Training-based image processing: Example-based analysis and synthesis of images

Bill Freeman,
Fredo Durand
6.098/6.882 MIT

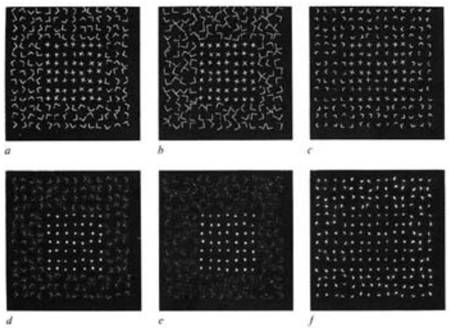
Collaborators:
Alyosha Efros, CMU, Ray Jones, MIT, Egon Pasztor, Google
March, 2006

A brief and biased history of texture synthesis methods

Learn: use filters.

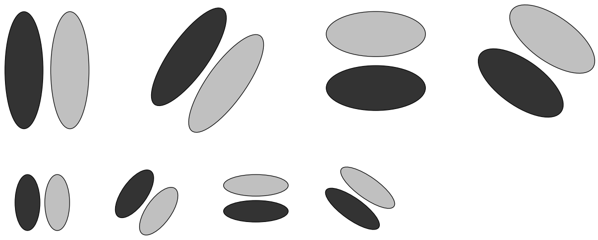
Bergen and Adelson, Nature 1988

Fig. 1 *Top row*, Textures consisting of Xs within a texture composed of Ls. The micropatterns are placed at random orientations on a randomly perturbed lattice. *a*, The bars of the Xs have the same length as the bars of the Ls. *b*, The bars of the Ls have been lengthened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is enhanced. *c*, The bars of the Ls have been shortened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is impaired. *Bottom row*: the responses of a size-tuned mechanism *d*, response to image *a*; *e*, response to image *b*; *f*, response to image *c*.



Learn: use lots of filters, multi-ori&scale.

Malik and Perona



Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms. J OPT SOC AM A 7: (5) 923-932 MAY 1990

Learn: use filter marginal statistics.

Bergen and Heeger

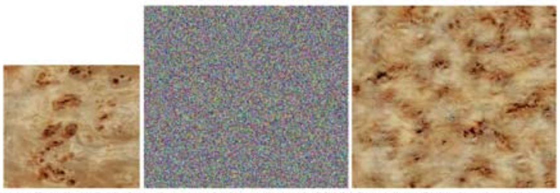


Figure 2: (Left) Input digitized sample texture: burled mappa wood. (Middle) Input noise. (Right) Output synthetic texture that matches the appearance of the digitized sample. Note that the synthesized texture is larger than the digitized sample; our approach allows generation of as much texture as desired. In addition, the synthetic textures tile seamlessly.

Bergen and Heeger results

Figure 3: In each pair left image is original and right image is synthetic: stucco, iridescent ribbon, green marble, panda fur, slag stone, figured yew wood.

Bergen and Heeger failures

Figure 8: Examples of failures: wood grain and red coral.

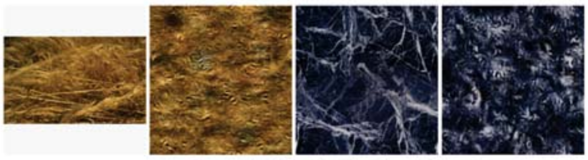


Figure 9: More failures: hay and marble.

De Bonet (and Viola)

SIGGRAPH 1997

Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images

Jeremy S. De Bonet –
Learning & Vision Group
Artificial Intelligence Laboratory
Massachusetts Institute of Technology

EMAIL: jsd@ai.mit.edu
HOMEPAGE: http://www.ai.mit.edu/_jsd

Learn: use filter conditional statistics across scale.

DeBonet

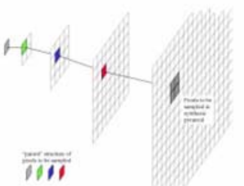


Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the "parent" structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

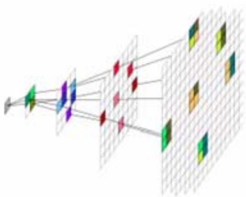
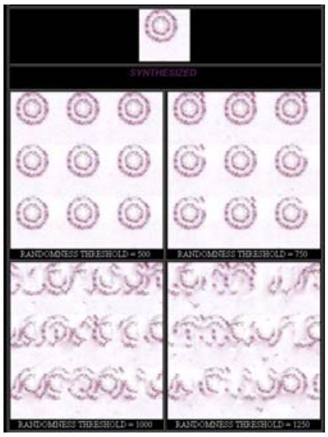
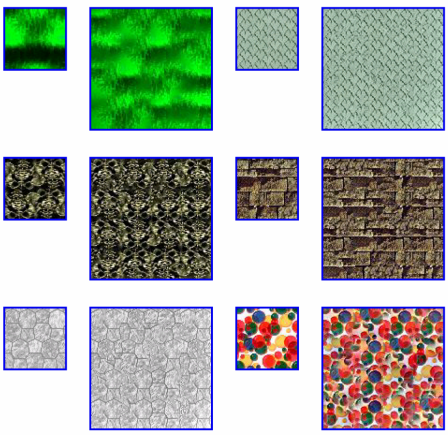


Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.



DeBonet



DeBonet

What we've learned from the previous texture synthesis methods

From Adelson and Bergen:

examine filter outputs

From Perona and Malik:

use multi-scale, multi-orientation filters.

From Heeger and Bergen:

use marginal statistics (histograms) of filter responses.

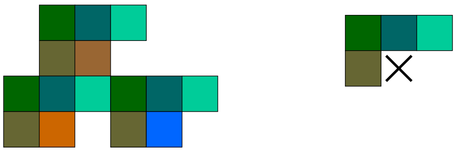
From DeBonet:

use conditional filter responses across scale.

IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

Texture Synthesis by Non-parametric Sampling

Alexei A. Efros and Thomas K. Leung
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, U.S.A.
{efros,leungt}@cs.berkeley.edu



Efros & Leung ’99

- [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”*
- Notice how well local structure is preserved!
  - Now, instead of letters let’s try pixels…

Efros and Leung

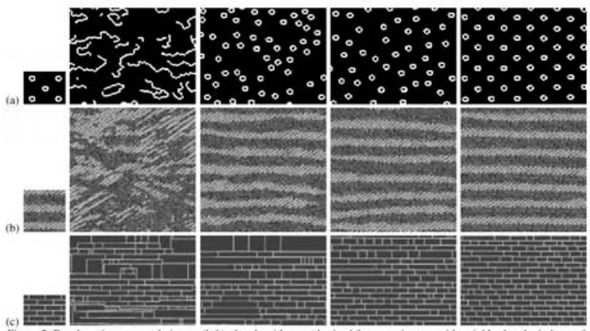


Figure 2. Results: given a sample image (left), the algorithm synthesized four new images with neighborhood windows of width 5, 11, 15, and 23 pixels respectively. Notice how perceptually intuitively the window size corresponds to the degree of randomness in the resulting textures. Input images are: (a) synthetic rings, (b) Brodatz texture D11, (c) brick wall.

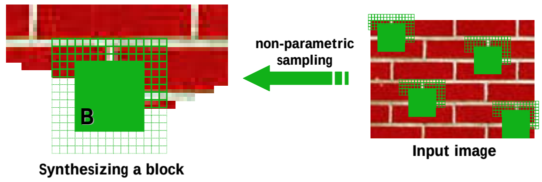
What we learned from Efros and Leung regarding texture synthesis

- Don’t need conditional filter responses across scale
- Don’t need marginal statistics of filter responses.
- Don’t need multi-scale, multi-orientation filters.
- Don’t need filters.

Efros & Leung '99

- The algorithm
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow
- Optimizations and Improvements
  - [Wei & Levoy,'00] (based on [Popat & Picard,'93])
  - [Harrison,'01]
  - [Ashikhmin,'01]

Efros & Leung '99 extended



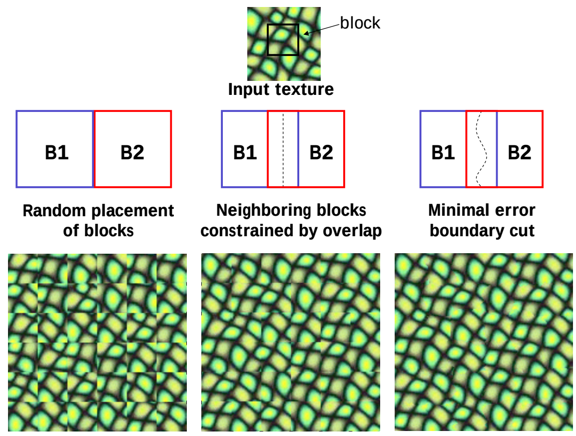
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

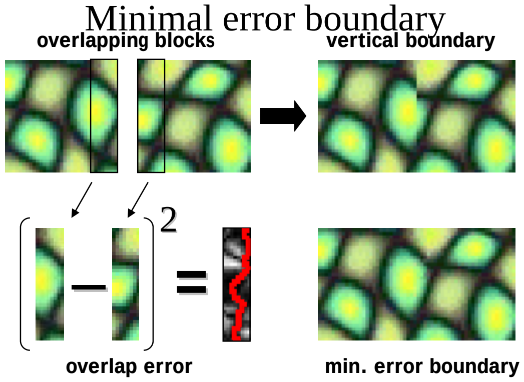
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

Image Quilting

- Idea:
  - let's combine random block placement of Chaos Mosaic with spatial constraints of Efros & Leung
- Related Work (concurrent):
  - Real-time patch-based sampling [Liang et.al. '01]
  - Image Analogies [Hertzmann et.al. '01]



Minimal error boundary



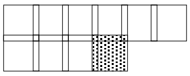
Our Philosophy

- 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

Algorithm

– Pick size of block and size of overlap

– Synthesize blocks in raster order

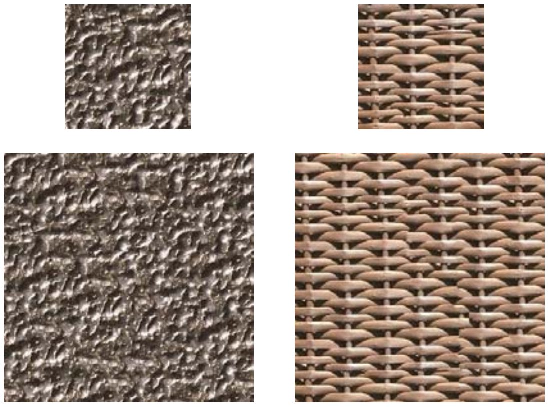


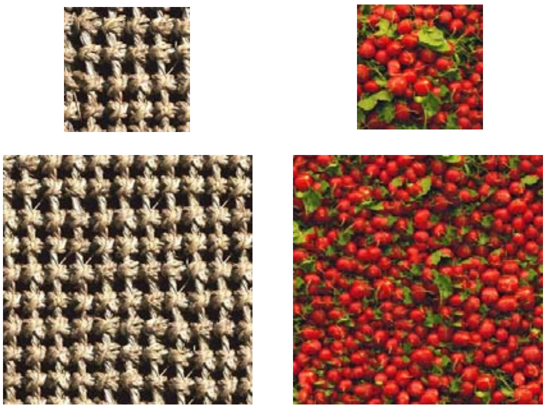
– Search input texture for block that satisfies overlap constraints (above and left)

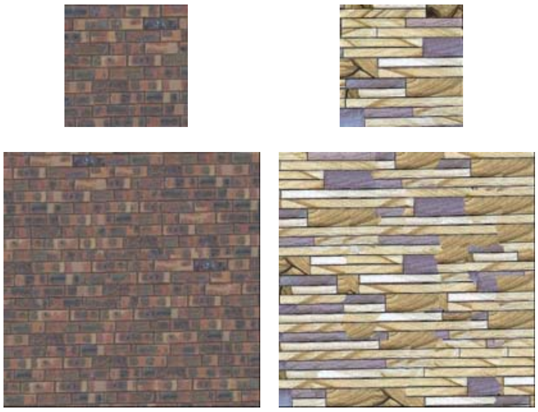
  • Easy to optimize using NN search [Liang et.al., ’01]

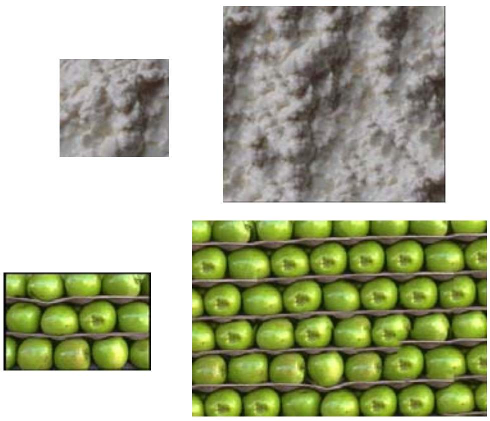
– Paste new block into resulting texture

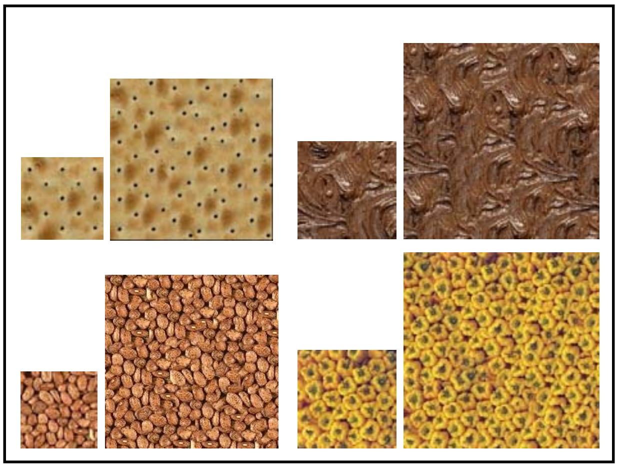
  • use dynamic programming to compute minimal error boundary cut

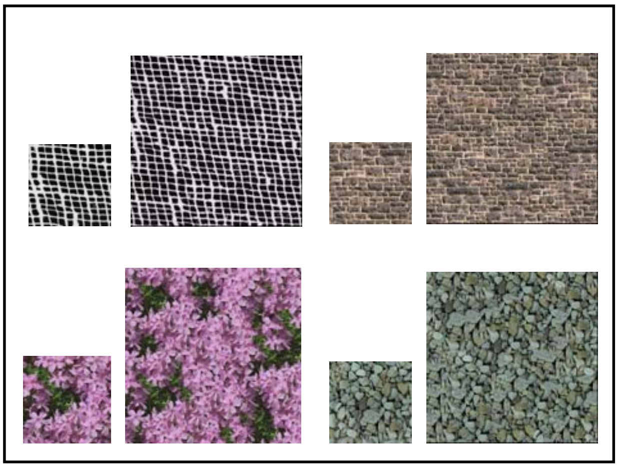




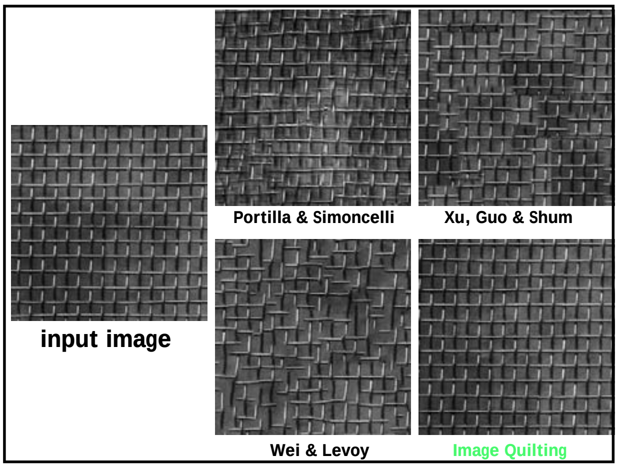


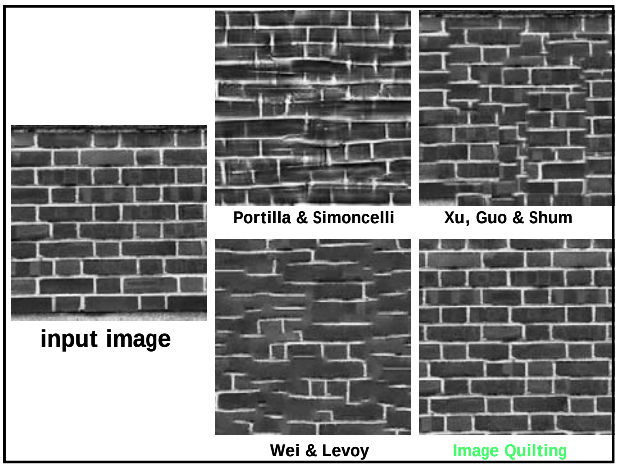


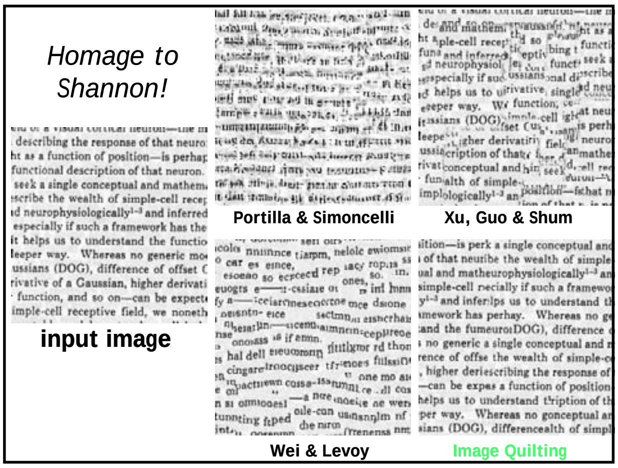





Failures
(Chernobyl
Harvest)

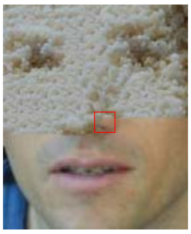

input image
Portilla & Simoncelli
Xu, Guo & Shum
Wei & Levoy
Image Quilting


input image
Portilla & Simoncelli
Xu, Guo & Shum
Wei & Levoy
Image Quilting

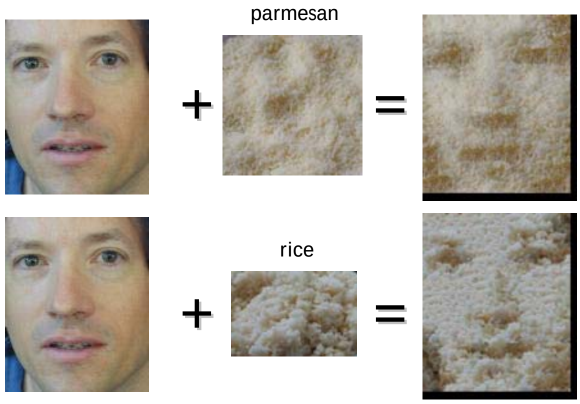

Homage to
Shannon!
input image
Portilla & Simoncelli
Xu, Guo & Shum
Wei & Levoy
Image Quilting

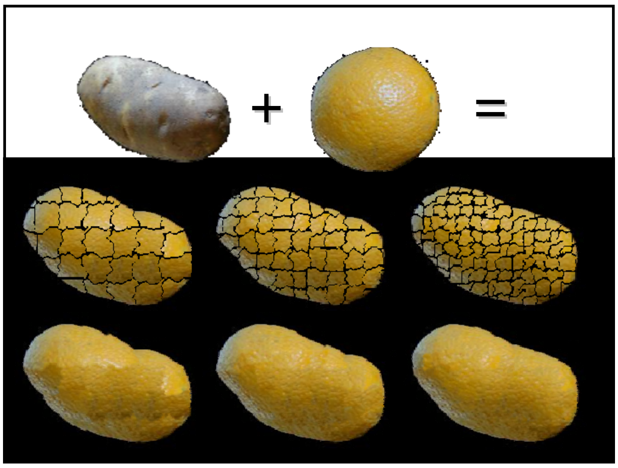
Texture Transfer

- 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





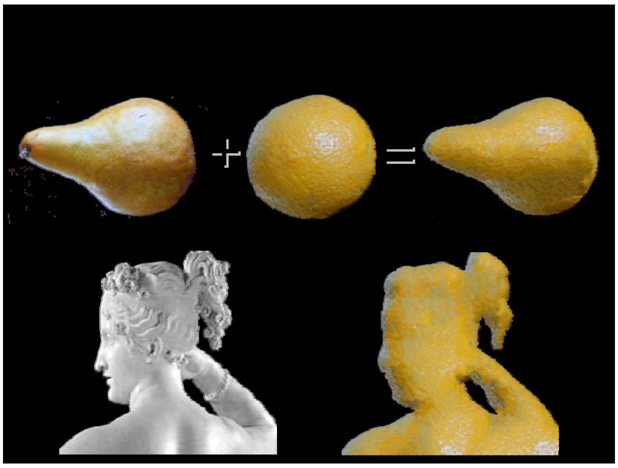
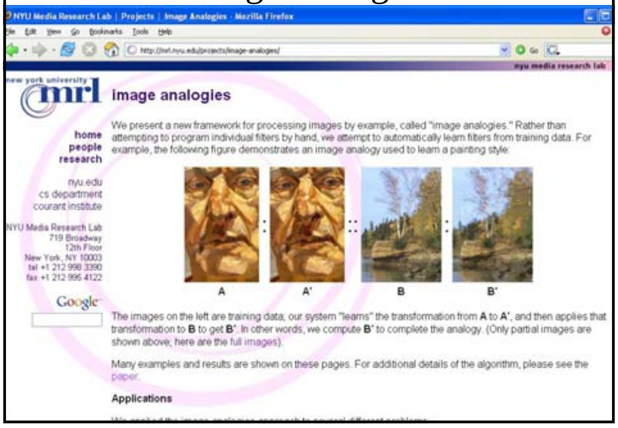
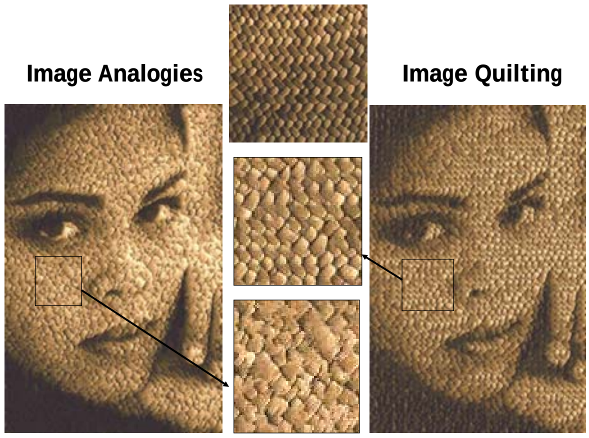


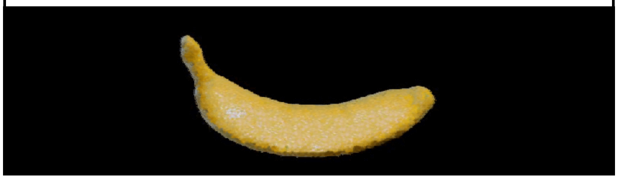
Image analogies





Summary of image quilting

- Quilt together patches of input image
  - randomly (texture synthesis)
  - constrained (texture transfer)
- Image Quilting
  - No filters, no multi-scale, no one-pixel-at-a-time!
  - fast and very simple
  - Results are not bad



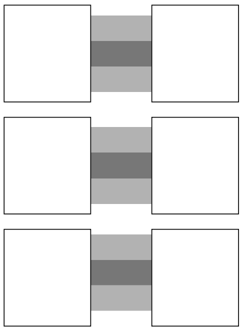
Part 2

- Data driven approach for other image processing and computer vision problems. Example: super-resolution.

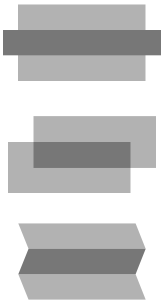
Prescription for doing vision

"Propagate local evidence"

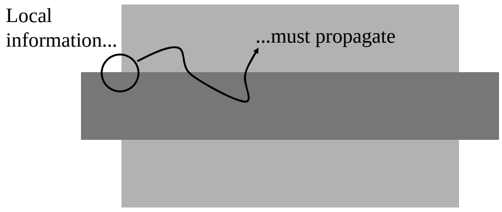
Identical image intensities...



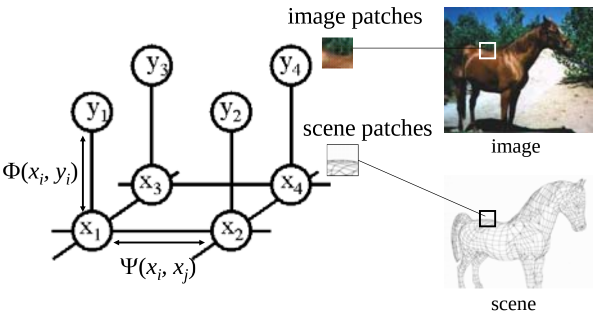
…different interpretations



Information must propagate over the image.



Model image and scene patches as nodes in a Markov network



Network joint probability

$$P(x,y) = \frac{1}{Z} \prod_{i,j} \Psi(x_i, x_j) \prod_i \Phi(x_i, y_i)$$

scene

image

Scene-scene compatibility function

neighboring scene nodes

Image-scene compatibility function

local observations

How represent the local image interpretations?

- Gaussian distributions of parameters
- Particles
  - Condensation
  - Non-parametric belief propagation
- Examples

Exemplars

- Gives you a discrete set of states; makes system easy to debug.
- Easy to propagate hypotheses.
- Add realistic details with real-world samples.
- Key implementation issue: need to use tricks to squeeze as much as you can out of each example.

Outline

- Fun with exemplars
  - Super-resolution
  - (Texture synthesis and style modification)
- Limitations of exemplars; other directions

Examples of exemplars

- Super-resolution
- (Texture synthesis and transfer)

- Line drawing style modification
- Shape-from-shading/reflectance estimation
- Motion estimation
- Human body animation

Examples of exemplars

- Super-resolution
- (Texture synthesis and transfer)

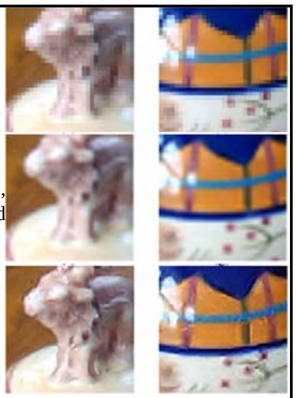
- Line drawing style modification
- Shape-from-shading/reflectance estimation
- Motion estimation
- Human body animation

Super-resolution

- Image: low resolution image
- Scene: high resolution image

ultimate goal...

Pixel-based images are not resolution independent



Pixel replication

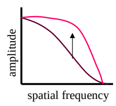
Cubic spline, sharpened

Training-based super-resolution

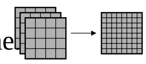
Polygon-based graphics images are resolution independent

3 approaches to perceptual sharpening

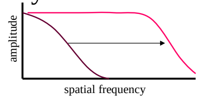
(1) Sharpening; boost existing high frequencies.



(2) Use multiple frames to obtain higher sampling rate in a still frame



(3) Estimate high frequencies not present in image, although implicitly defined.



In this talk, we focus on (3), which we'll call "super-resolution".

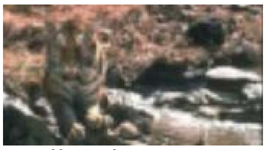
Super-resolution: other approaches

- Schultz and Stevenson, 1994
- Pentland and Horowitz, 1993
- fractal image compression (Polvere, 1998; Iterated Systems)
- astronomical image processing (eg. Gull and Daniell, 1978; “pixons” http://casswww.ucsd.edu/puetter.html)

Training images, **~100,000 image/scene patch pairs**

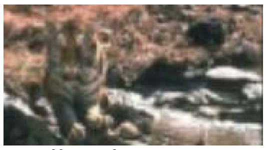
Images from two Corel database categories:
“giraffes” and “urban skyline”.

Do a first interpolation



Zoomed low-resolution

Low-resolution



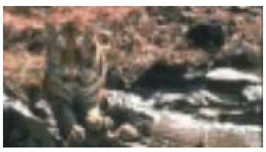
Zoomed low-resolution

Full frequency original

Low-resolution

Representation

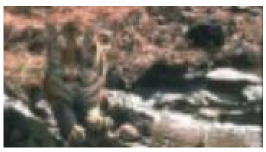
Zoomed low-freq.



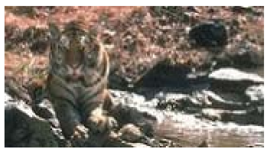
Full freq. original

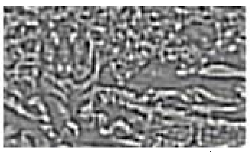
Representation

Zoomed low-freq.

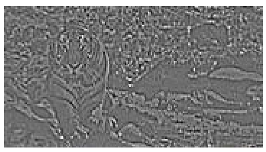


Full freq. original





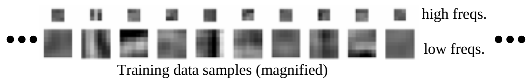
Low-band input
(contrast normalized,
PCA fitted)



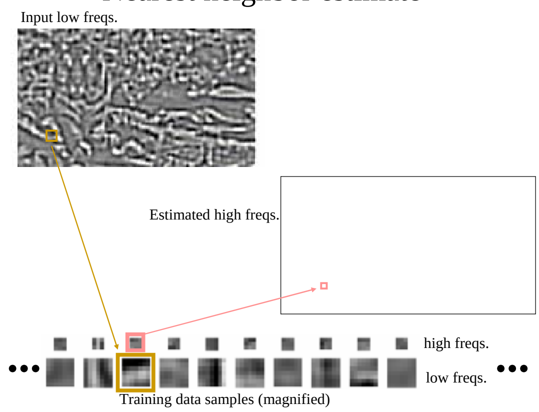
True high freqs

(to minimize the complexity of the relationships we have to learn,
we remove the lowest frequencies from the input image,
and normalize the local contrast level).

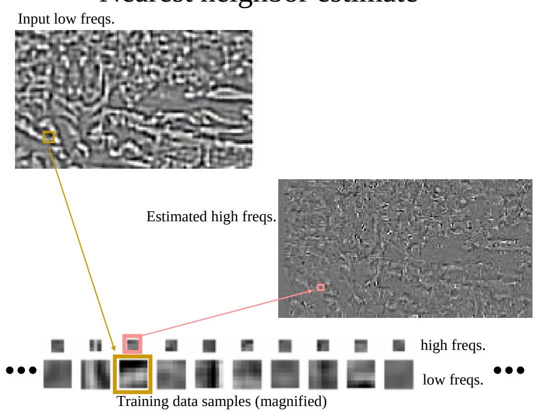
Gather ~100,000 patches



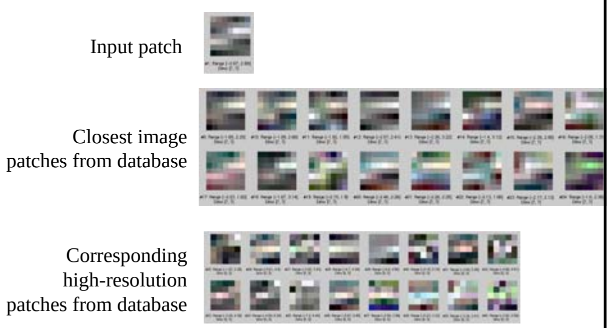
Nearest neighbor estimate

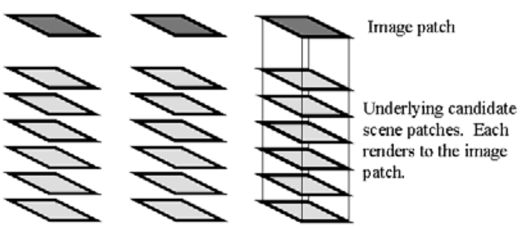


Nearest neighbor estimate

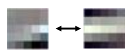


Example: input image patch, and closest matches from database



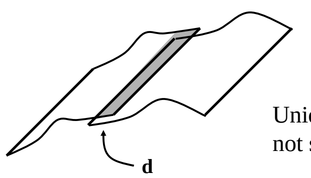


Scene-scene compatibility function, $\Psi(x_i, x_j)$



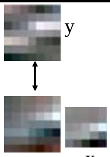
Assume overlapped regions, d, of hi-res. patches differ by Gaussian observation noise:

$$\Psi(x_i, x_j) = \exp^{-|d_i - d_j|^2/2\sigma^2}$$



Uniqueness constraint, not smoothness.

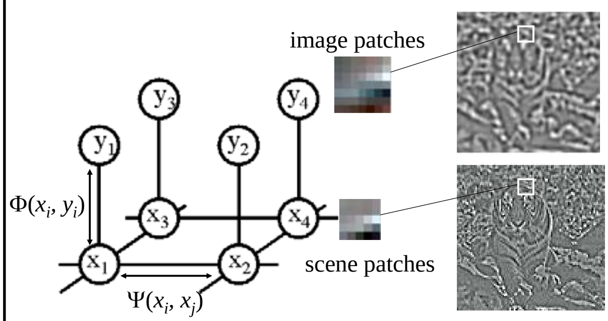
Image-scene compatibility function, $\Phi(x_i, y_i)$



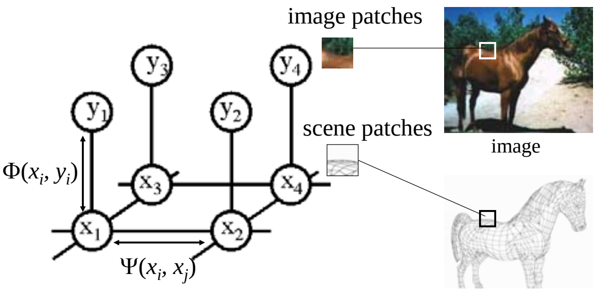
Assume Gaussian noise takes you from observed image patch to synthetic sample:

$$\Phi(x_i, y_i) = \exp^{-|y_i - y(x_i)|^2 / 2\sigma^2}$$

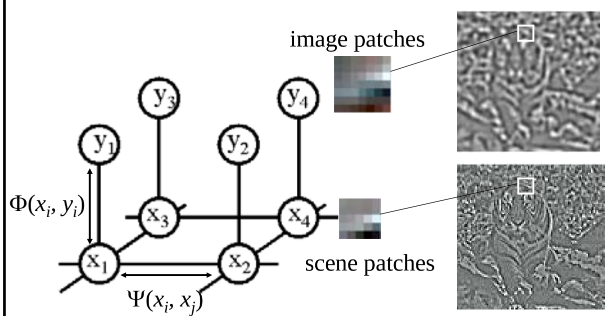
Markov network



VISTA-- Vision by Image-Scene TrAining



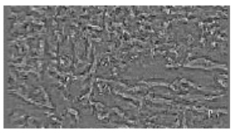
Super-resolution application



Belief Propagation

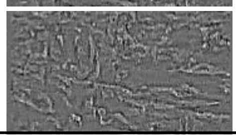
After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

Input



Iter. 0

Iter. 1



Iter. 3

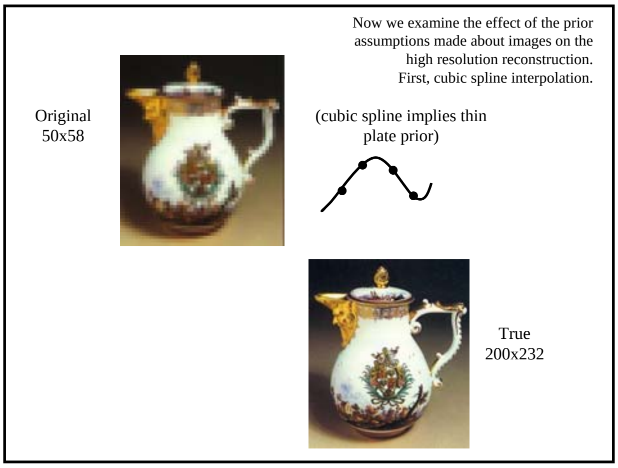
Zooming 2 octaves

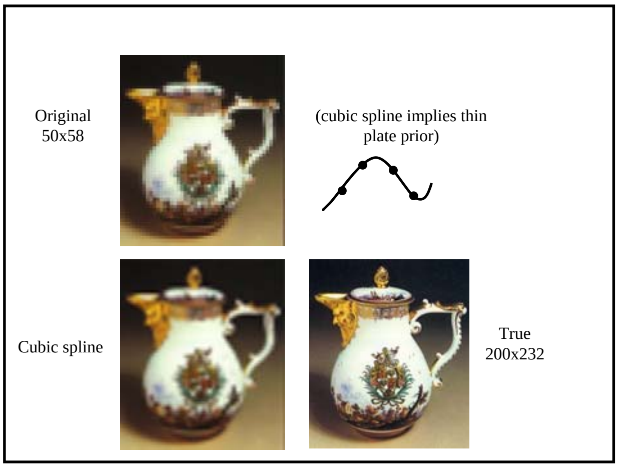
We apply the super-resolution algorithm recursively, zooming up 2 powers of 2, or a factor of 4 in each dimension.

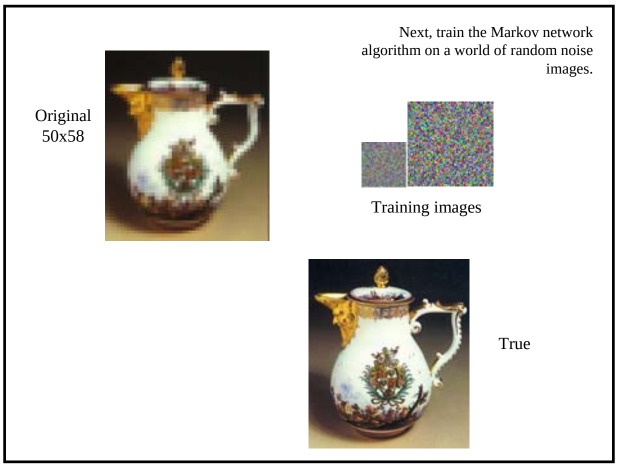
85 x 51 input

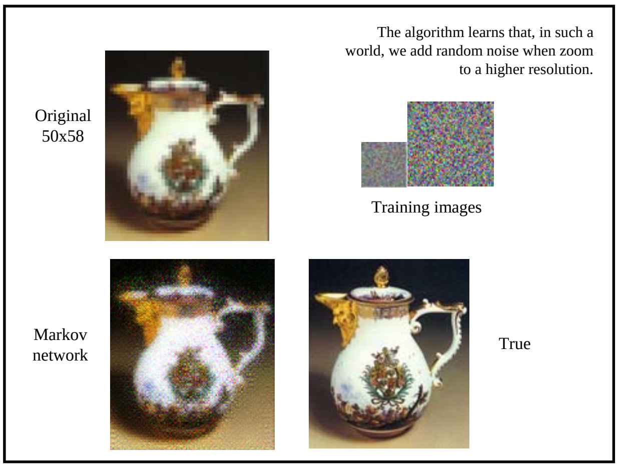
Cubic spline zoom to 340x204

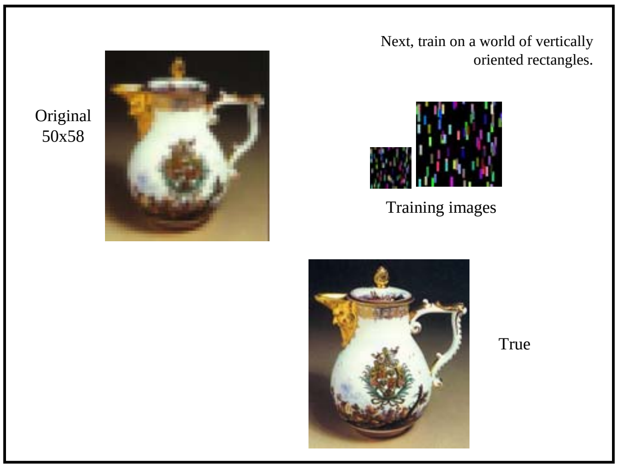
Max. likelihood zoom to 340x204

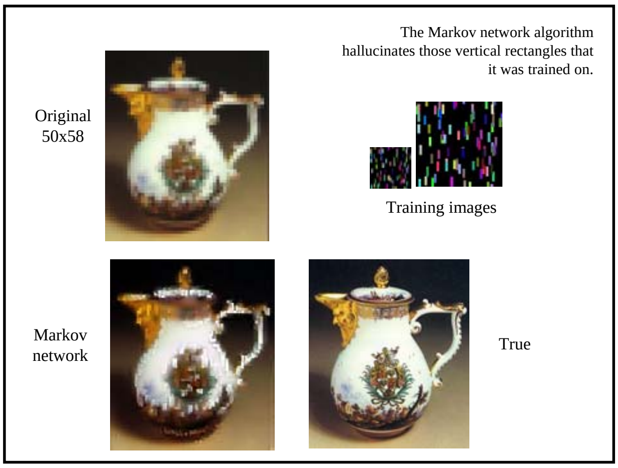

Now we examine the effect of the prior assumptions made about images on the high resolution reconstruction.
First, cubic spline interpolation.
Original
50x58
(cubic spline implies thin plate prior)
True
200x232


Original
50x58
(cubic spline implies thin plate prior)
Cubic spline
True
200x232


Next, train the Markov network algorithm on a world of random noise images.
Original
50x58
Training images
True


The algorithm learns that, in such a world, we add random noise when zoom to a higher resolution.
Original
50x58
Training images
Markov
network
True


Next, train on a world of vertically oriented rectangles.
Original
50x58
Training images
True


The Markov network algorithm hallucinates those vertical rectangles that it was trained on.
Original
50x58
Training images
Markov
network
True

Original
50x58

Now train on a generic collection of images.

Training images

True

The algorithm makes a reasonable guess at the high resolution image, based on its training images.

Original
50x58

Training images

Markov network

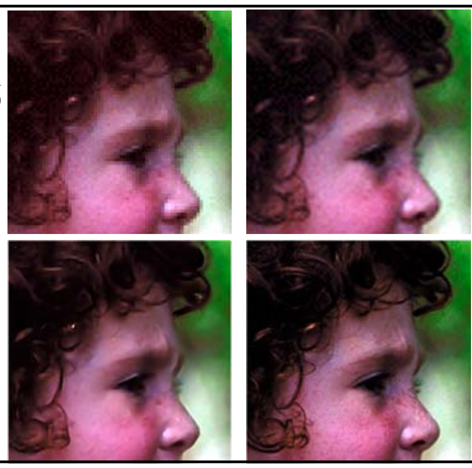
True

Generic training images

Next, train on a generic set of training images. Using the same camera as for the test image, but a random collection of photographs.

Original
70x70

Cubic
Spline



Markov
net,
training:
generic

True
280x280

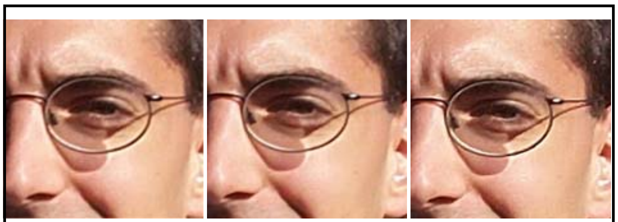
Kodak Imaging Science Technology Lab test.

3 test images, 640x480, to be zoomed up by 4 in each dimension.

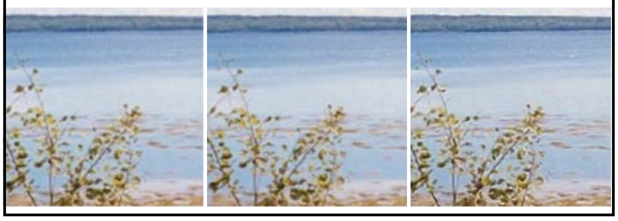
8 judges, making 2-alternative, forced-choice comparisons.

Algorithms compared

- Bicubic Interpolation
- Mitra's Directional Filter
- Fuzzy Logic Filter
- Vector Quantization
- VISTA



Bicubic spline Altamira VISTA

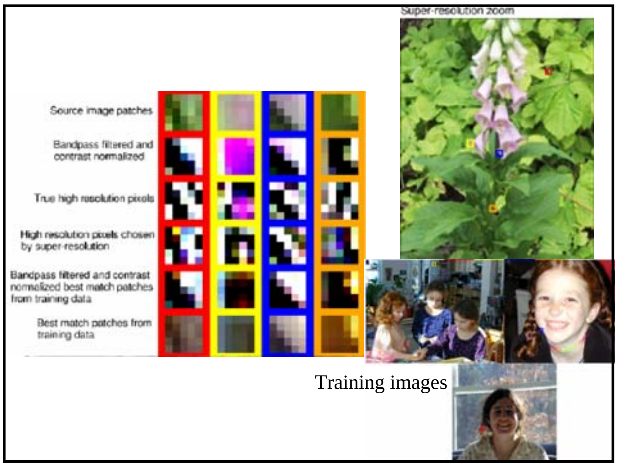


Bicubic spline Altamira VISTA

User preference test results

"The observer data indicates that six of the observers ranked Freeman's algorithm as the most preferred of the five tested algorithms. However the other two observers rank Freeman's algorithm as the least preferred of all the algorithms….

Freeman's algorithm produces prints which are by far the sharpest out of the five algorithms. However, this sharpness comes at a price of artifacts (spurious detail that is not present in the original scene). Apparently the two observers who did not prefer Freeman's algorithm had strong objections to the artifacts. The other observers apparently placed high priority on the high level of sharpness in the images created by Freeman's algorithm."



Training images

Training image

Processed image

Conclusions

- Exemplars (local, non-parametric image representations) are useful, fun, easy-to-use.
- Requirement: find ways to get by with too few exemplars.

end