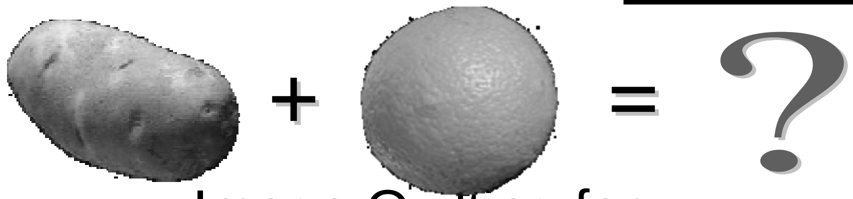
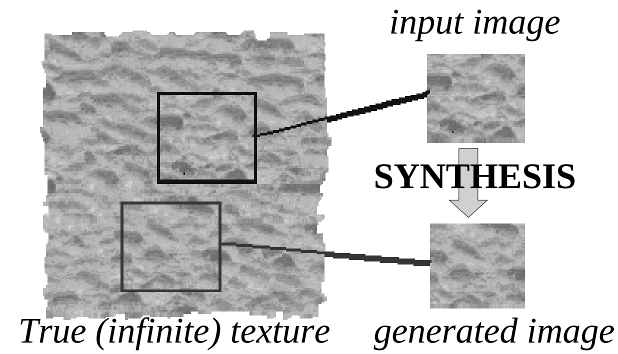


Image Quilting for Texture Synthesis & Transfer

Alexei Efros *(UC Berkeley)*

Bill Freeman *(MERL)*

The Goal of Texture Synthesis



- Given a finite sample of some texture, the goal is to synthesize other samples from that same texture
  - The sample needs to be "large enough“

The Challenge

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

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])

Efros & Leung '99

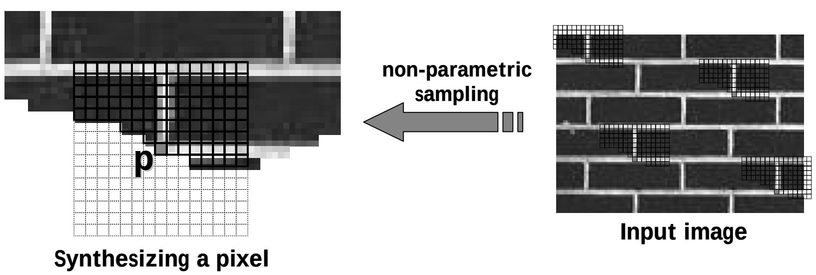
- [Shannon,'48] proposed a way to generate English-looking text using N-grams:
  - Assume a generalized Markov model
  - Use a large text to compute prob. distributions of each letter given N-1 previous letters
  - Starting from a seed repeatedly sample this Markov chain to generate new letters
  - Also 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 & Leung '99



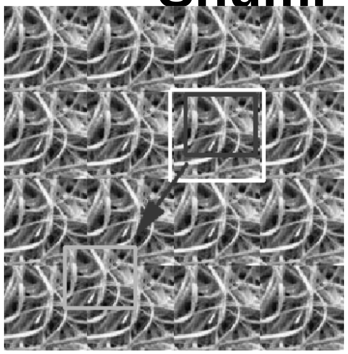
- Assuming Markov property, compute P(**p**|N(**p**))
  - Building explicit probability tables infeasible
  - Instead, let's *search the input image* for all similar neighborhoods — that's our histogram for **p**
- To synthesize **p**, just pick one match at random

Efros & Leung '99

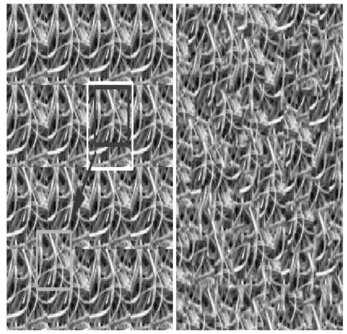
- The algorithm
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow
- Optimizations and Improvements
  - [Wei & Levoy,'00] (based on [Popat & Picard,'93])
  - [Harrison,'01]
  - [Ashikhmin,'01]

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

input



idea



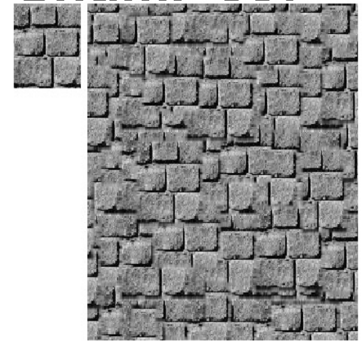
result

- **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



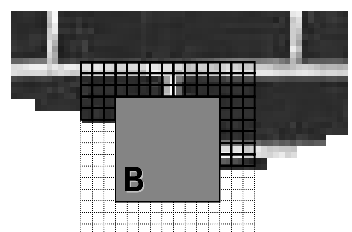
result

- Of course, doesn't work for structured textures

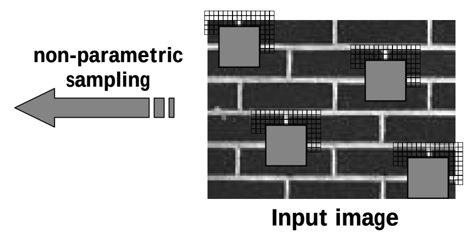
Image Quilting

- 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]

Efros & Leung '99 extended



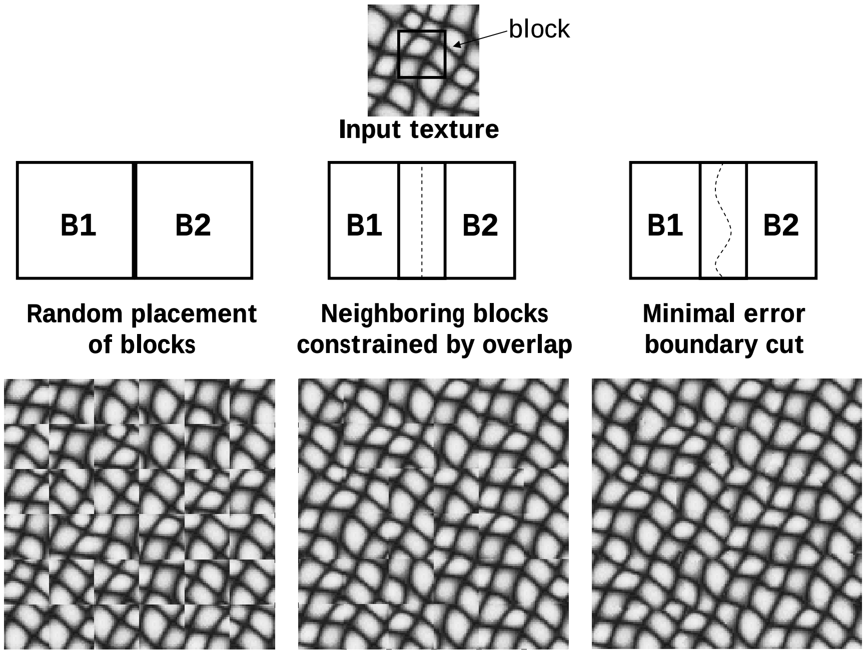
Synthesizing a block



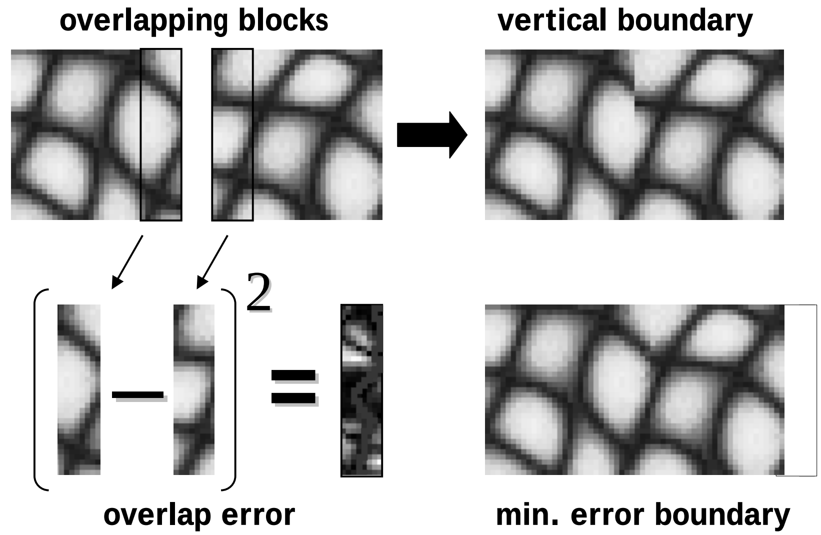
- Observation: neighbor pixels are highly correlated

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!



Minimal error boundary

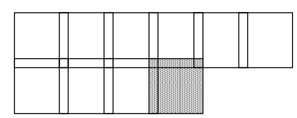


Our Philosophy

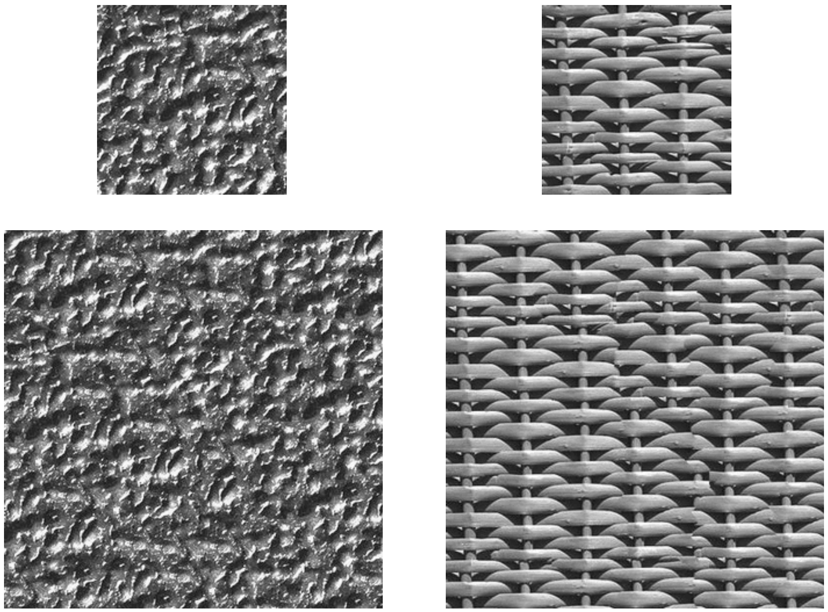
- The “Corrupt Professor’s Algorithm”:
  - Plagiarize as much of the source image as you can
  - Then try to cover up the evidence
- Rationale:
  - Texture blocks are by definition correct samples of texture so problem only connecting them together

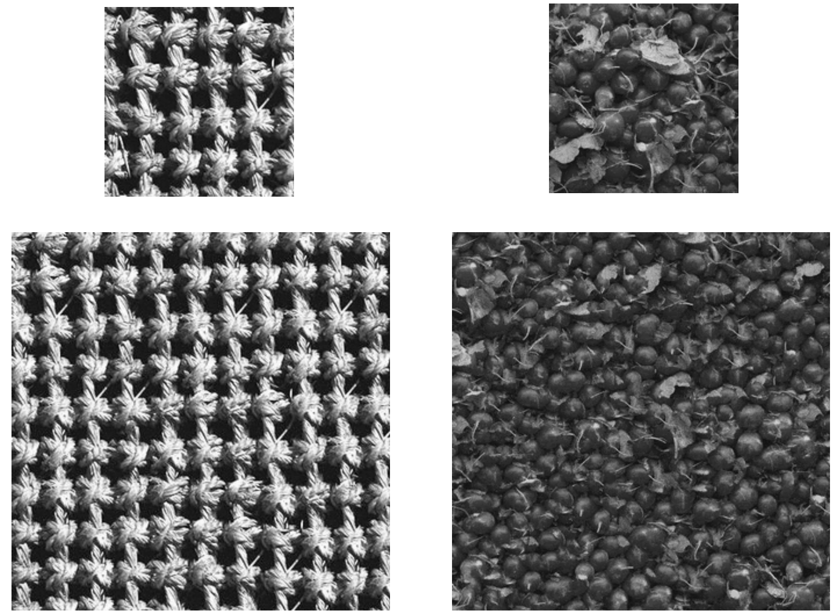
Algorithm

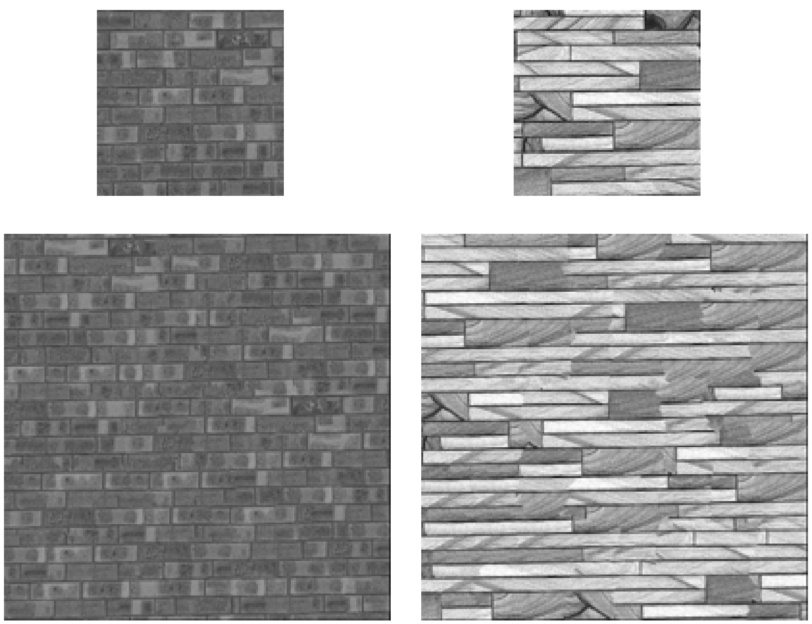
- Pick size of block and size of overlap
- Synthesize blocks in raster order

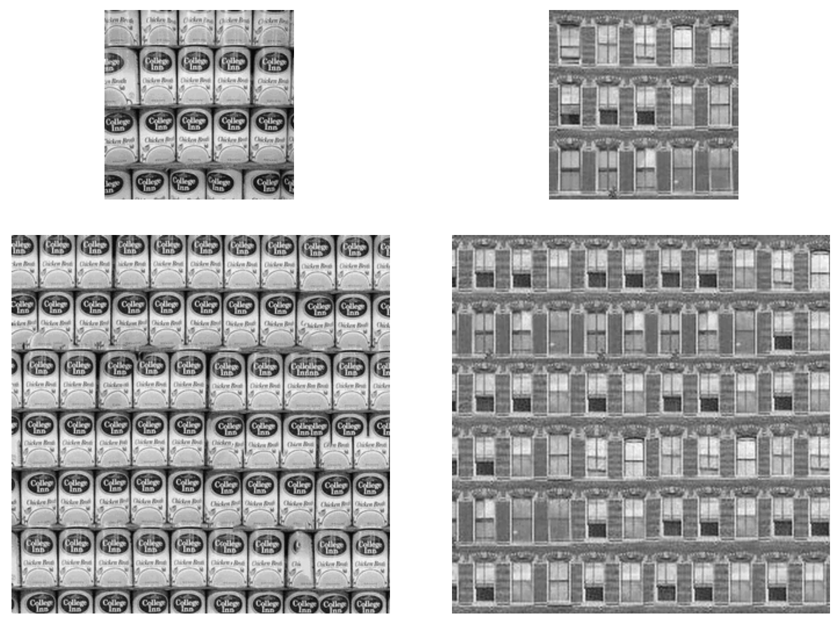


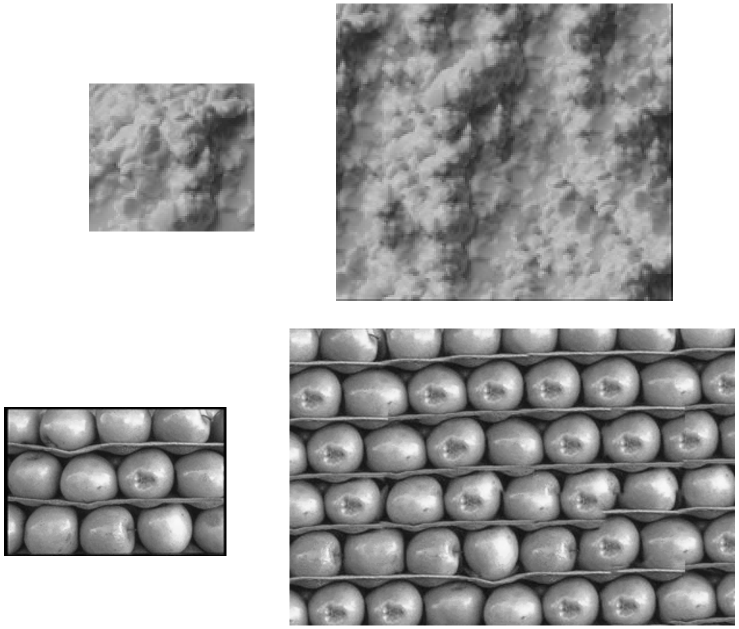
- Search input texture for block that satisfies overlap constraints (above and left)
  - Easy to optimize using NN search [Liang et.al., ’01]
- Paste new block into resulting texture
  - use dynamic programming to compute minimal error boundary cut

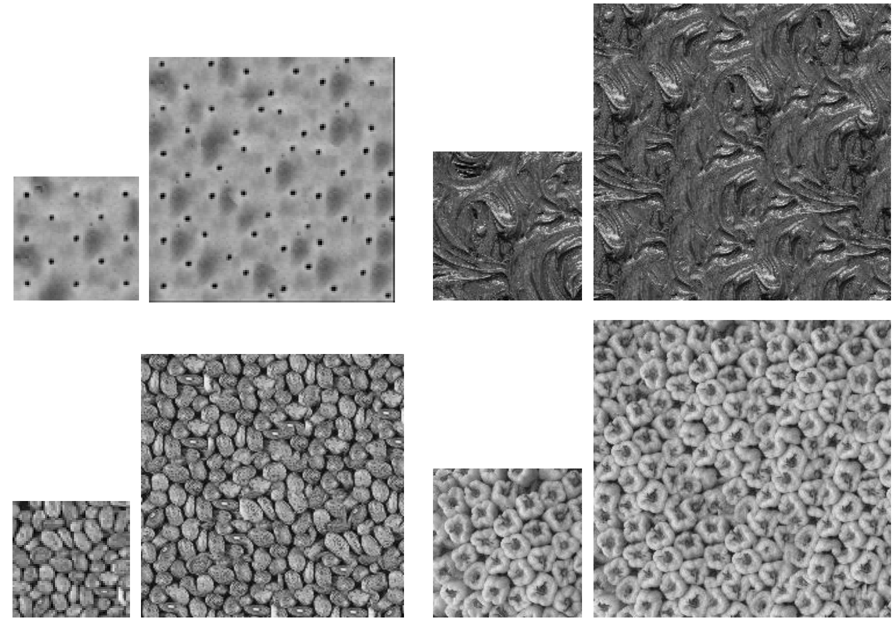


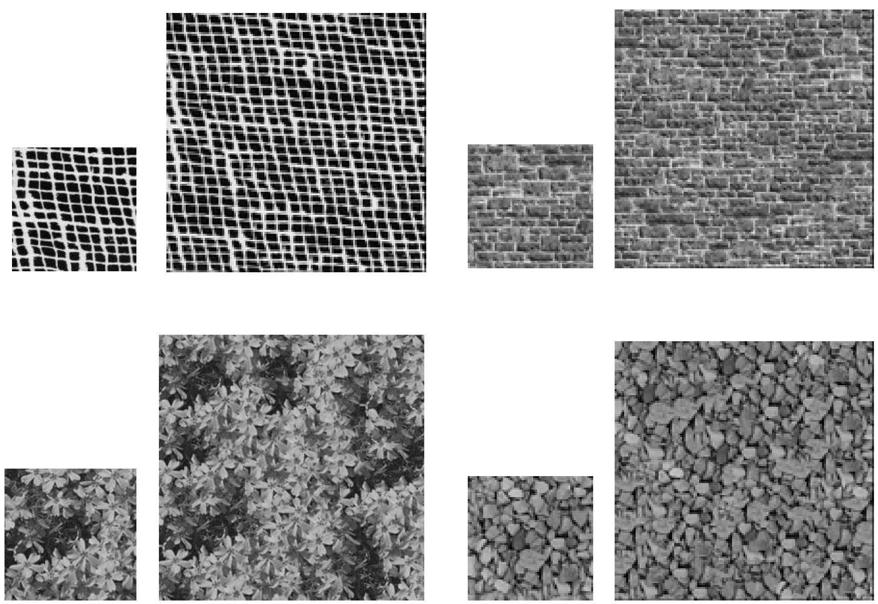




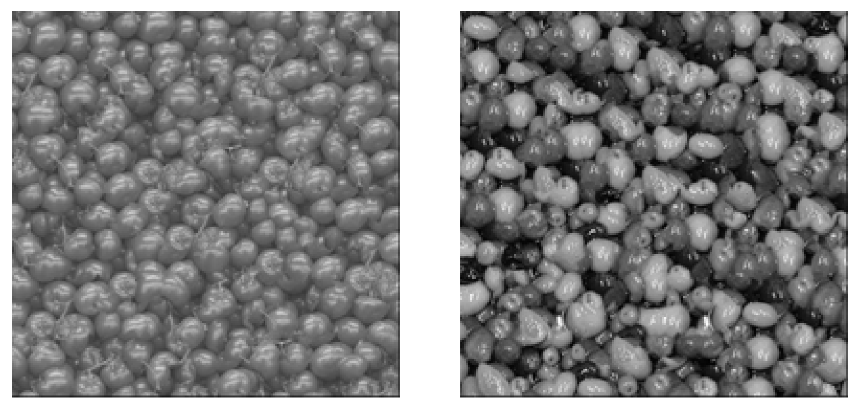

College Inn
Chicken Broth





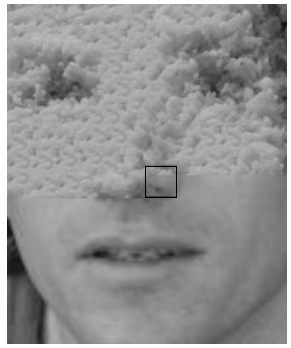


Failures
(Chernobyl Harvest)

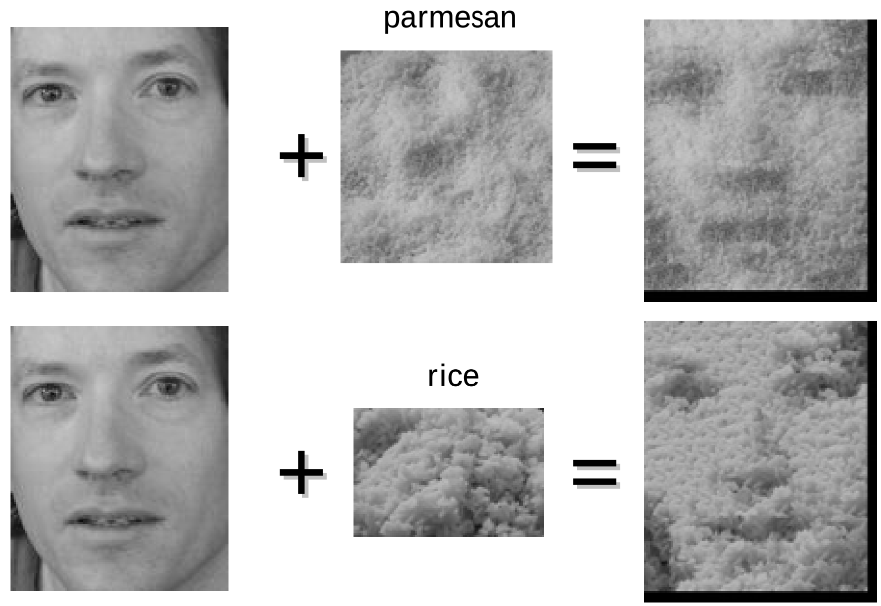


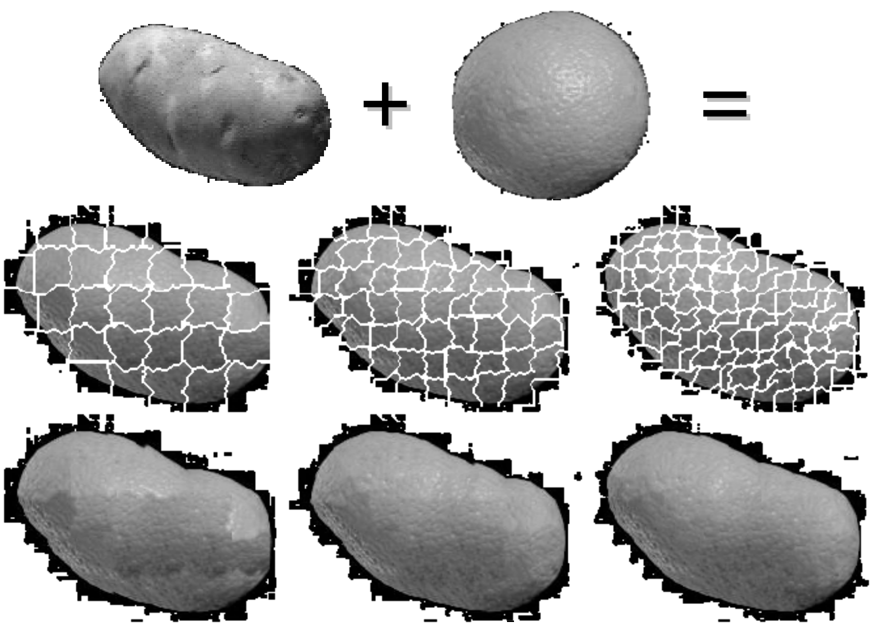
Texture Transfer

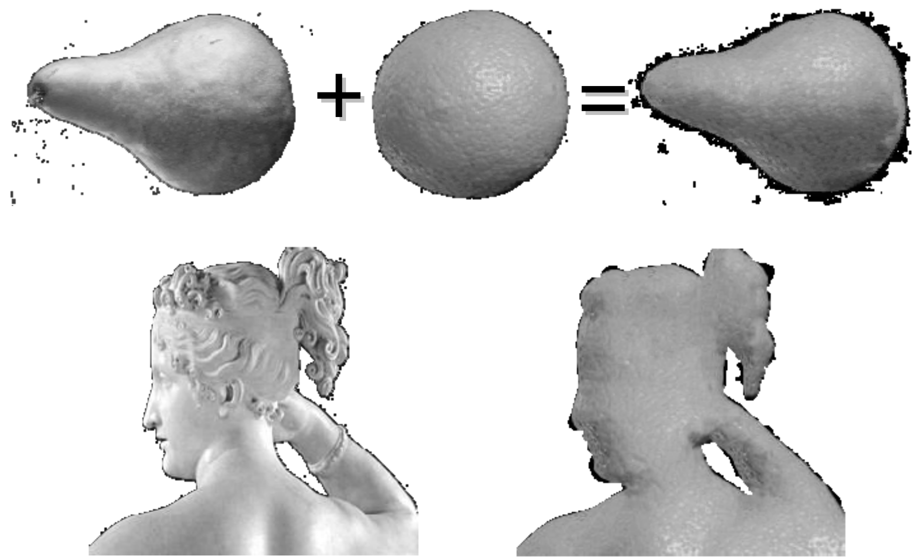
- Take the texture from one object and “paint” it onto another object
  - This requires separating texture and shape
  - That’s HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading



Then, just add another constraint when sampling: similarity to underlying image at that spot

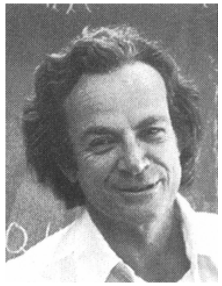






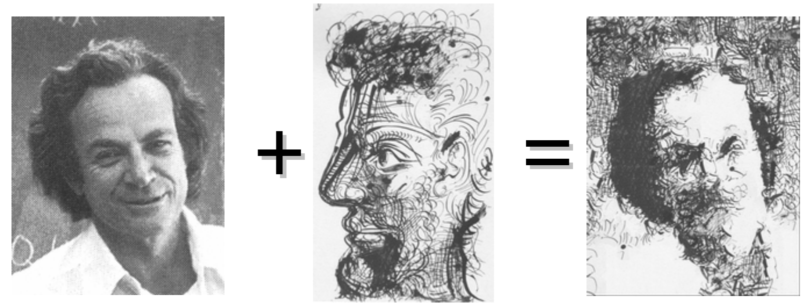
Source texture

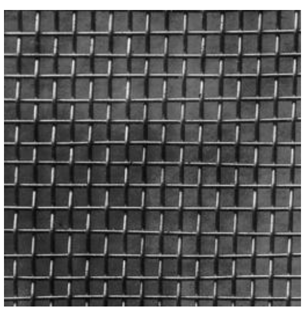
Target image



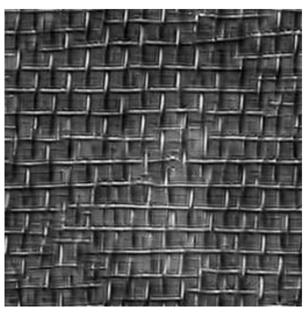
Source correspondence image

Target correspondence image

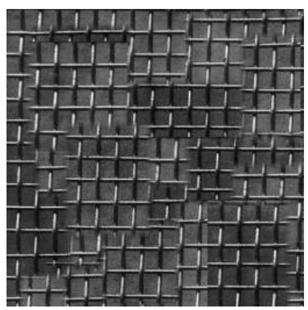




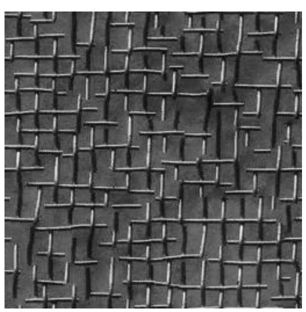
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

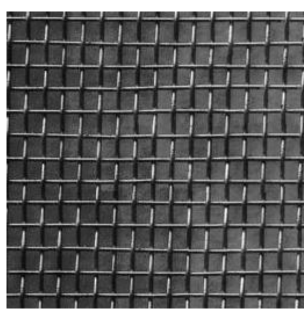
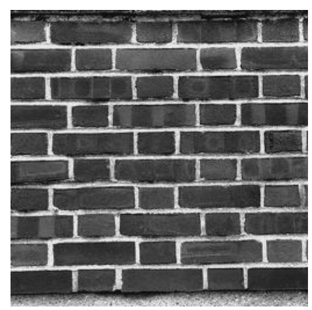
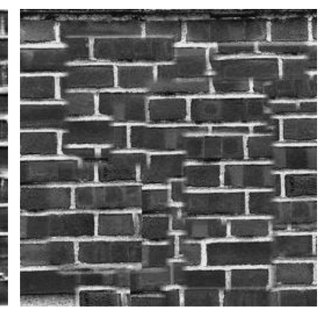


Image Quilting

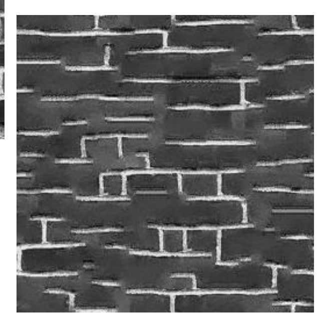


input image

Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

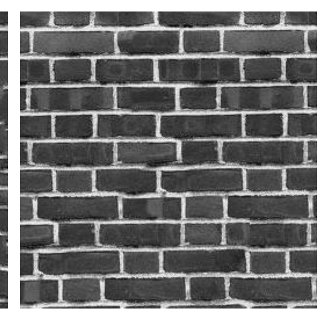
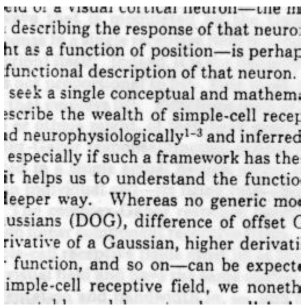
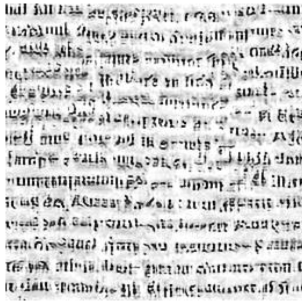


Image Quilting

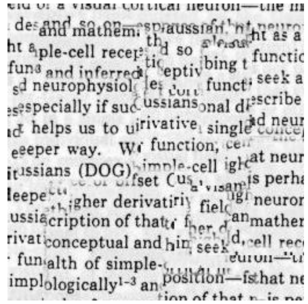
Homage to Shannon!



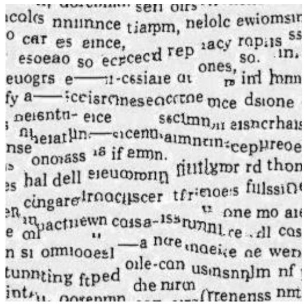
input image



Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy

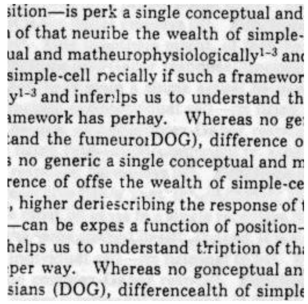


Image Quilting

Conclusion

- 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