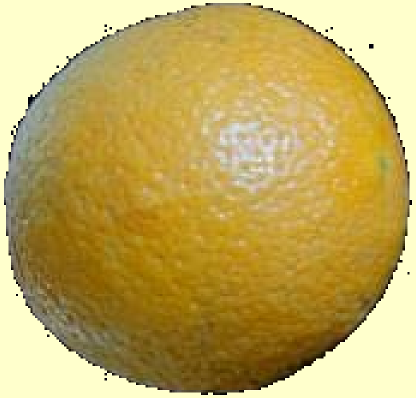
Texture Synthesis

Daniel Cohen-Or

+

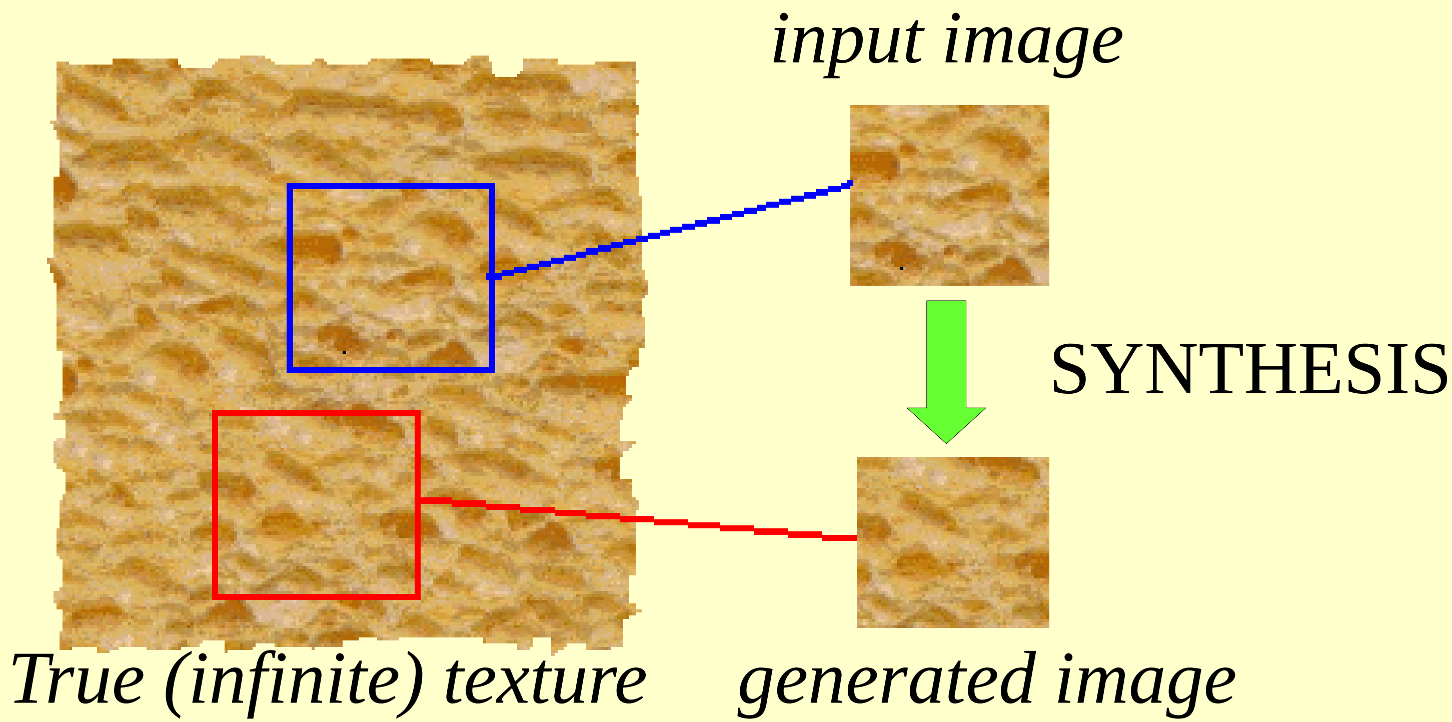


= ?

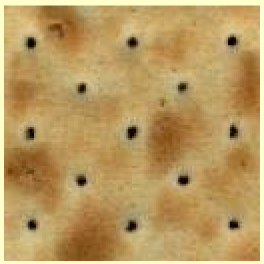
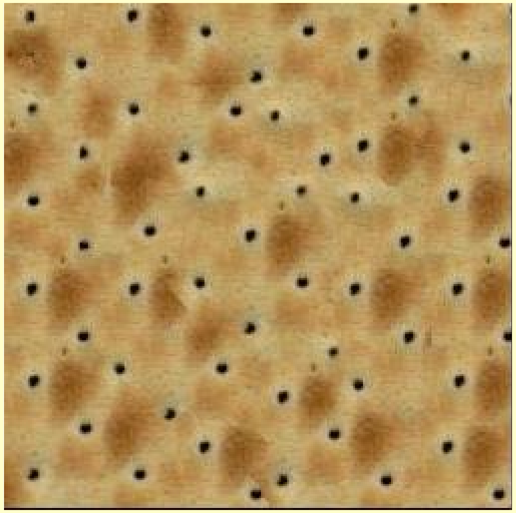
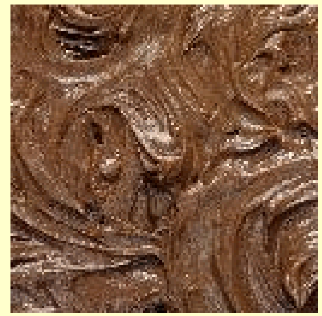
+

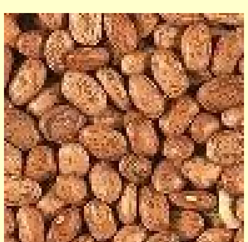
= ?

The Goal of Texture Synthesis



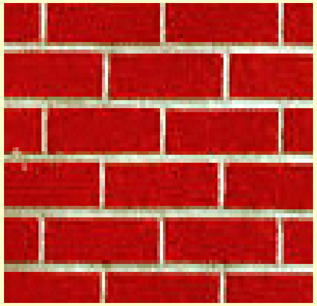
- Given a finite sample (large enough) of some texture, the goal is to synthesize other samples from that same texture.

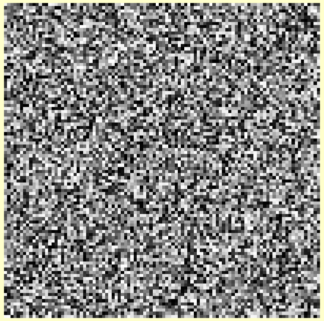



The Challenge

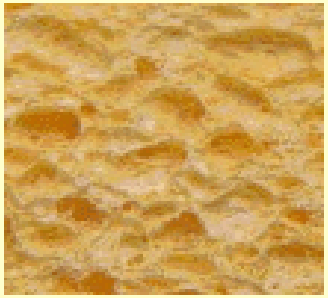
Need to model the whole spectrum: from repeated to stochastic texture



repeated

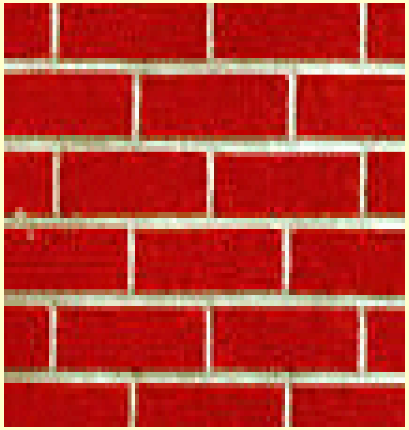


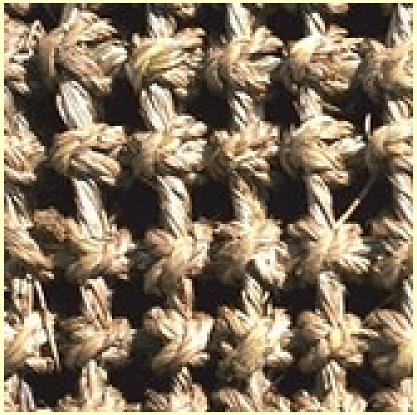
stochastic

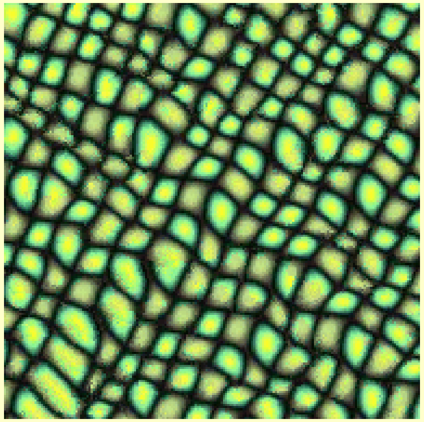


Both

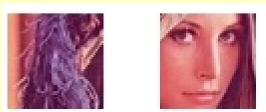
Texture Types

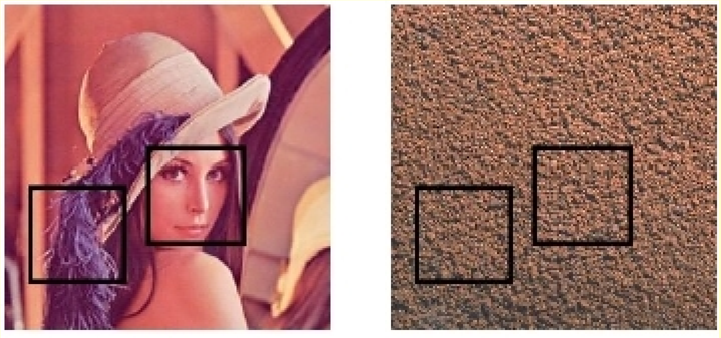


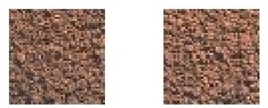





Texture model





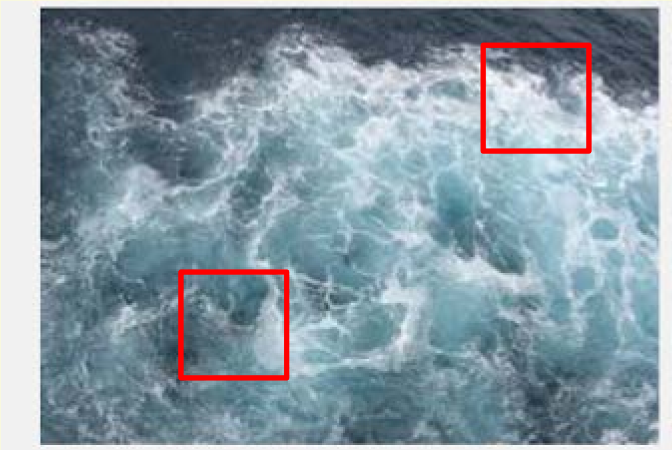


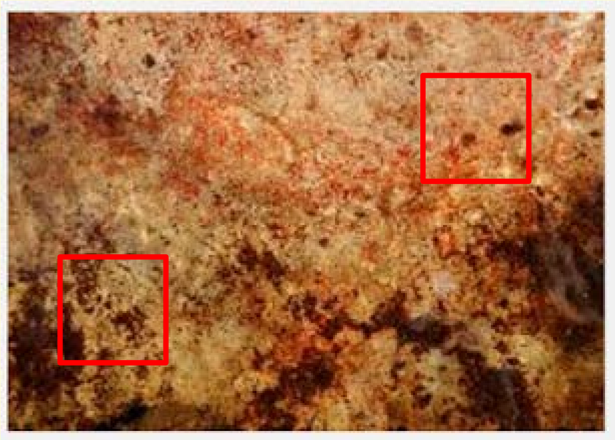
Stationary - under a proper window size, the observable portion always appears similar.

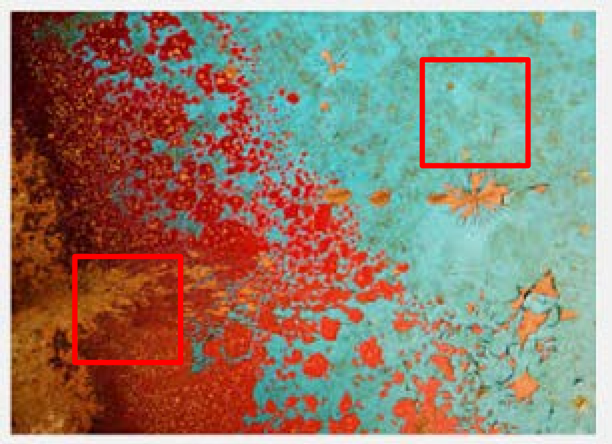
Local - each pixel is predictable from a small set of neighboring pixels and independent of the rest of the image.

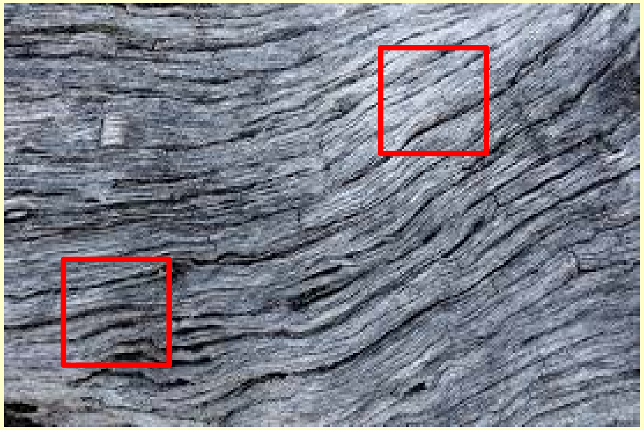
Non-Stationary

Non-Stationary









Texture Synthesis for Graphics

- Inspired by Texture Analysis and Psychophysics
  - [Heeger & Bergen,'95]
  - [DeBonet,'97]
  - [Portilla & Simoncelli,'98]
- …but didn't work well for structured textures
  - [Efros & Leung,'99]
    - (originally proposed by [Garber,'81])

“By Example” Texture Synthesis

- Input patch boundary.
- Input texture example.

- Fill boundary with texture.

Texture Synthesis by Non Parametric Sampling

- Generate English-looking text using N-grams, [Shannon,'48]

- Assuming Markov Chain on letters:
  - *P(letter | Proceeding n-letters)*

E N C Y C L O P E D I ? A

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

Unit of Synthesis

- Letter-by-letter: Used to name planets in early 80s game “Elite”.
- Word-by-word: M.V. Shaney (Bell Labs) 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“.*

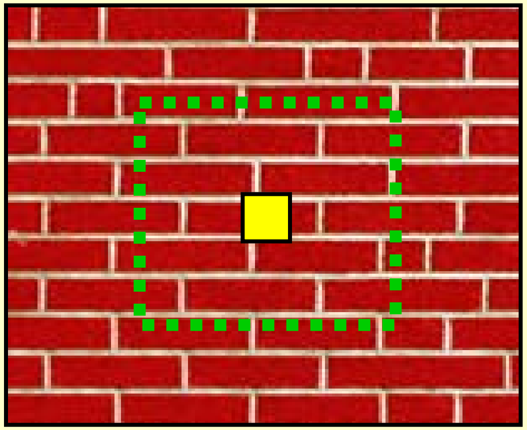
Mark V. Shaney (Bell Labs)

- Notice how well local structure is preserved!
  - Now, instead of letters let's try pixels…

Efros & Leung 99*

- Assuming Markov property, compute
  - *P(p | N(p))*.
  - – Explicit probability tables infeasible.
  - – Instead, *search input image* for similar neighbourhoods - that's our histogram for *p*.

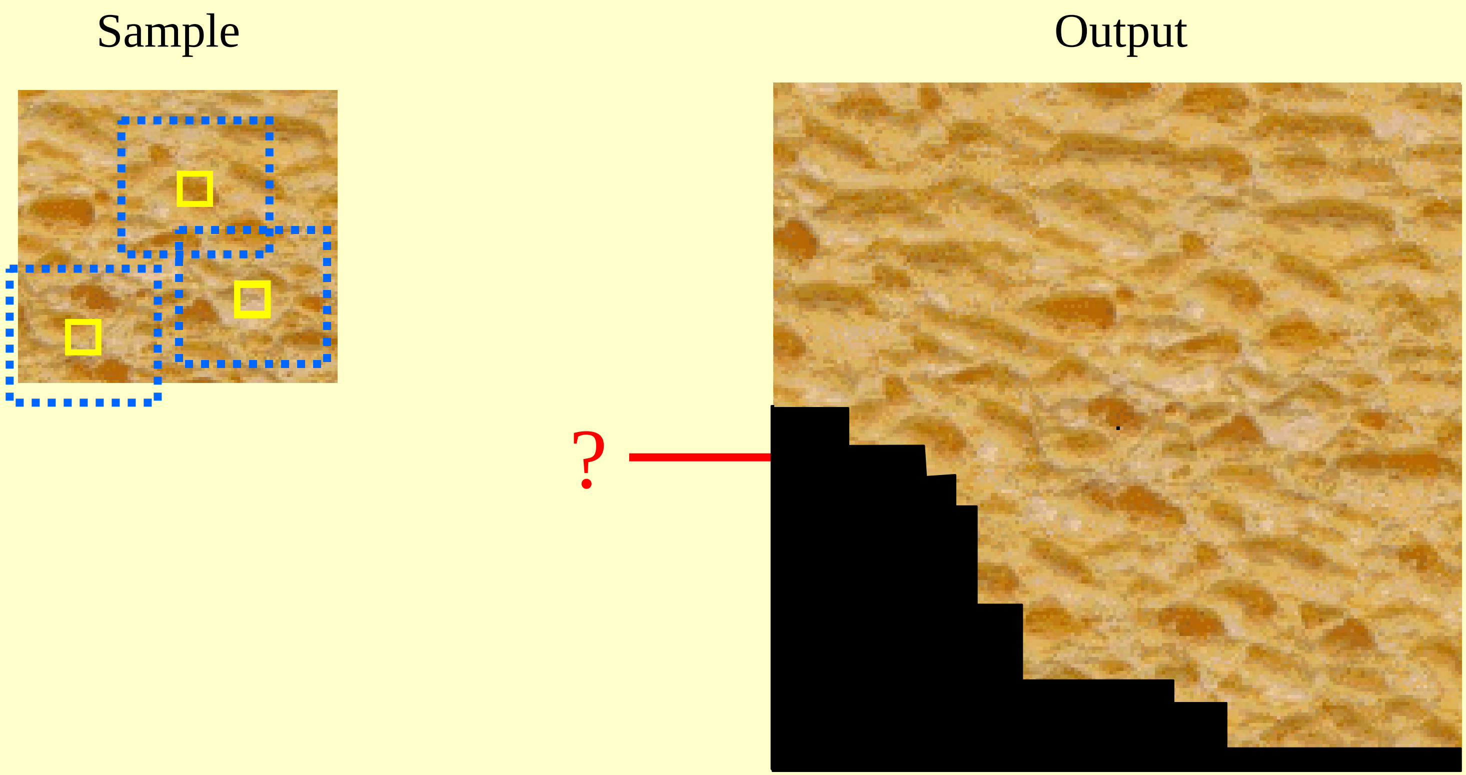
Non-parametric sampling



* A.A.Efros, T.K.Leung; "Texture synthesis by non-parametric sampling"; ICCV99.

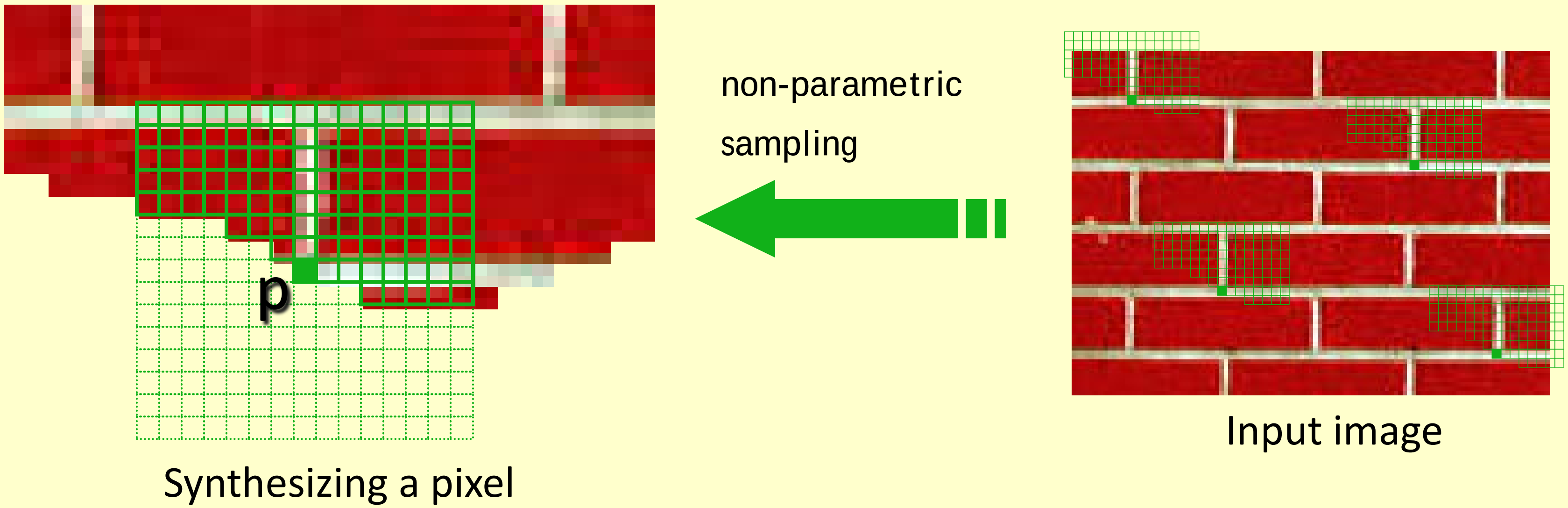
(originally proposed by [Garber,'81])

Efros & Leung 99 - Algorithm



- Causal neighborhood – Neighboring pixels with known values.

Efros & Leung '99



- Assuming Markov property, compute P(p|N(p))
  - Building explicit probability tables infeasible
  - Instead, let's search the input image for all similar neighborhoods — that's our histogram for p
- To synthesize p, just pick one match at random

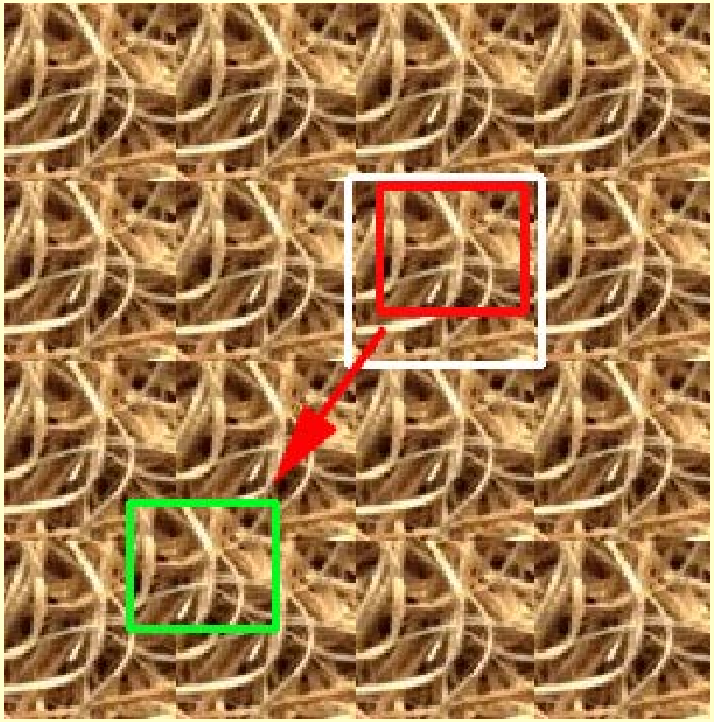
Efros & Leung ’99

- 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]
  - PatchMatch [Barnes et al. 2009]

Chaos Mosaic [Xu, Guo & Shum, '00]

input



idea

result

- Process: 1) tile input image; 2) pick random blocks and place them in random locations 3) Smooth edges

Used in Lapped Textures [Praun et.al,'00]

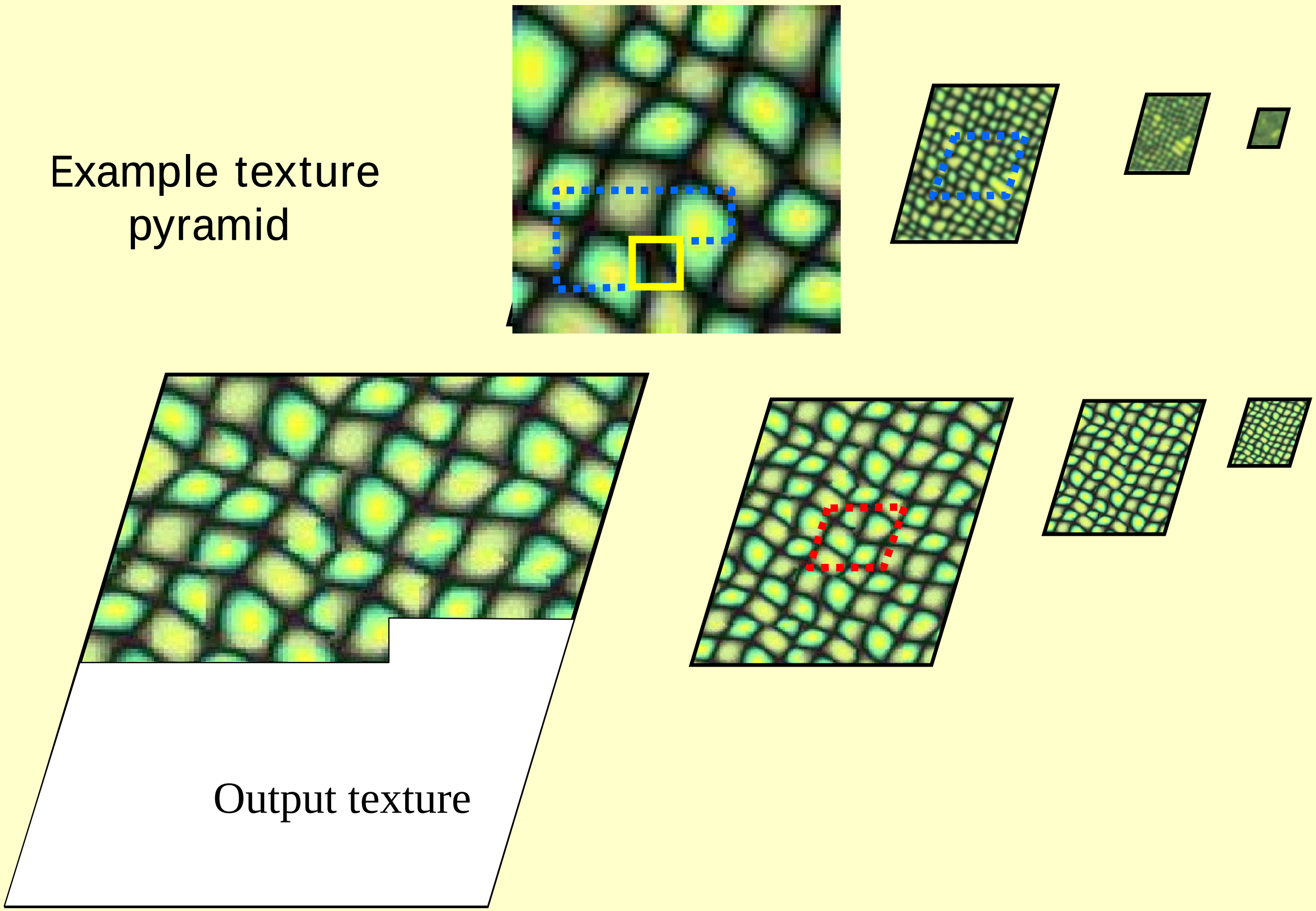
Chaos Mosaic [Xu, Guo & Shum, ‘00]

input

result

Of course, doesn’t work for structured textures

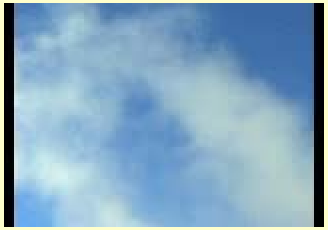
Multi-Resolution Pyramids*



* L.-Y.Wei, M.Levoy; "*Fast Texture Synthesis using Tree-structured Vector Quantization*"; SIGGRAPH00.

Extension to 3D Textures

- Motion both in space and time
  - fire, smoke, ocean waves.
- How to synthesize?
  - extend 2D algorithm to 3D.



The Problems of Causal Scanning

- Scanning order:
  - Efros&Leung[1]: Pixels with most neighbors.
  - Wei&Levoi[2]: Raster scan.
- These are "causal" scans.

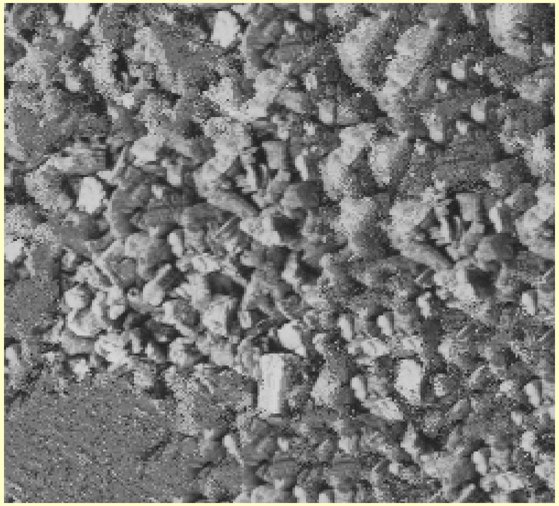
(1) A.A.Efros, T.K.Leung; "*Texture synthesis by non-parametric sampling*"; ICCV99.

(originally proposed by [Garber,'81])

(2) L.-Y.Wei, M.Levoy; "*Fast Texture Synthesis using Tree-structured Vector Quantization*"; SIGGRAPH00.

The Problems of Causal Scanning

- Can grow garbage.
- No natural means of refining synthesis.
- Cannot be parallelized.
- Problems are made worst for synthesis of 3D space-time volumes (a.k.a. video)...

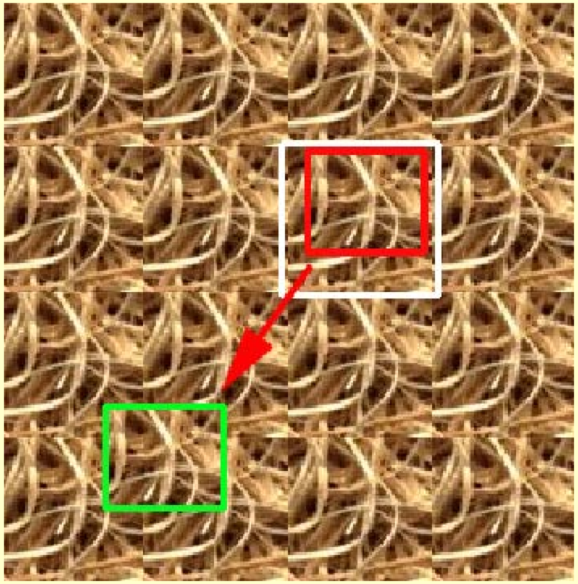


A.A.Efros, T.K.Leung; "*Texture synthesis by non-parametric sampling*"; ICCV99.

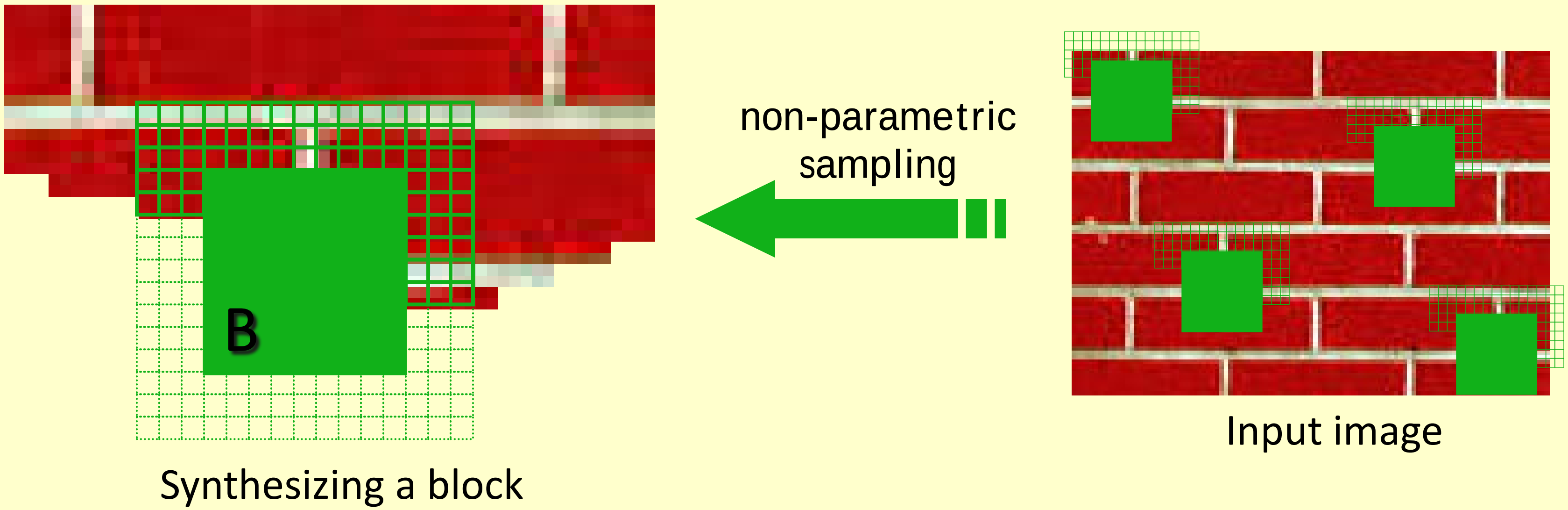
Image Quilting

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

- Idea:
  - let's combine random block placement of Chaos Mosaic with spatial constraints of Efros & Leung.

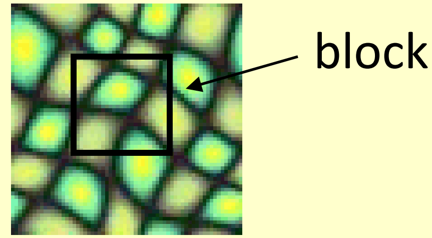


Efros & Leung '99 extended

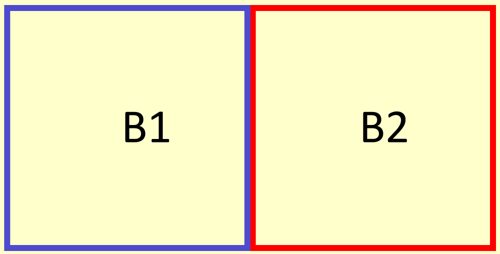


Idea: **unit of synthesis = block**

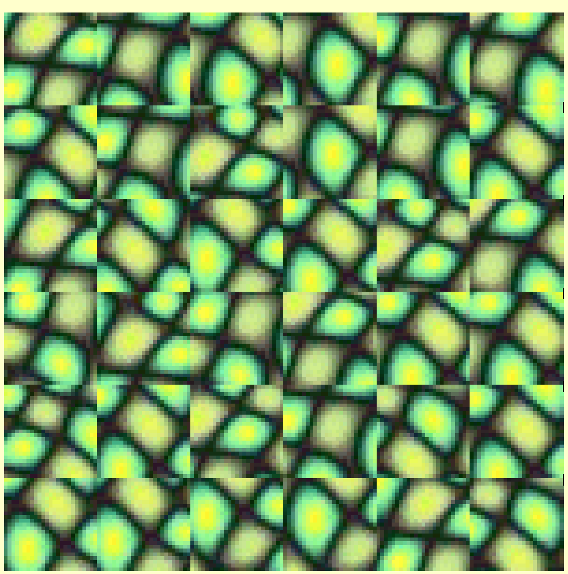
- Exactly the same but now we want P(B|N(B))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

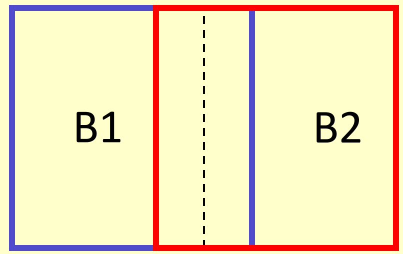


Input texture

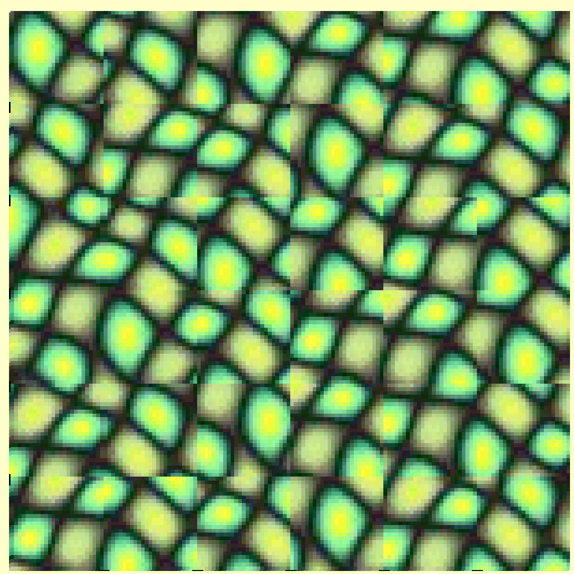


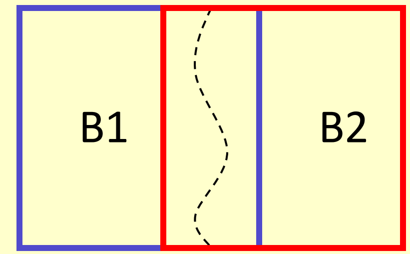
Random placement of blocks



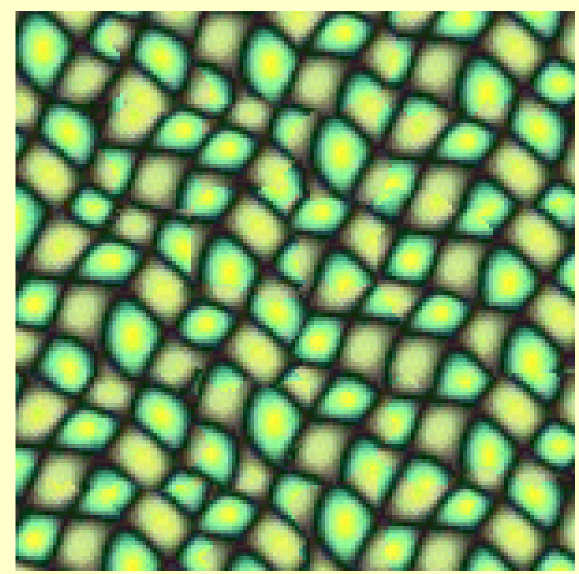


Neighboring blocks constrained by overlap





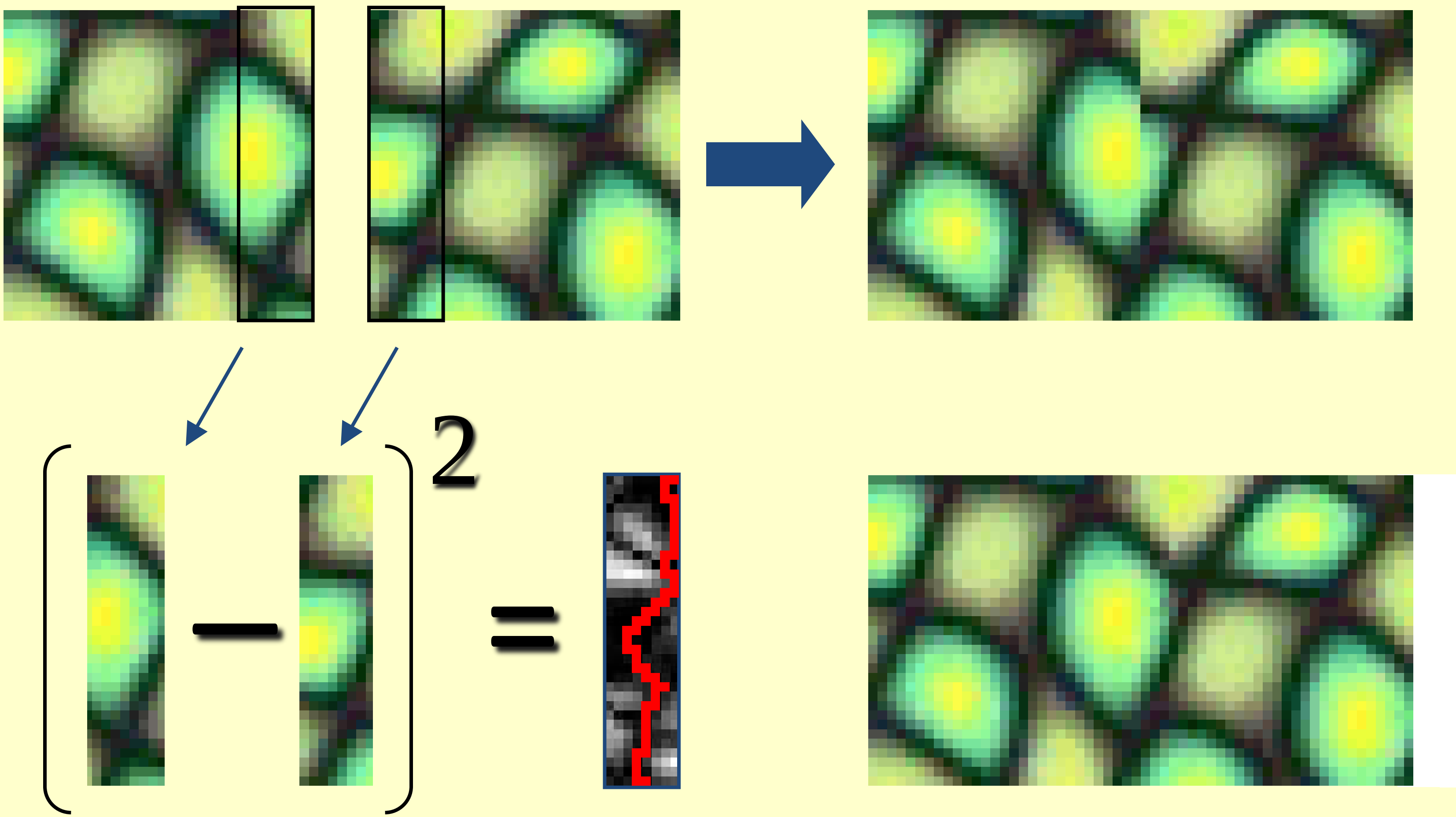
Minimal error boundary cut



Minimal error boundary

overlapping blocks

vertical boundary



overlap error

min. error boundary

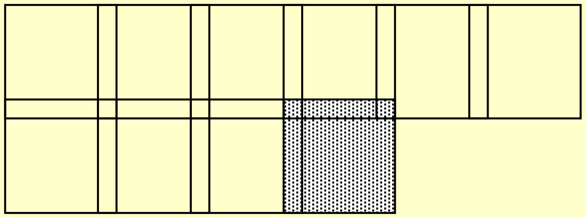
Our Philosophy

- 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

Image Quilting 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)

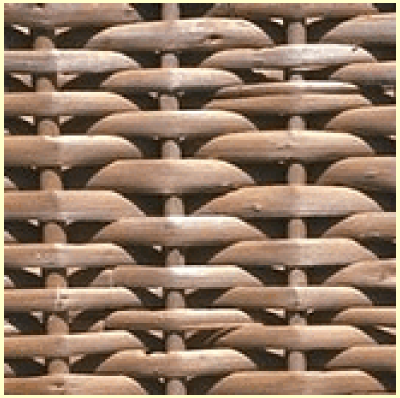
- Easy to optimize using NN search [Liang et.al., '01]

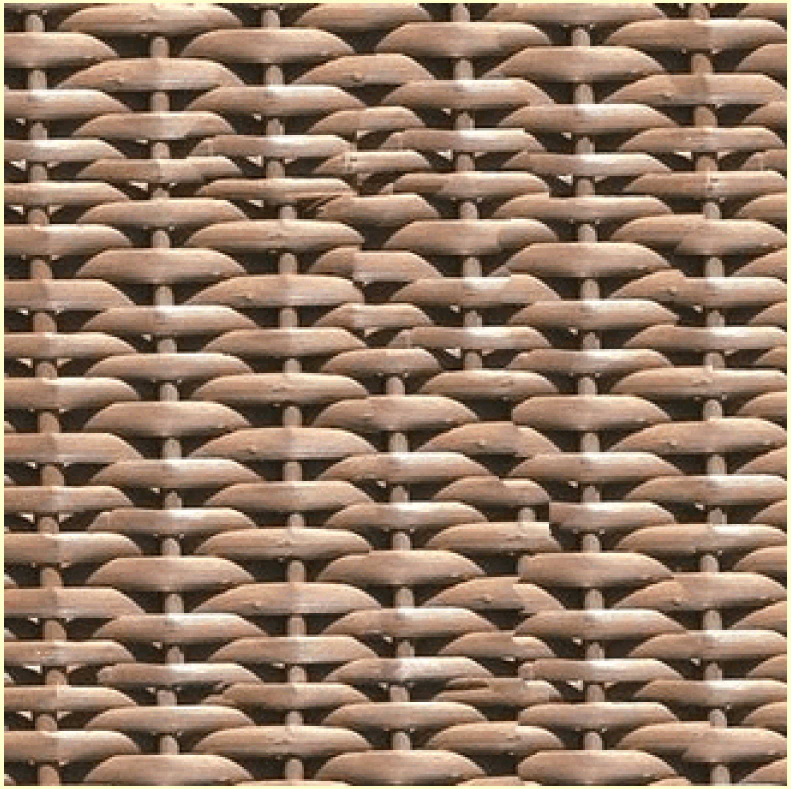
– Paste new block into resulting texture

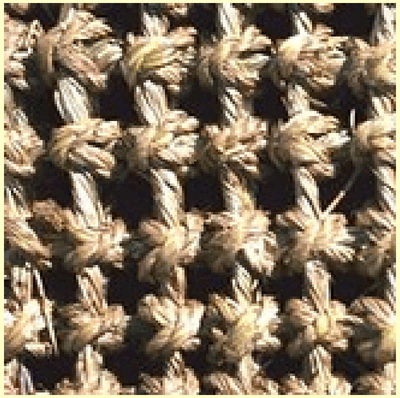
- use dynamic programming to compute minimal error boundary cut

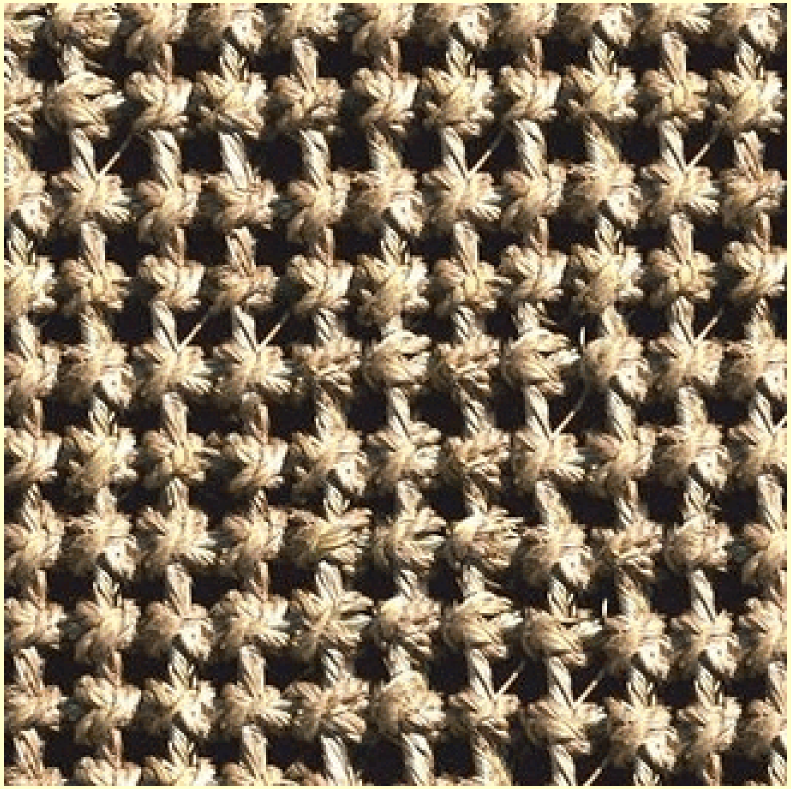
Video

See https://www.youtube.com/watch?v=t6DzioKuVEs

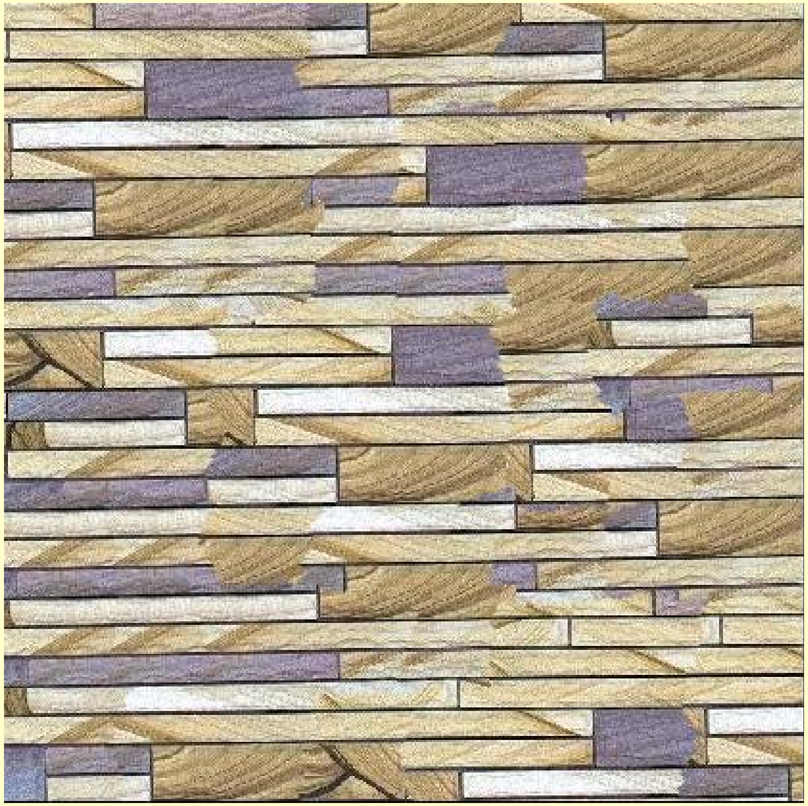




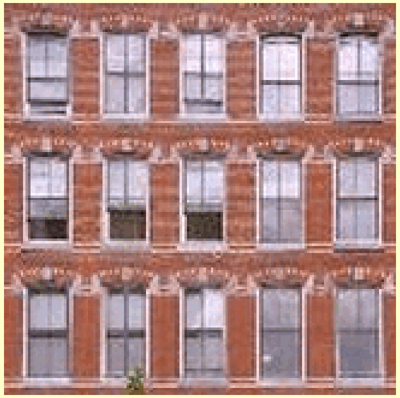




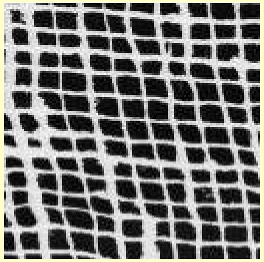
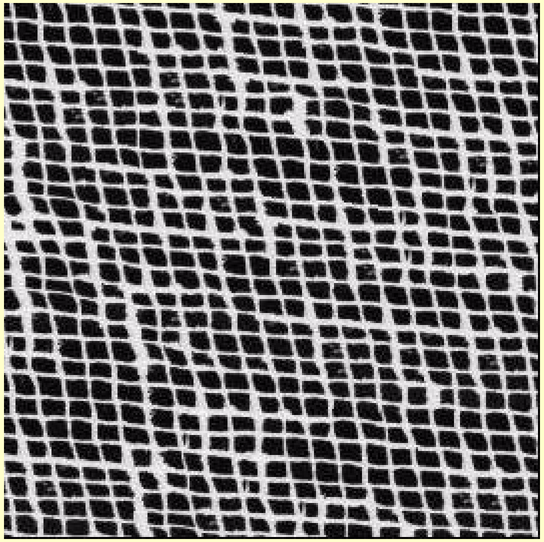


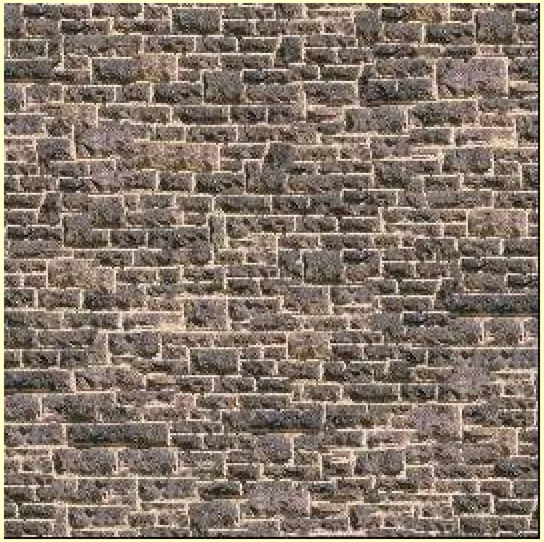



College Inn




College Inn

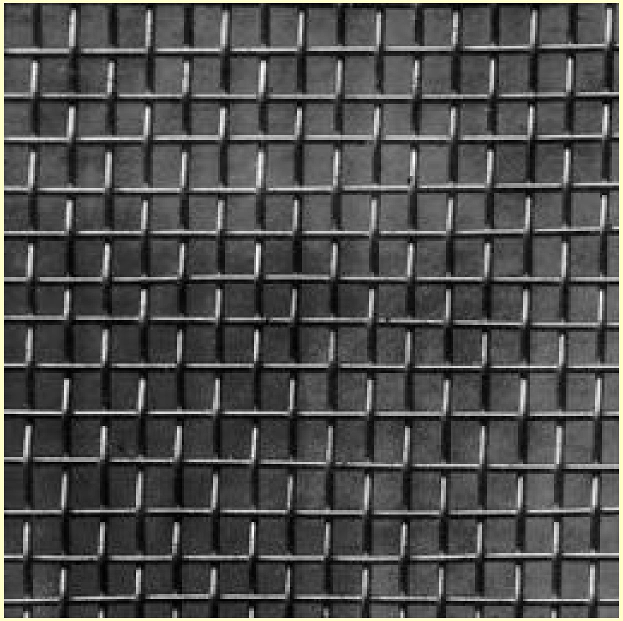





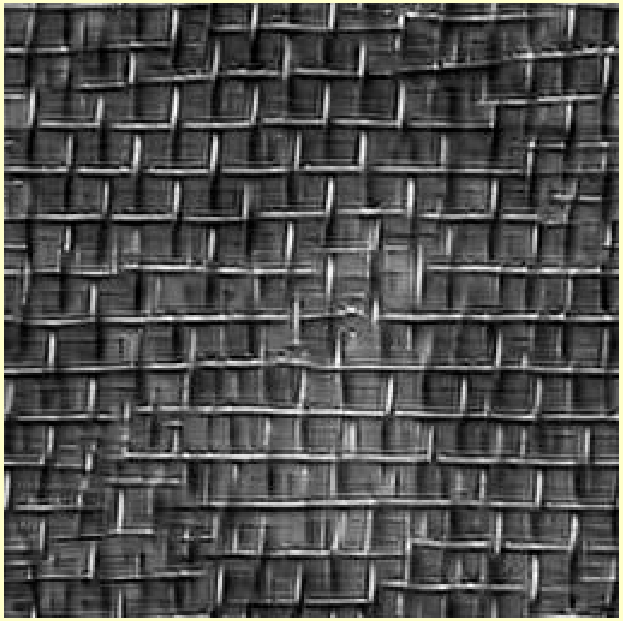


Failures

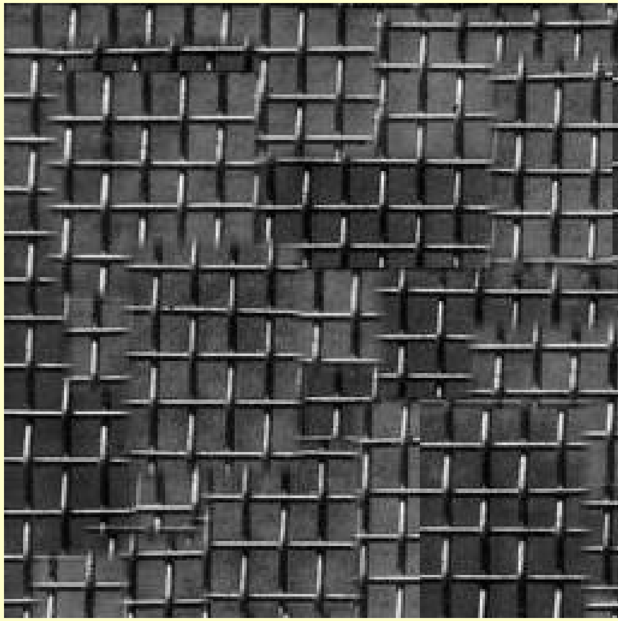
(Chernobyl Harvest)



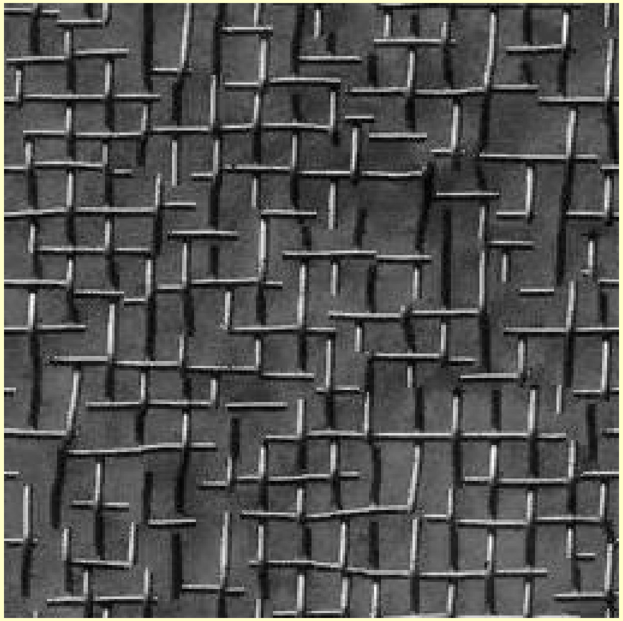
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

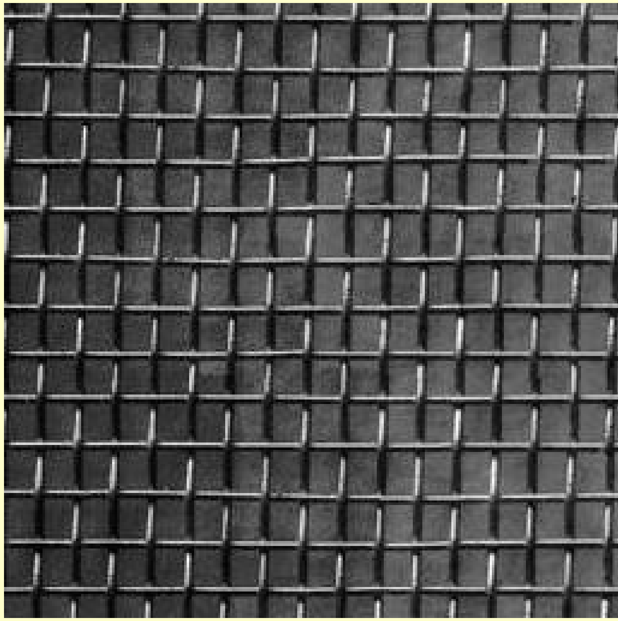
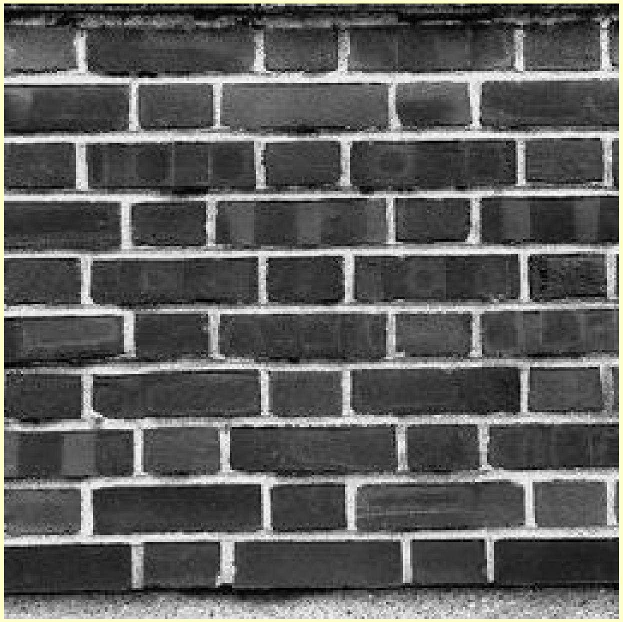
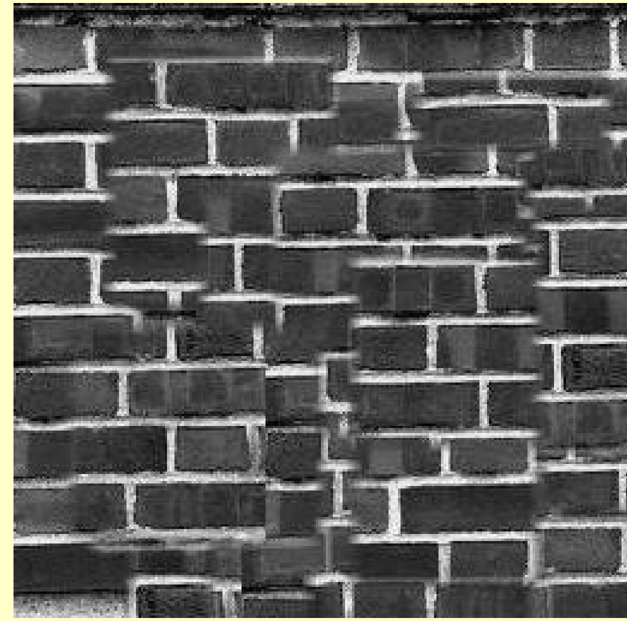


Image Quilting

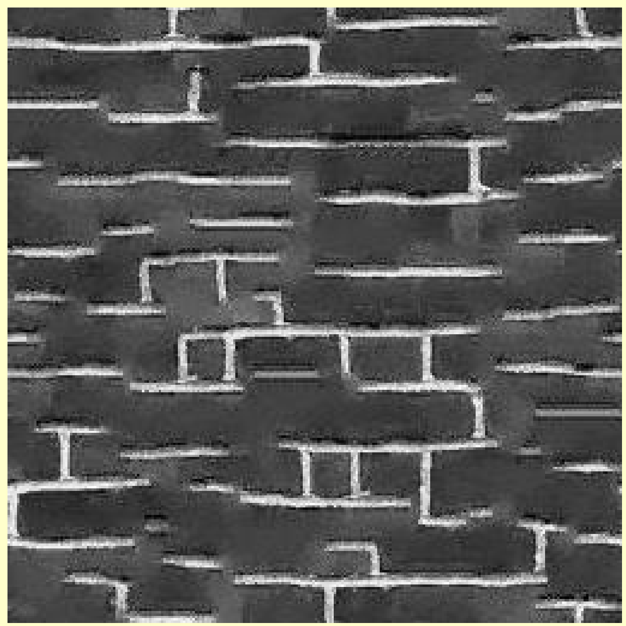


input image

Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

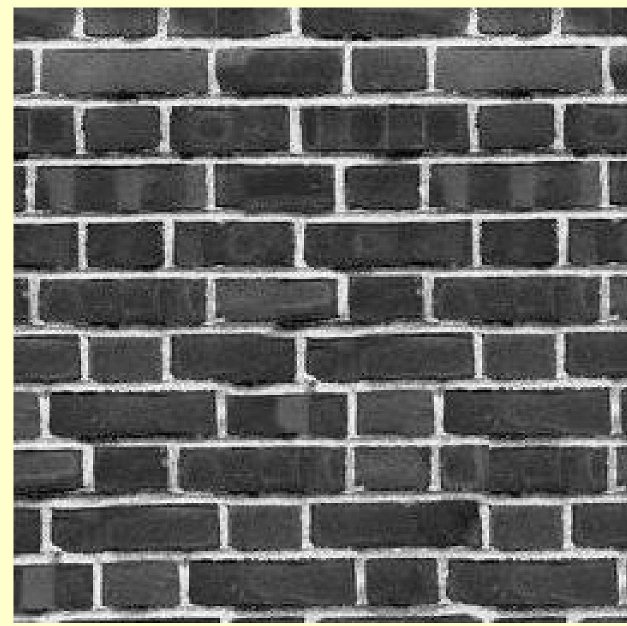
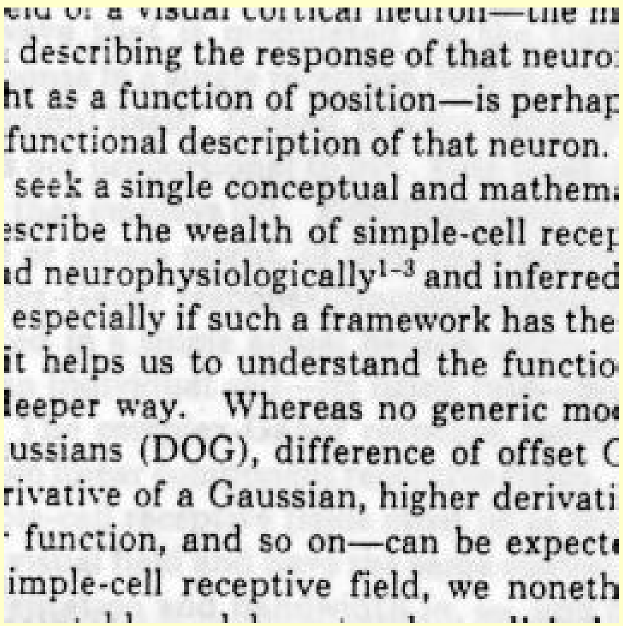
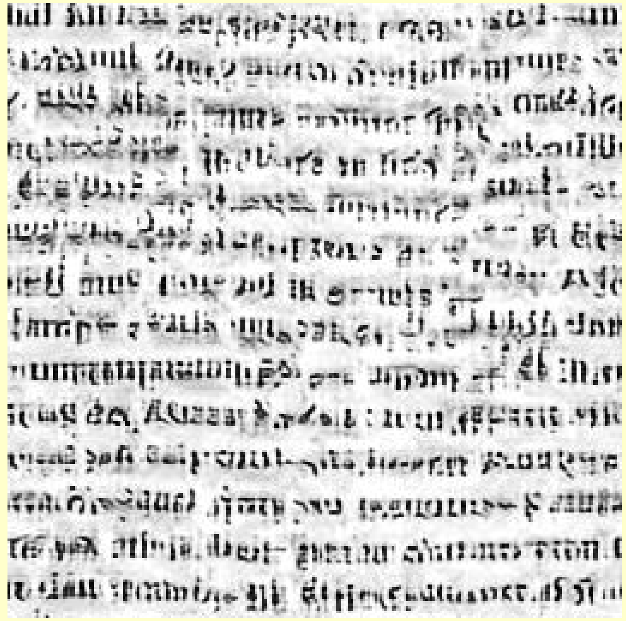


Image Quilting

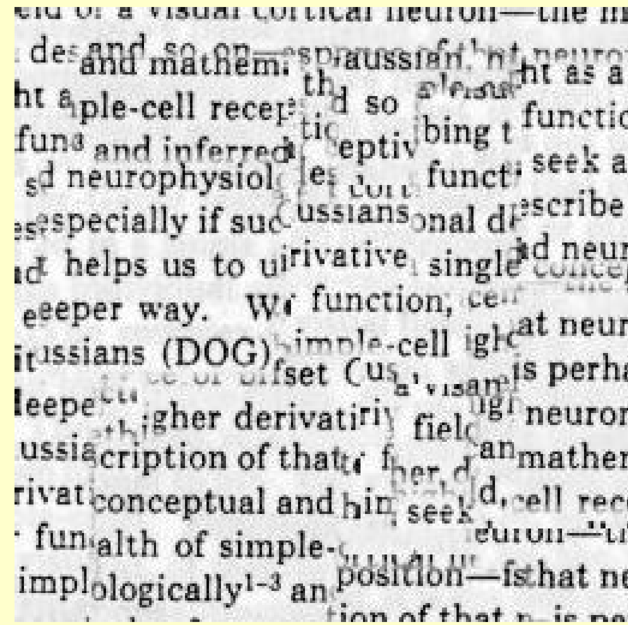
Homage to Shannon!


describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathema
escribe the wealth of simple-cell recep
d neurophysiologically[1-3] and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset C
rivative of a Gaussian, higher derivati
function, and so on—can be expect
imple-cell receptive field, we noneth

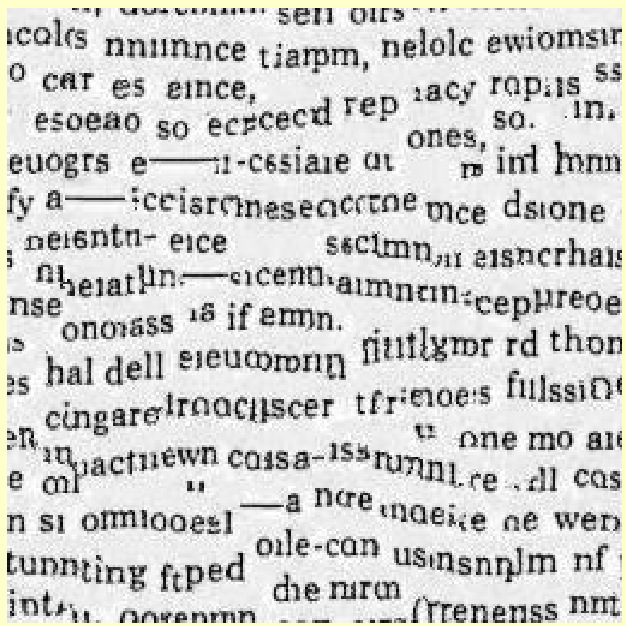
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

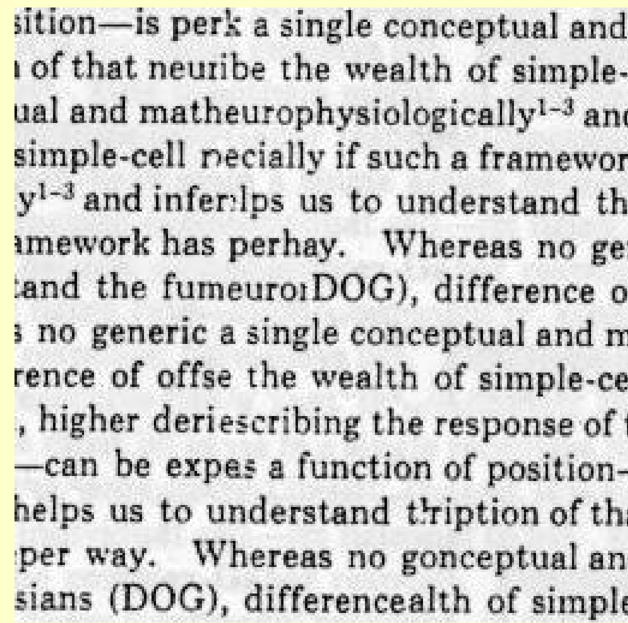
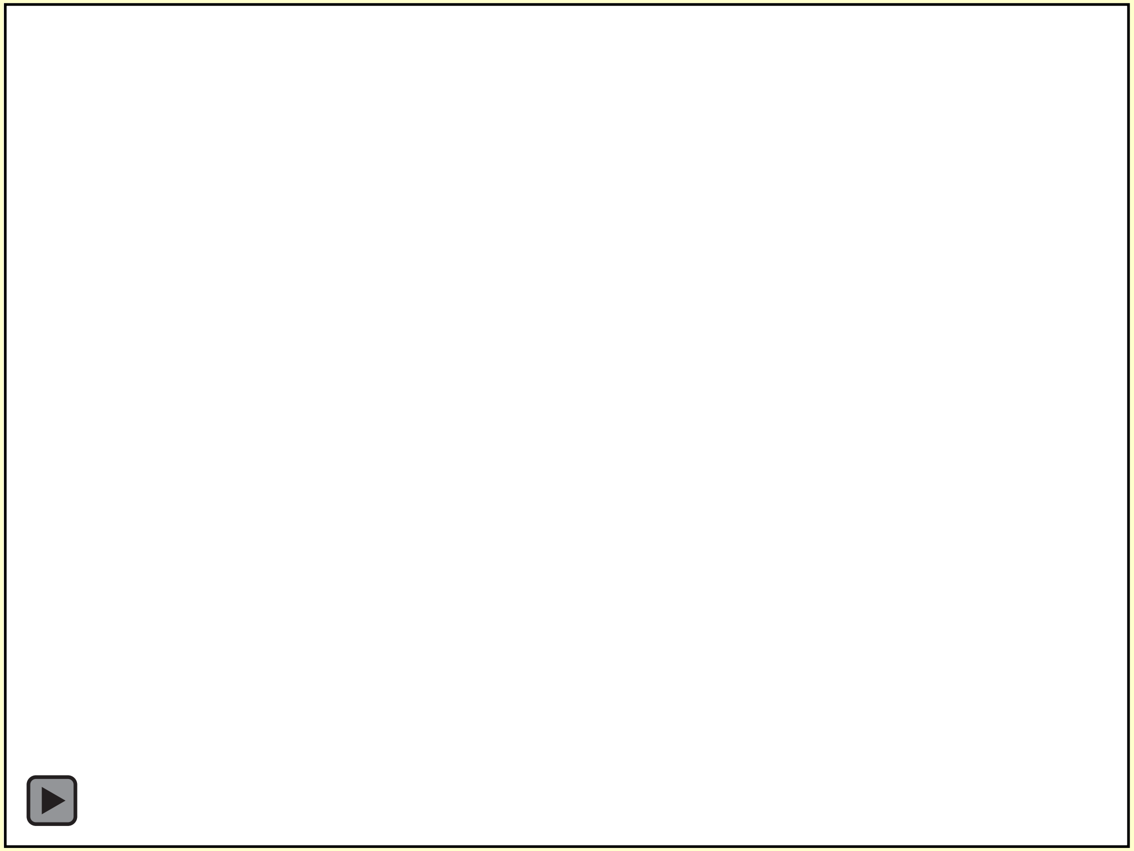
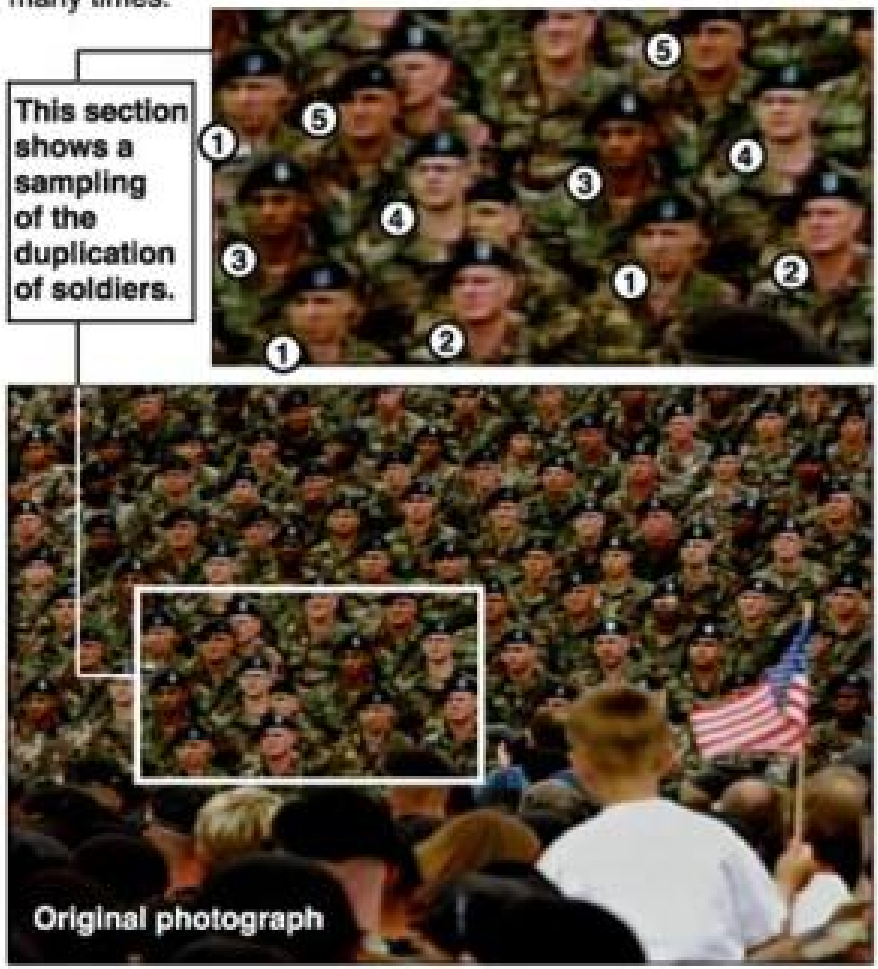


Image Quilting



Bush campaign digitally altered TV ad

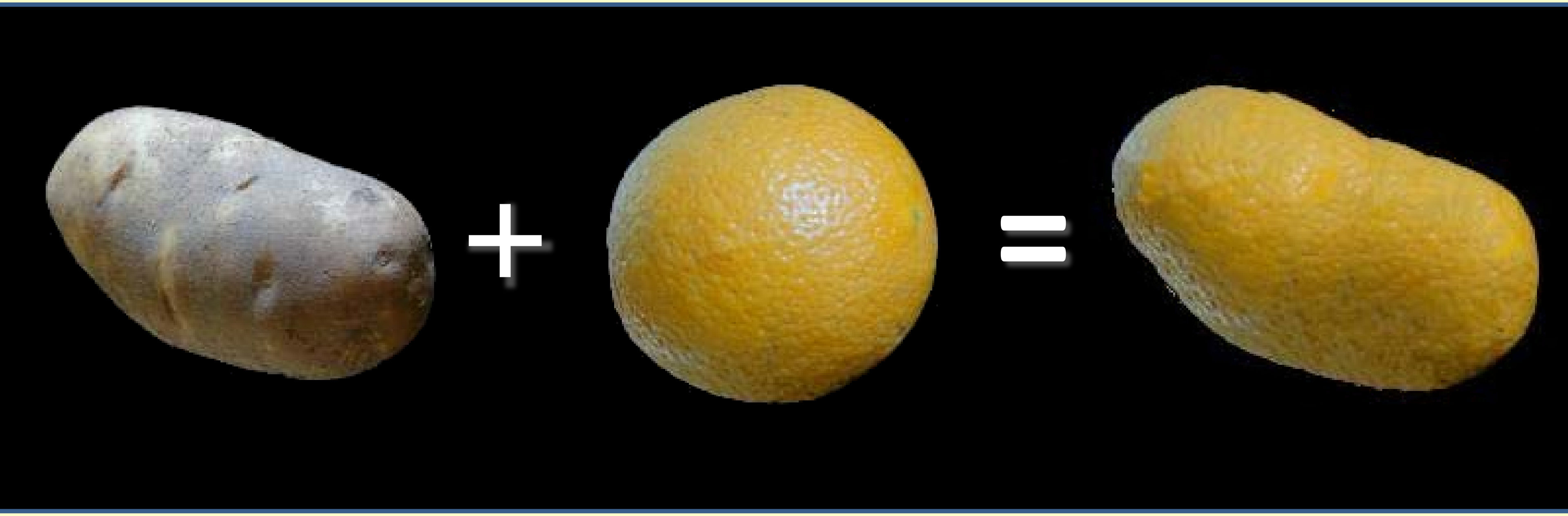
President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.



AP

Application: Texture Transfer

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



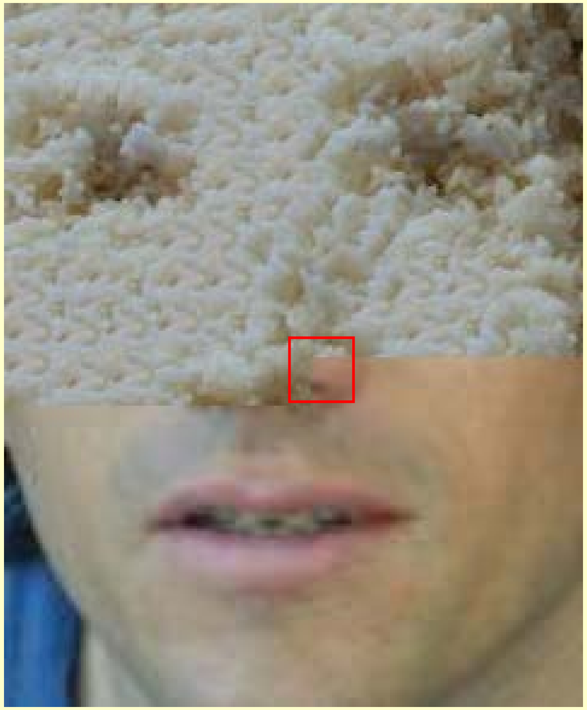
Texture Transfer

Constraint

Texture sample

Texture Transfer

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

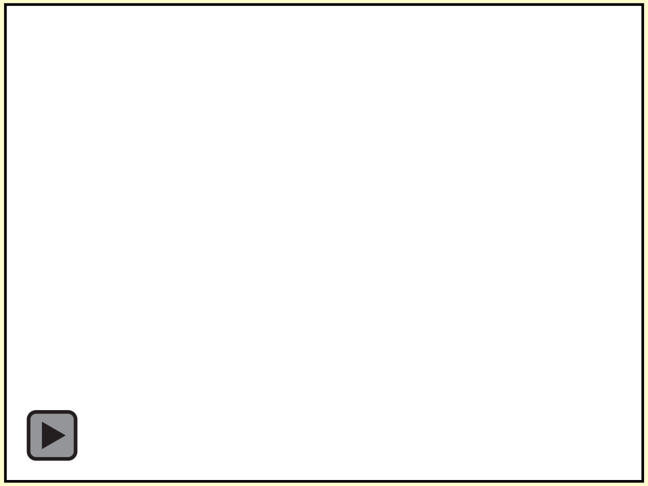


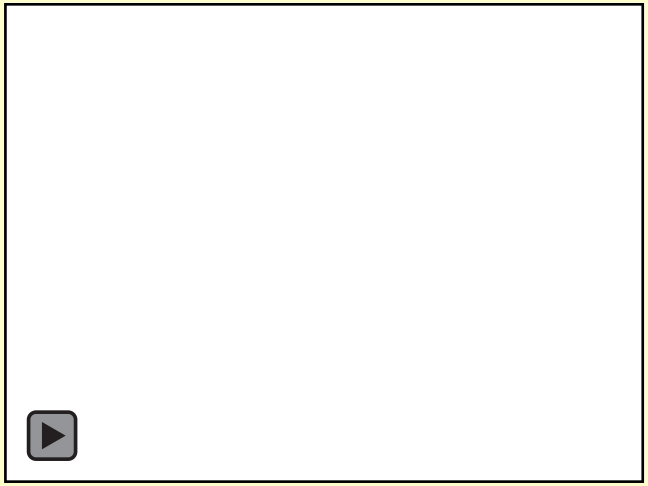
Same as texture synthesis, except an additional constraint:

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

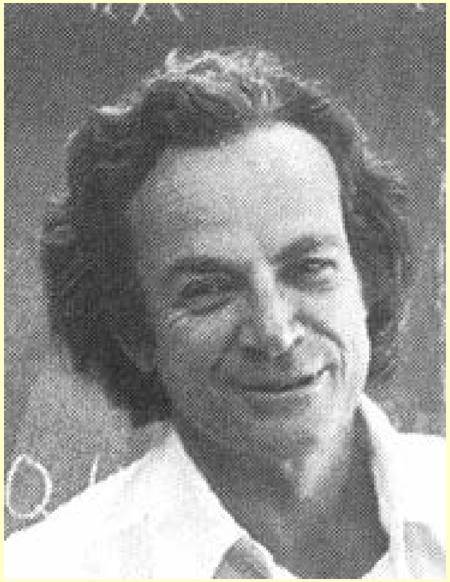
+

=



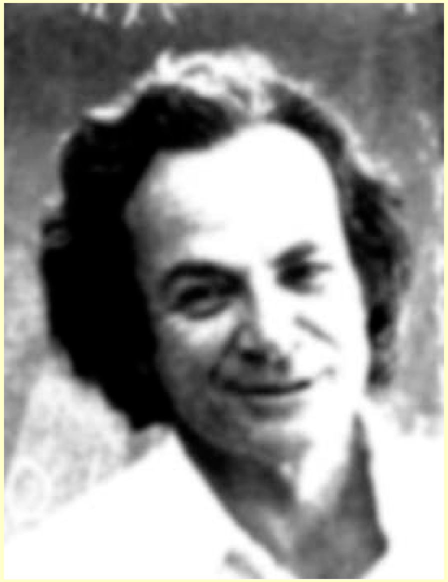


Source texture

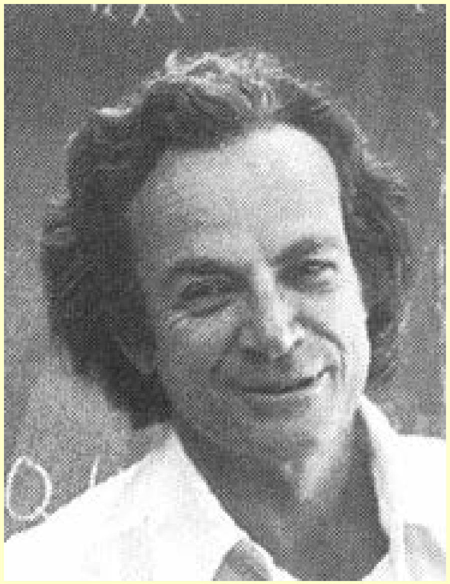


Target image

Source correspondence image

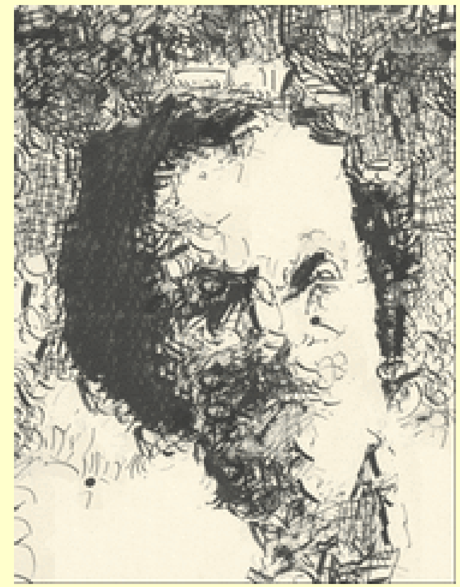


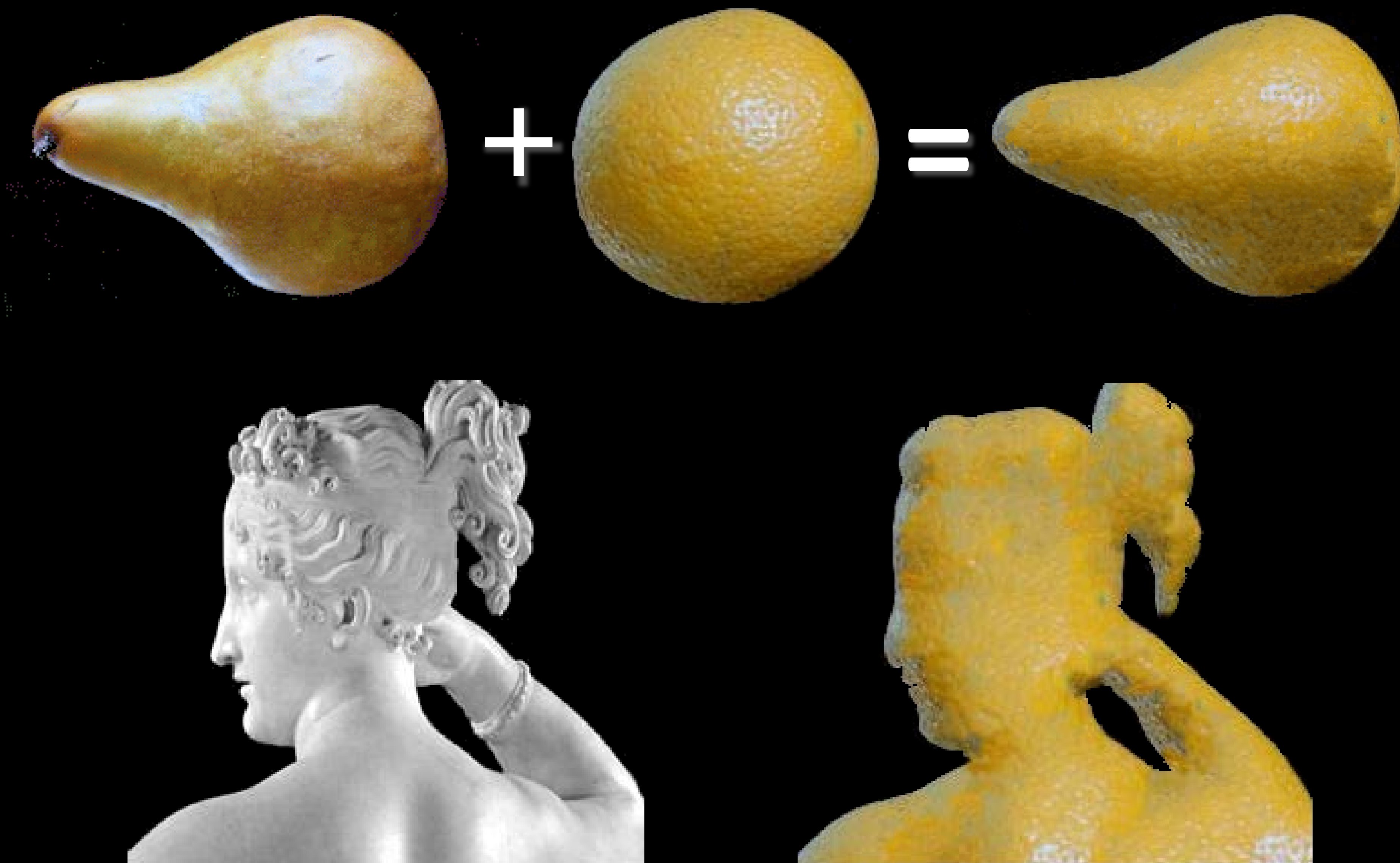
Target correspondence image



+

=







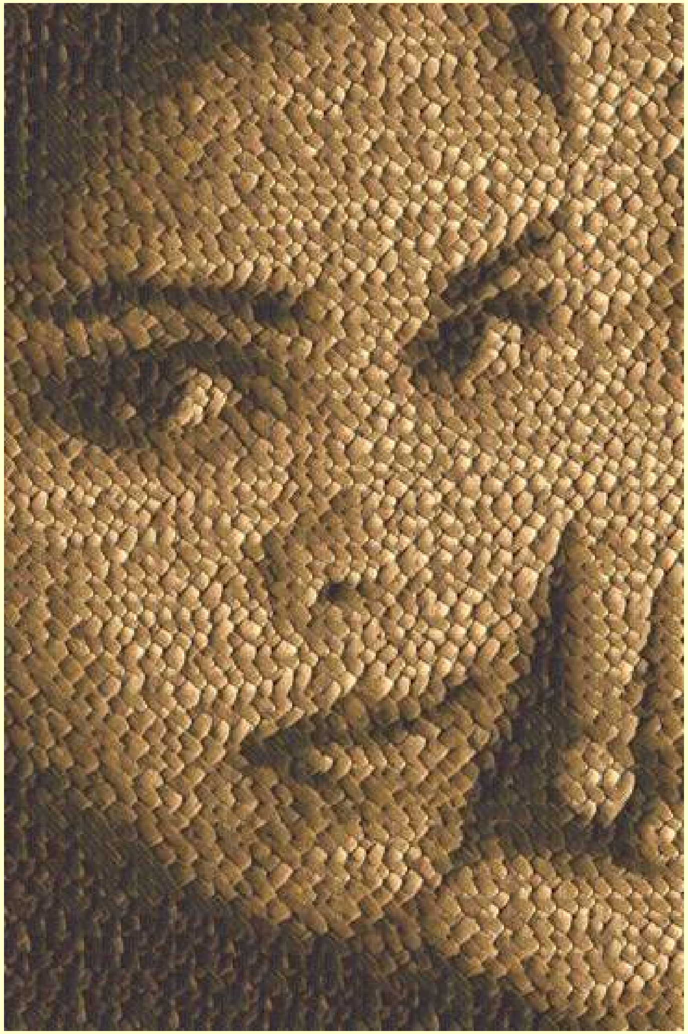
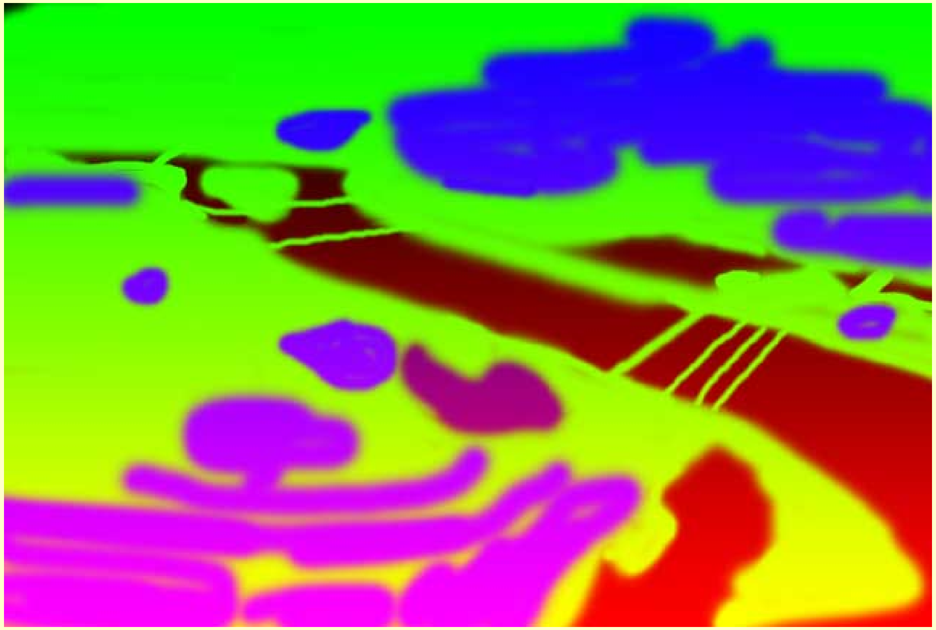


Image analogies (filter by example)

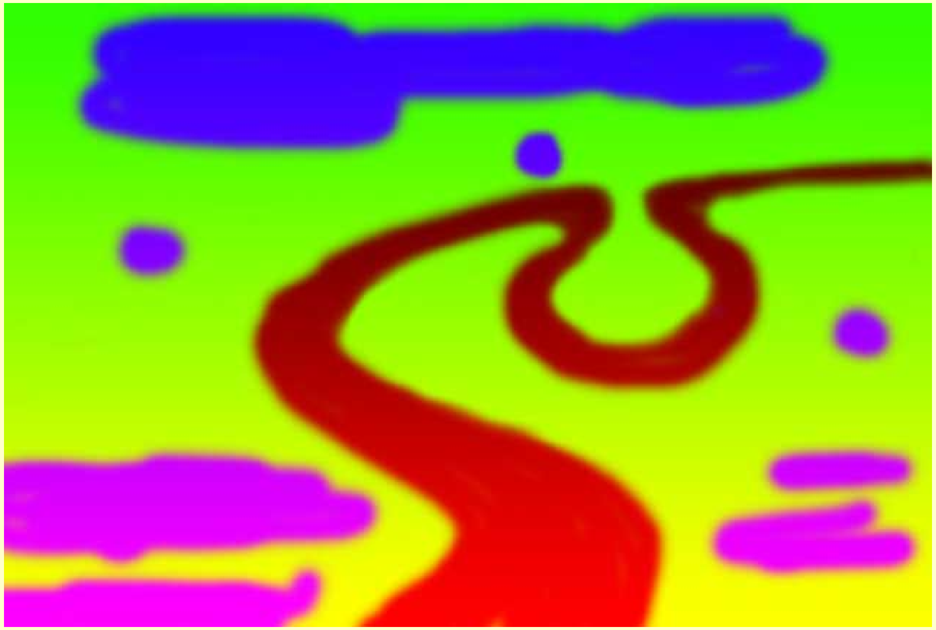
A to A' like B to B'

$$A_1,\ldots,A_n : A_1',\ldots,A_n' :: B : B'$$

texture segmentation



input



drawing with color coded textures

output

Applications – Artistic Filters (Cont.)

Target Pairs:

Source Pair:

“Texture By Numbers”

- By color-labeling source image parts a realistic synthesized image can be created

[Video]

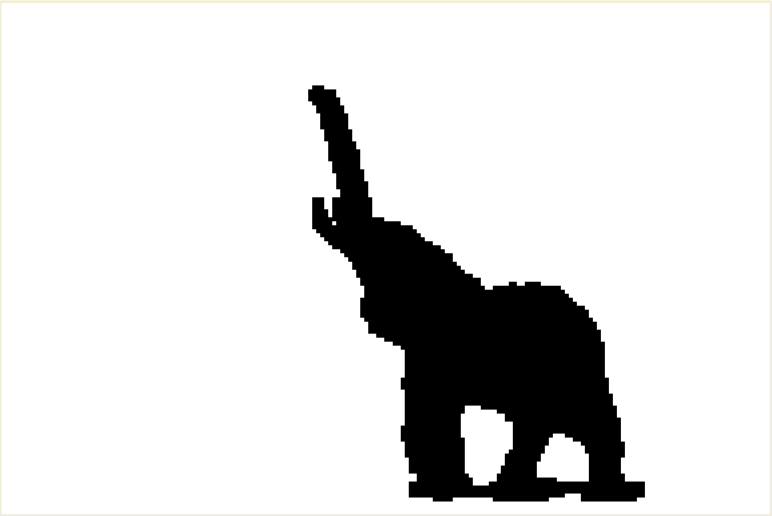
A

B

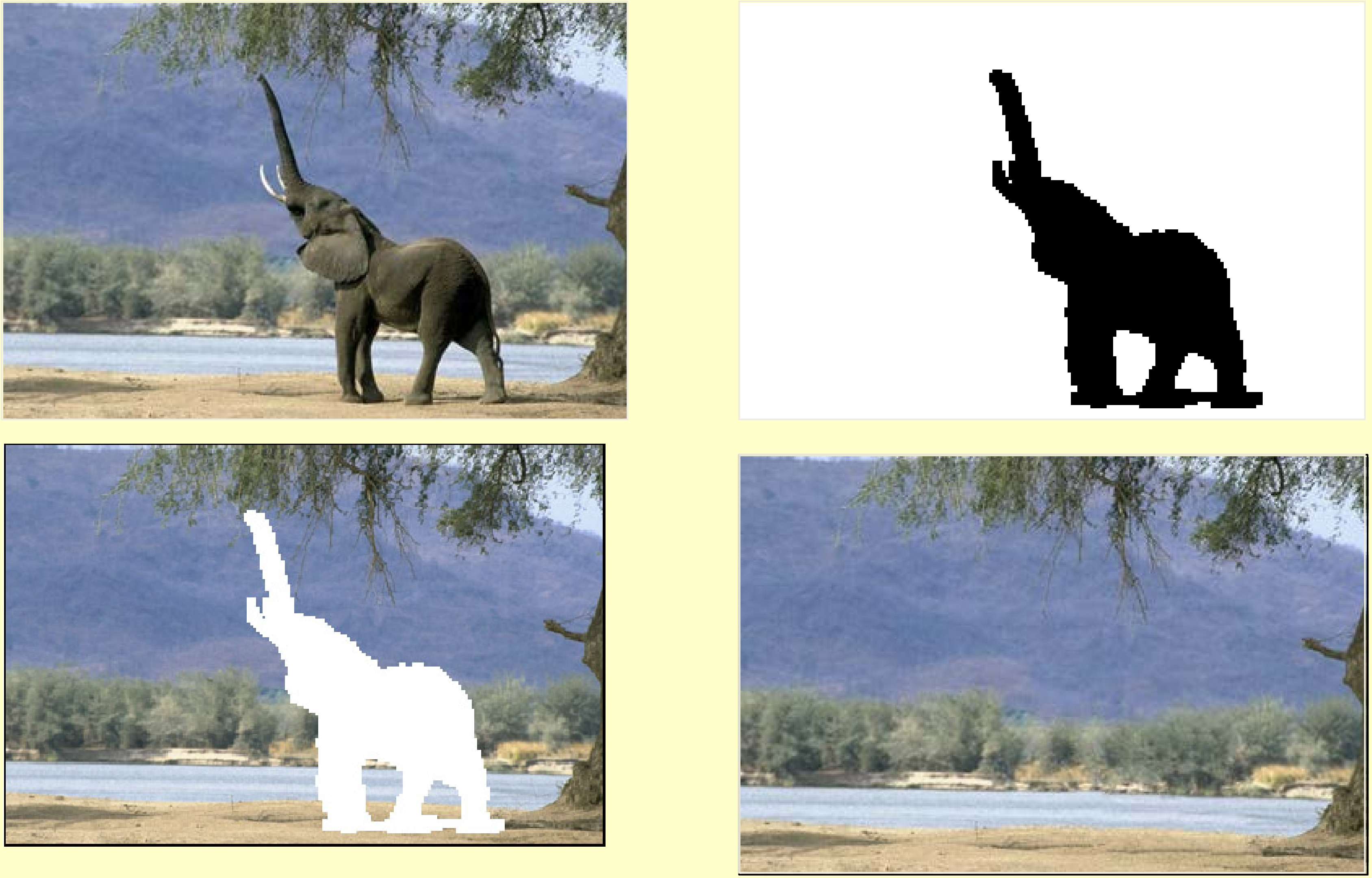
A`

B`

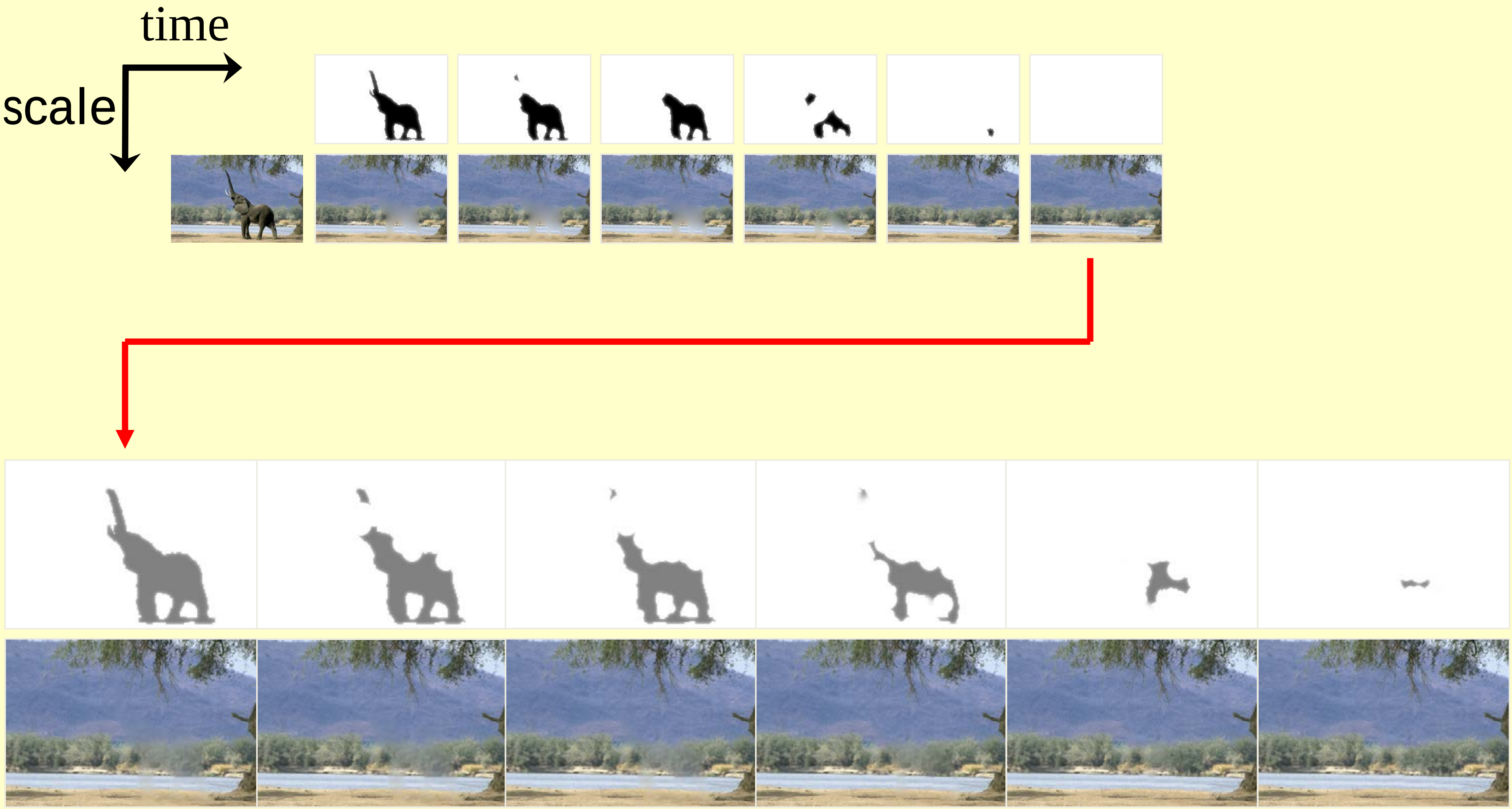
Fragment-based Image Completion (SIGGRAPH’03)



Fragment-based Image Completion (SIGGRAPH'03)

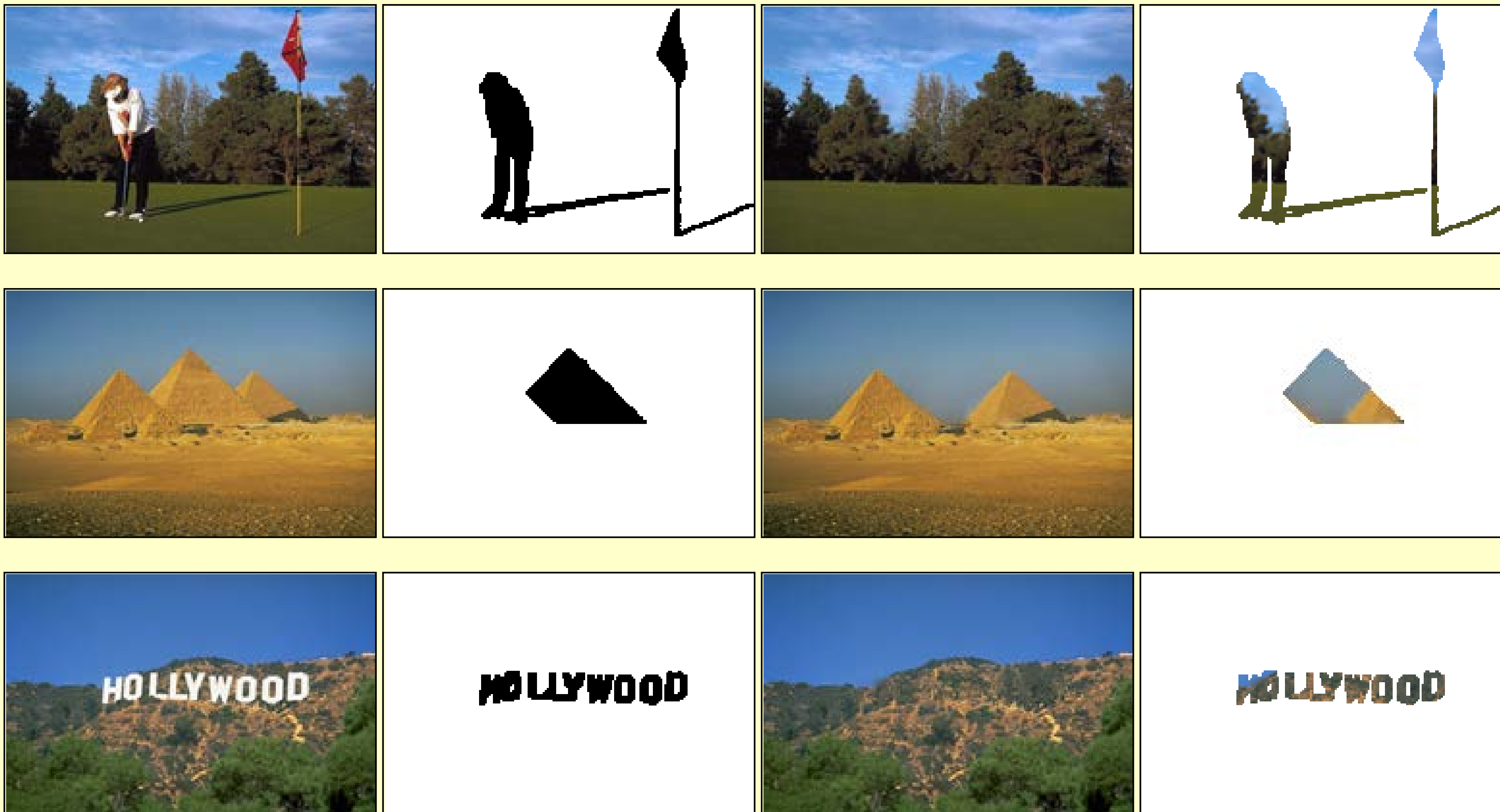


Completion process



confidence and color at different time steps and scales

Results

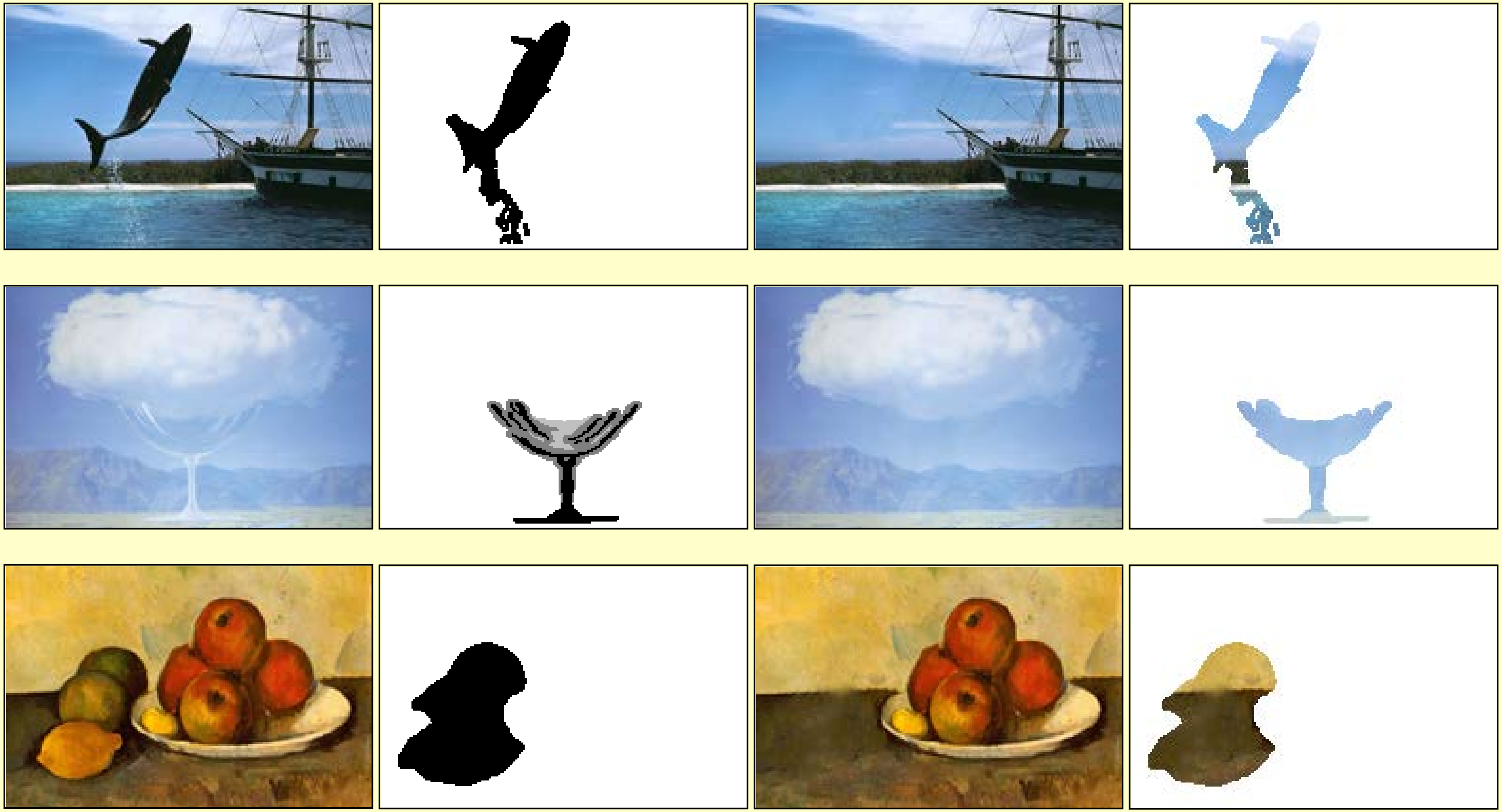


Results

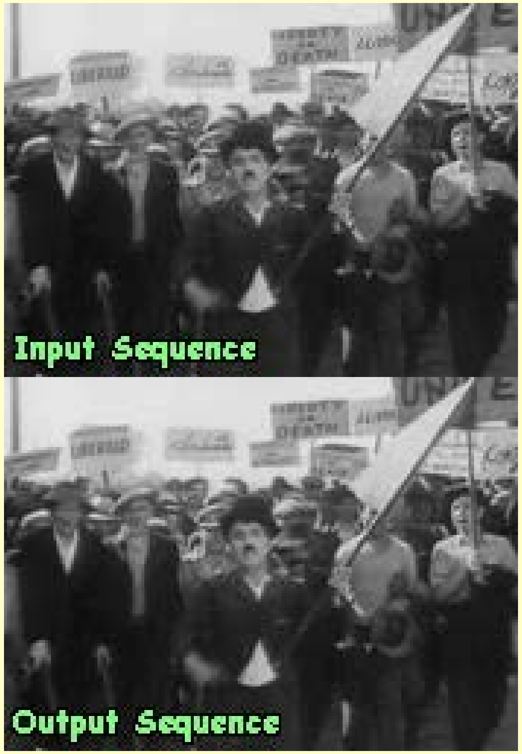
input image

completion

Results



Video Completion



. Wexler E. Shechtman M. Irani; “Space-Time Video Completion"; CVPR’04.

Thank You