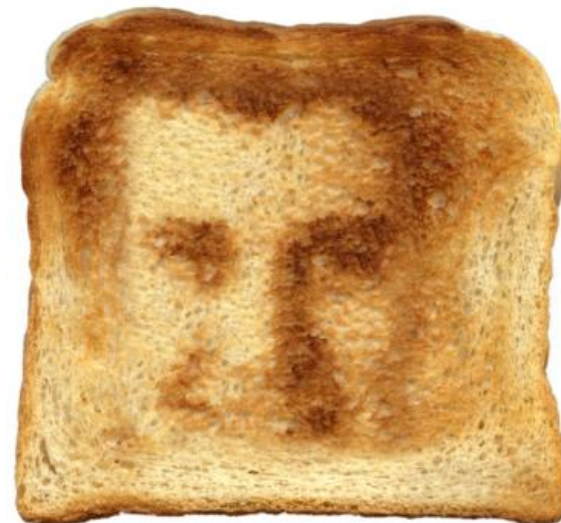
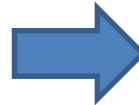
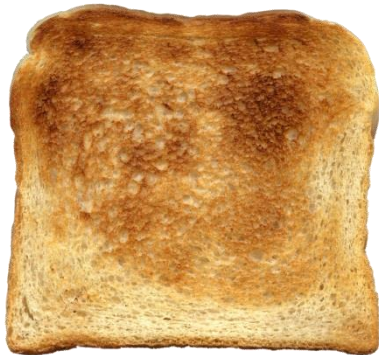
Project 2: texture synthesis and transfer

- <http://courses.engr.illinois.edu/cs498dh3/projects/quilting/ComputationalPhotographyProjectQuilting.html>
- Note: this is significantly more challenging than the first project

ut it becomes harder to lau
ound itself, at "this daily
ving rooms," as House Der
scribed it last fall. He fai
at he left a ringing question
ore years of Monica Lewin
inda Tripp?" That now seen
Political comedian Al Fran
ext phase of the story will



und itself, at this it becomes harder itself, at this o
ing rooms," as Hound itself, at "thisrooms," as Hous
cribed it last falling rooms," as Hoved it last fall. H
he left a ringing quibed it last fall. left a ringing qu
re years of Monica le left a ringing years of Monica L
da Tripp?" That noe years of Monic Tripp?" That now
olitical comedian ida Tripp?" That ntical comedian Al
ns," as Hoitself, at "this dre years of Monicaelf, at "
t last fal rooms," as Housda Tripp?" That norms," as
a ringing ed it last fall. He itical comedian A't last fa
of Moniceft a ringing ques "this dairooms," as Hous
p?" That rears of Monica Las Housed it last fall. F
comes hardins dairboms," as fall. He left a ringing qu
tself, at "t House ed it last fall. He years of Monica l
orns," as fall. He fft a ringing questTripp?" That nov
d it last faze years of Monica vica Les of Monicdiangir
ft a ringinda Tripp?" That nat now so?" That's of Mon
rs of Moolitical Comediandian Al Fcomediapp?" Tha



Texture Synthesis and Transfer Recap



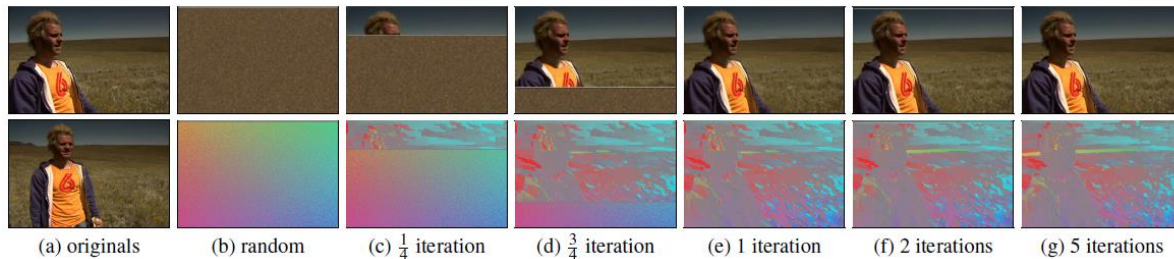
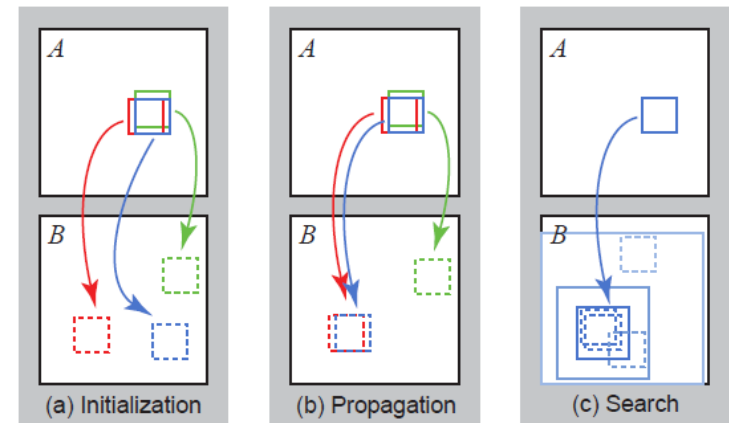
For each overlapping patch in the output image

1. Compute the cost to each patch in the sample
 - Texture synthesis: this cost is the SSD (sum of square difference) of pixel values in the overlapping portion of the existing output and sample
 - Texture transfer: cost is $\alpha * SSD_{overlap} + (1 - \alpha) * SSD_{transfer}$ The latter term enforces that the source and target correspondence patches should match.
2. Select one sample patch that has a small cost
3. Find a cut through the left/top borders of the patch based on overlapping region with existing output
 - Use this cut to create a mask that specifies which pixels to copy from sample patch
4. Copy masked pixels from sample image to corresponding pixel locations in output image

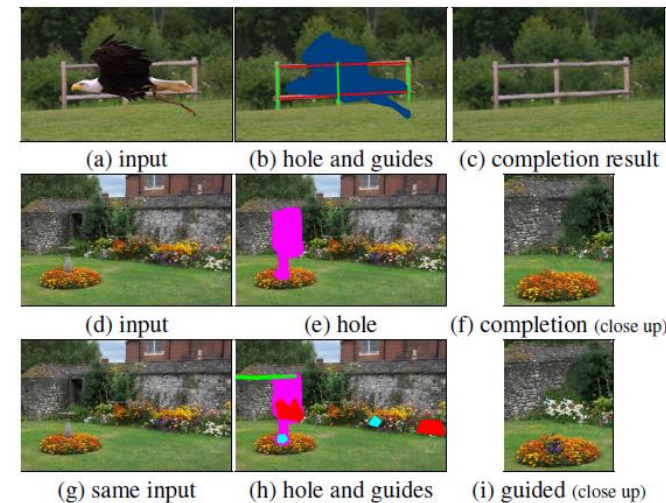
PatchMatch

More efficient search:

1. Randomly initialize matches
2. See if neighbor's offsets are better
3. Randomly search a local window for better matches
4. Repeat 3, 4 across image several times



Reconstructing bottom-left image with patches from top-left image



Applications to hole-filling, retargeting; constraints can guide search

Related idea: Image Analogies



A

⋮



A'



B

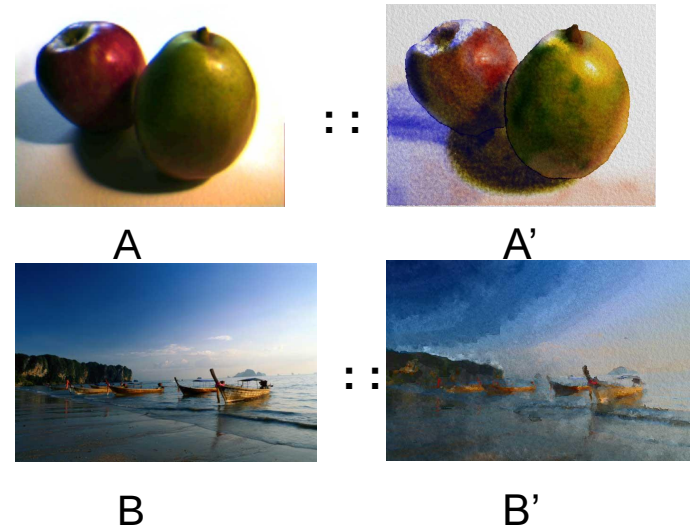
⋮



B'



Image analogies



- Define a similarity between A and B
- For each patch in B:
 - Find a matching patch in A, whose corresponding A' also fits in well with existing patches in B'
 - Copy the patch in A' to B'
- Algorithm is done iteratively, coarse-to-fine

Blur Filter



Unfiltered source (A)



Filtered source (A')



Unfiltered target (B)

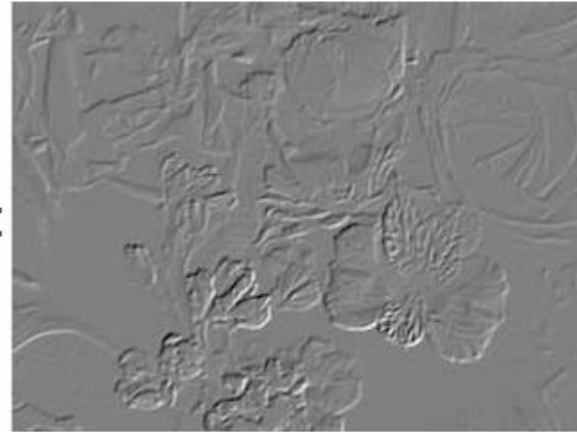


Filtered target (B')

Edge Filter



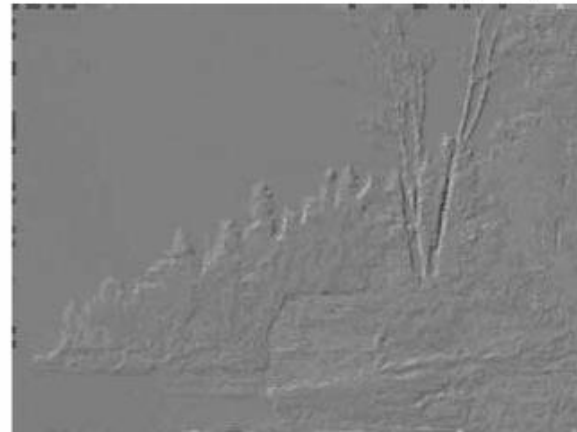
Unfiltered source (A)



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')

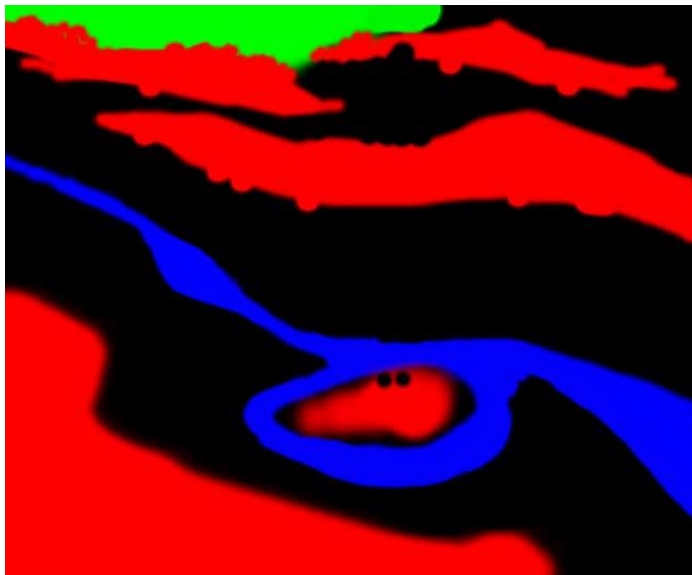
Texture-by-numbers



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 how to define similarity and efficiently find neighbors



Next class

- Cutting and seam finding