Textures & Image-Based Lighting

Digital Visual Effects, Spring 2005

Yung-Yu Chuang

2005/6/15

with slides by Alex Efros, Li-Yi Wei and Paul Debevec

Announcements

- Final project presentation on 6/28 1:30pm in Room 101
- What to hand in?

Outline

- Texture synthesis
- Acceleration by multi-resolution and TSVQ
- Patch-based texture synthesis
- Image analogies
- Image-based lighting

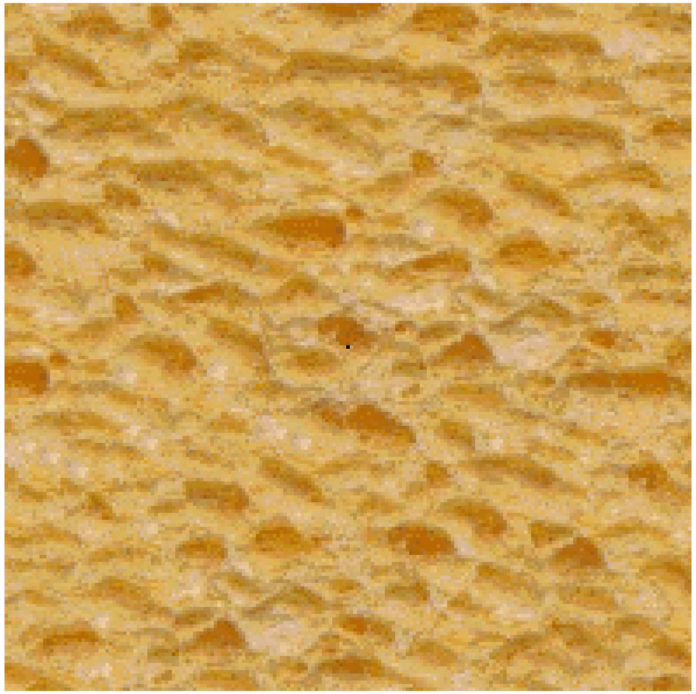
Texture synthesis

Texture synthesis

input image

synthesis

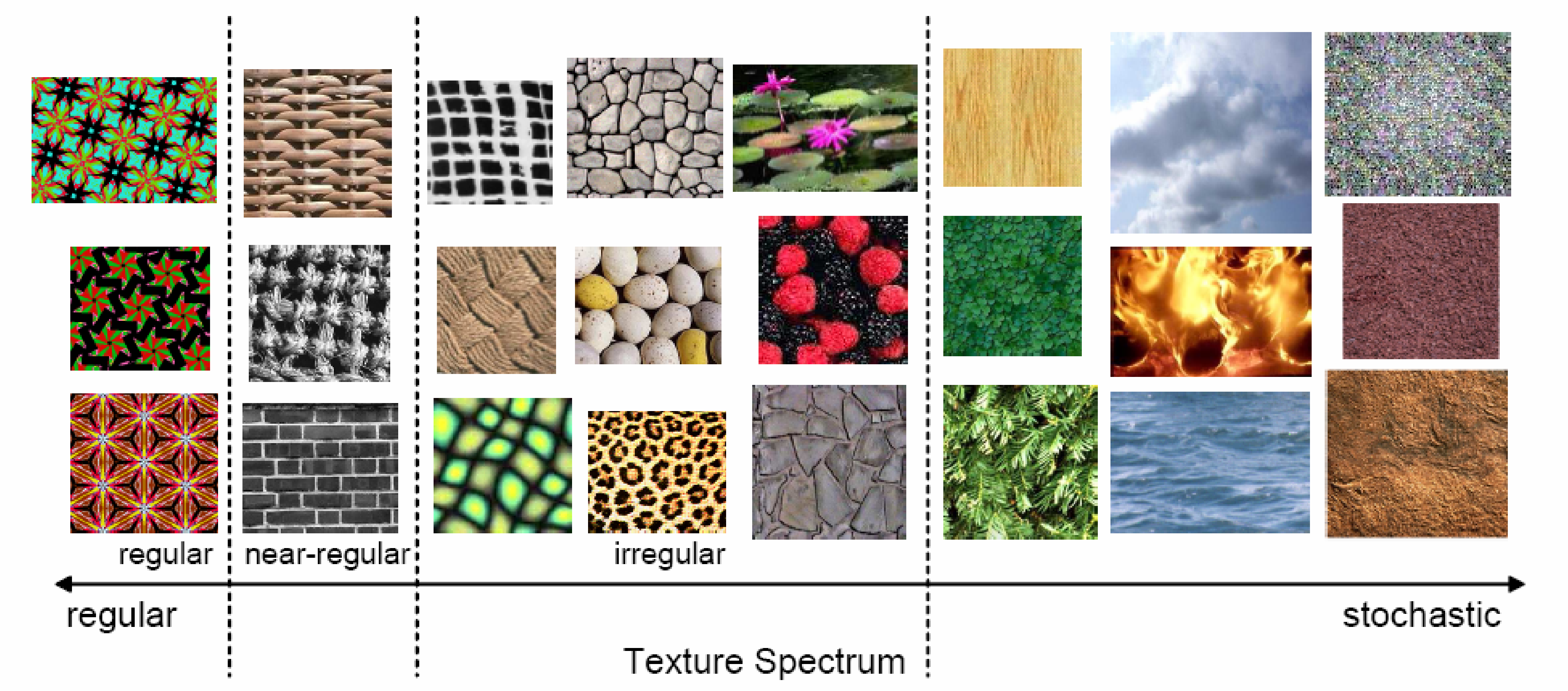
generated image



- 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"

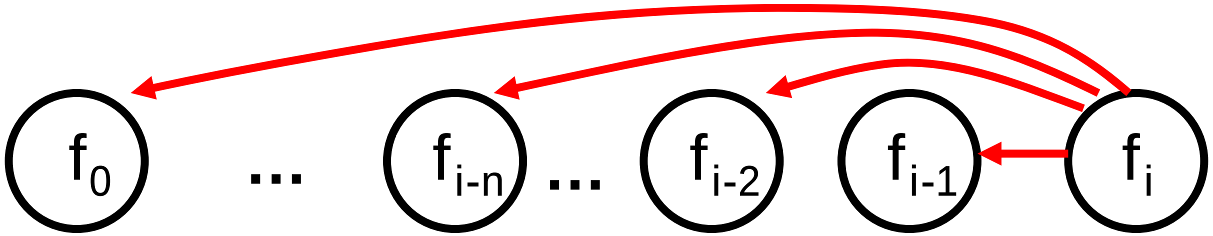
The challenge

- How to capture the essence of texture?
- Need to model the whole spectrum: from repeated to stochastic texture

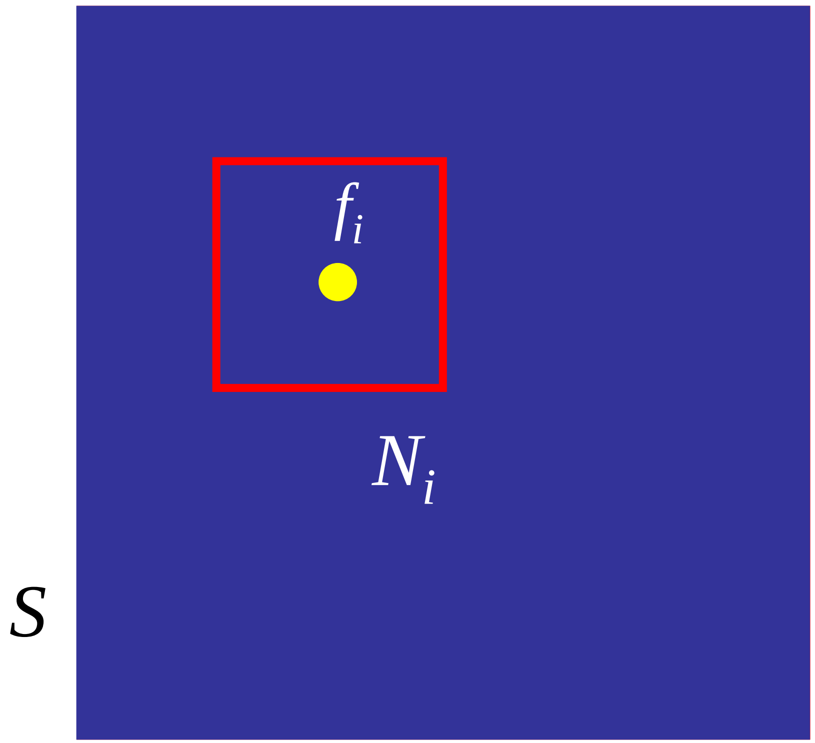


Markov property

- $P(f_i | f_{i-1}, f_{i-2}, f_{i-3}, \ldots f_0) = P(f_i | f_{i-1}, f_{i-2}, \ldots f_{i-n})$



- $P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$



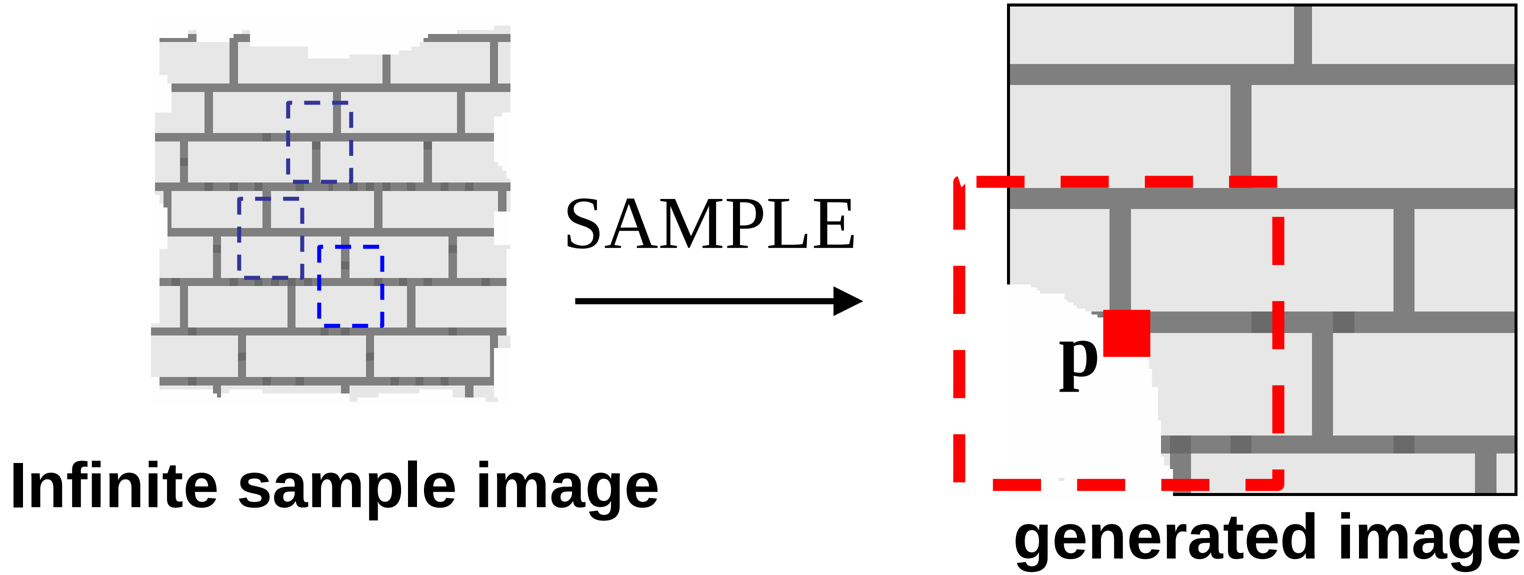
Motivation from language

- [Shannon'48] proposed a way to generate English-looking text using N-grams:
  - Assume a generalized Markov model
  - Use a large text to compute probability distributions of each letter given N-1 previous letters
    - precompute or sample randomly
  - Starting from a seed repeatedly sample this Markov chain to generate new letters
  - One can use whole words instead of letters too.

Mark V. Shaney (Bell Labs)

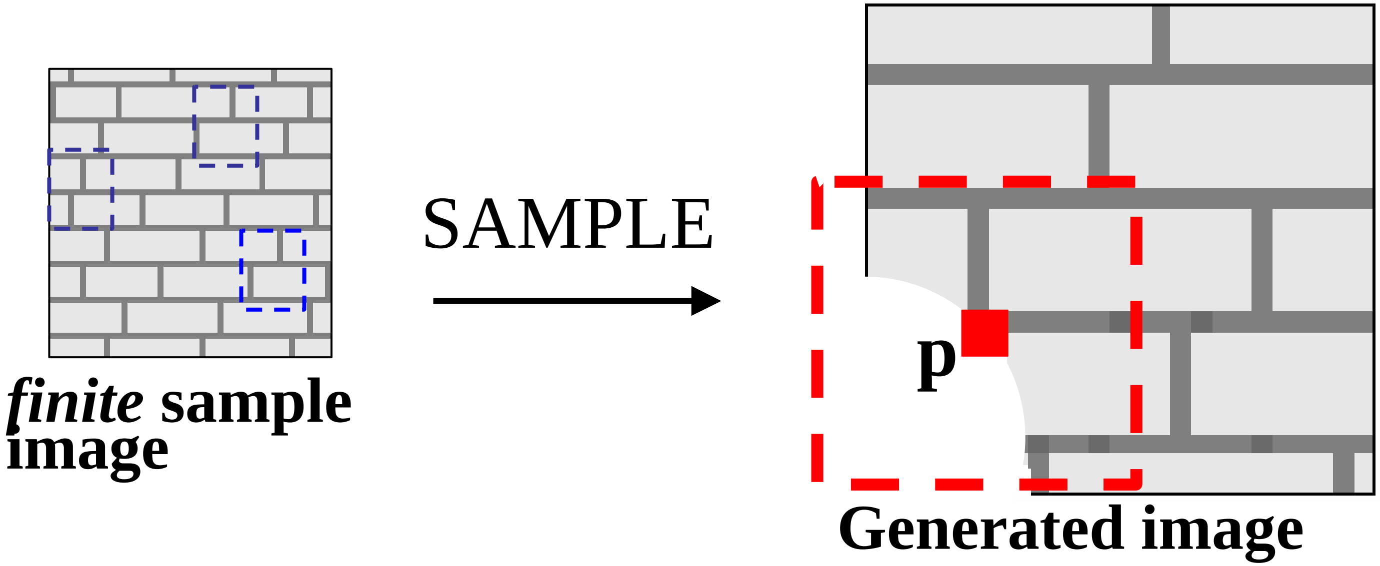
- Results (using alt.singles corpus):
  - *"One morning I shot an elephant in my arms and kissed him."*
  - *"I spent an interesting evening recently with a grain of salt"*
- Notice how well local structure is preserved!
  - Now let's try this in 2D…

Ideally



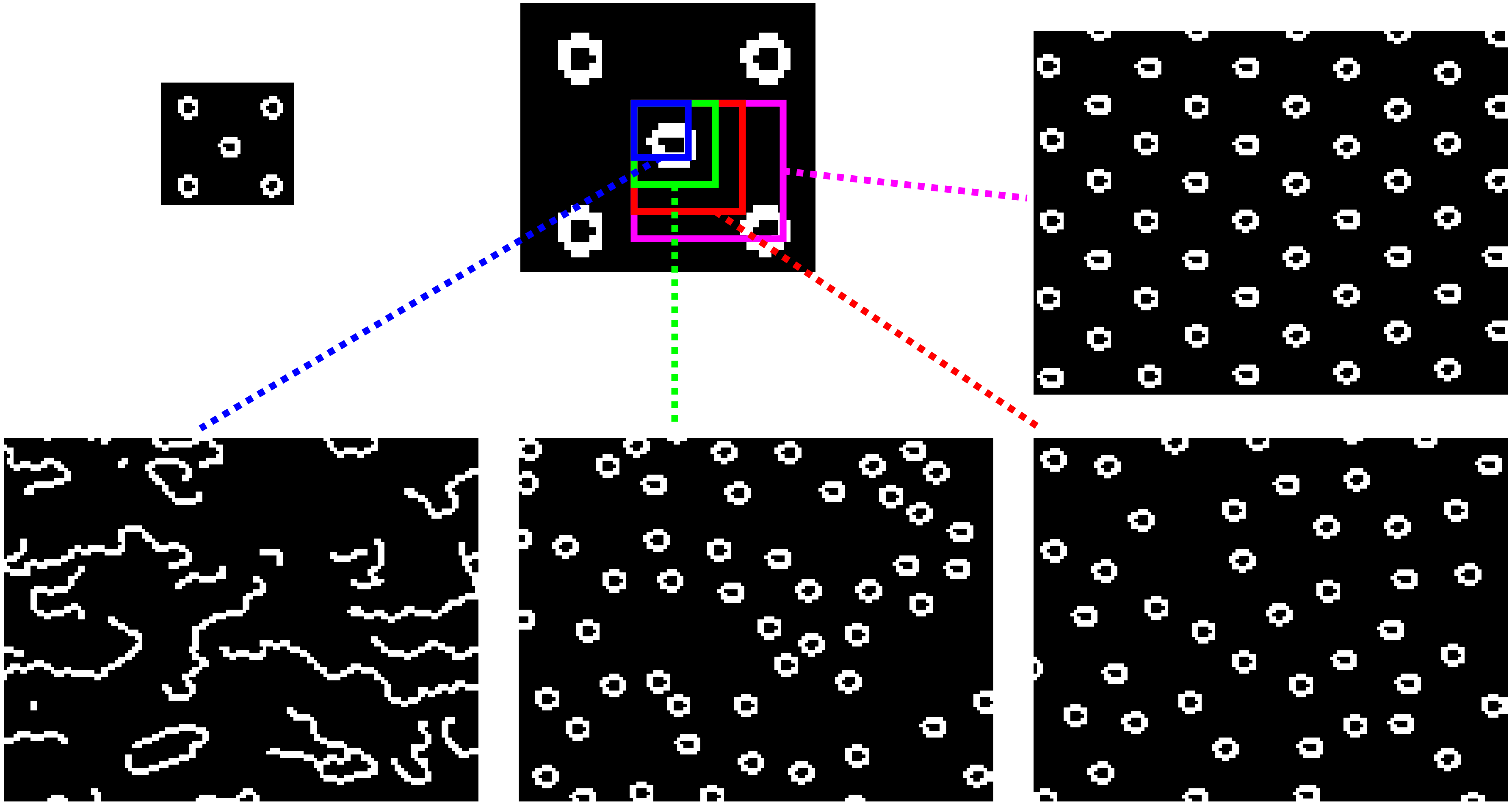
- Assuming Markov property, what is conditional probability distribution of p, given the neighbourhood window?
- Instead of constructing a model, let's directly search the input image for all such neighbourhoods to produce a histogram for p
- To synthesize p, just pick one match at random

In reality

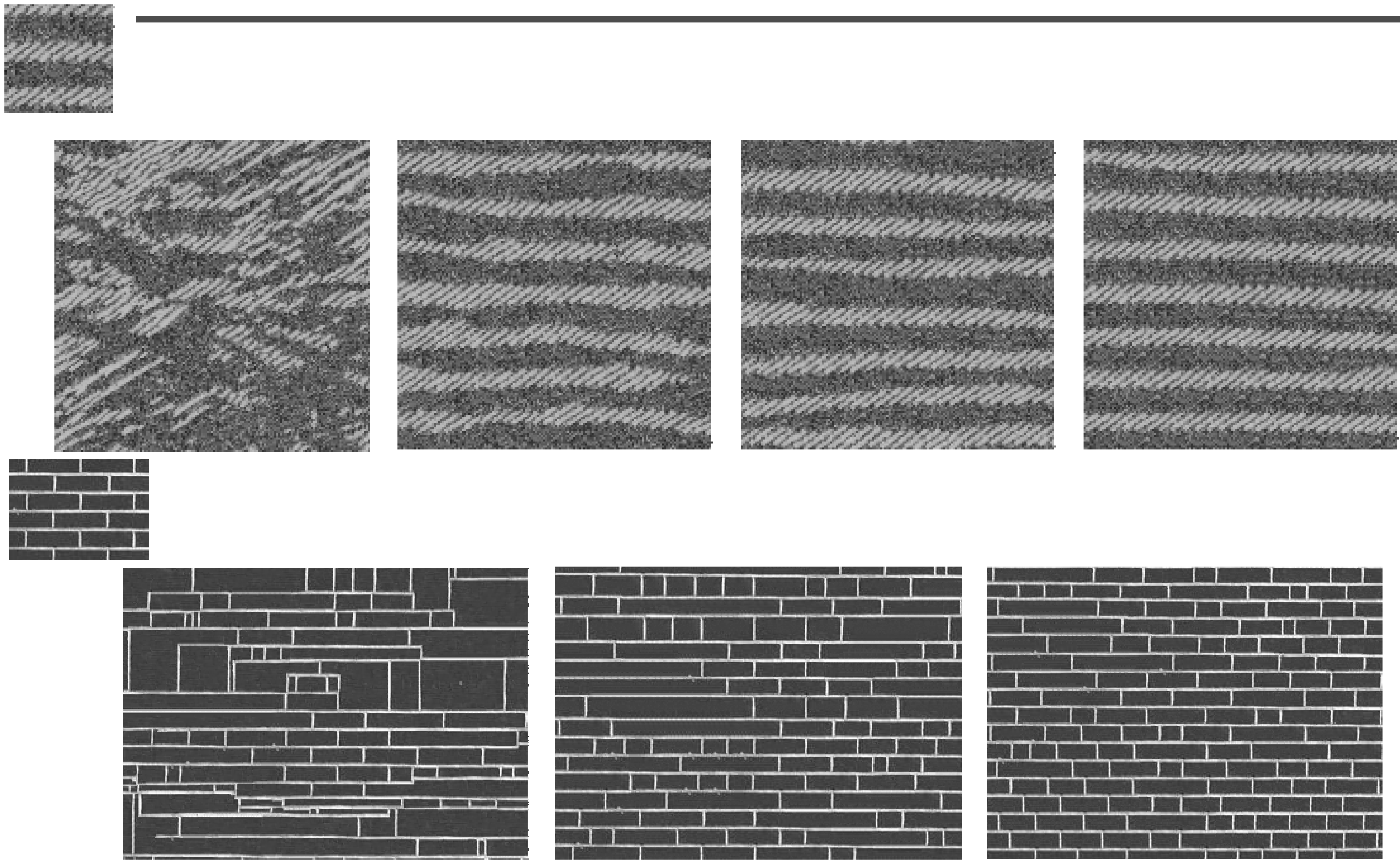


- However, since our sample image is finite, an exact neighbourhood match might not be present
- So we find the best match using SSD error (weighted by a Gaussian to emphasize local structure), and take all samples within some distance from that match
- Using *Gaussian-weighted* SSD is very important

Neighborhood size matters



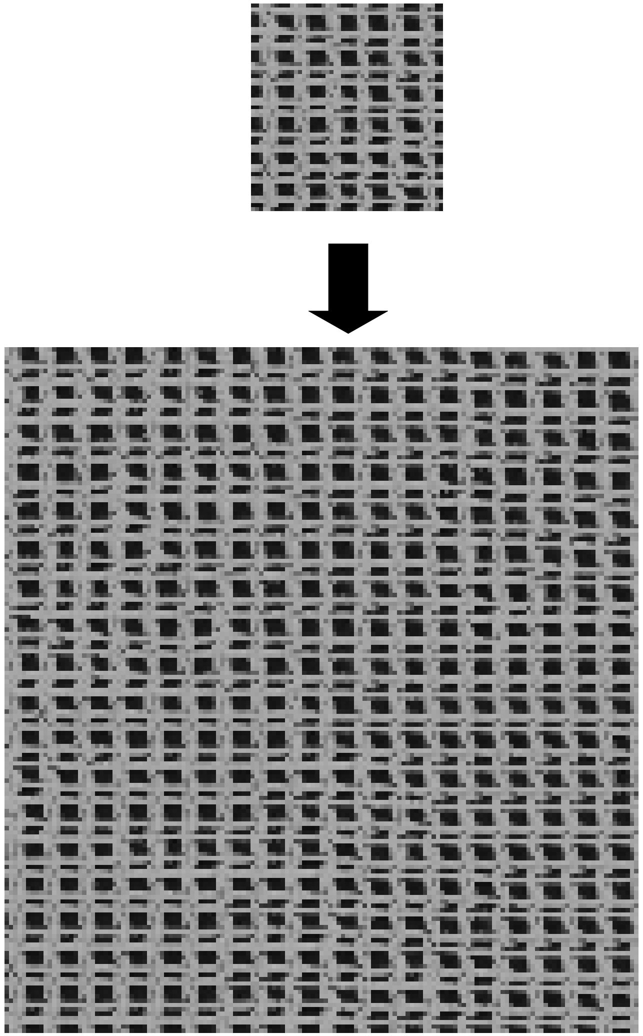
More results



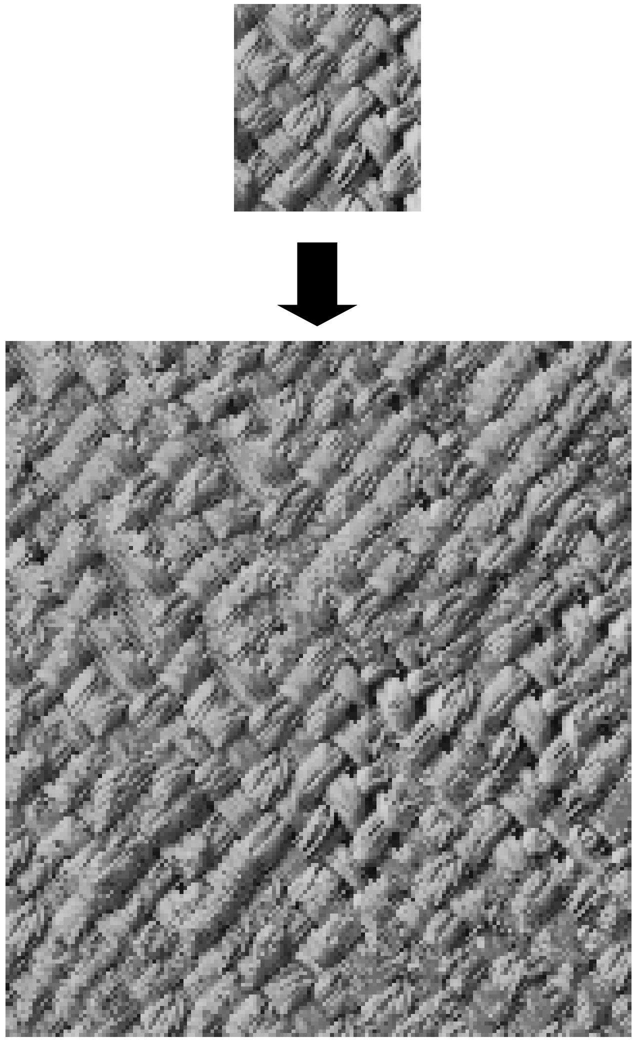
Increasing window size

More results

french canvas

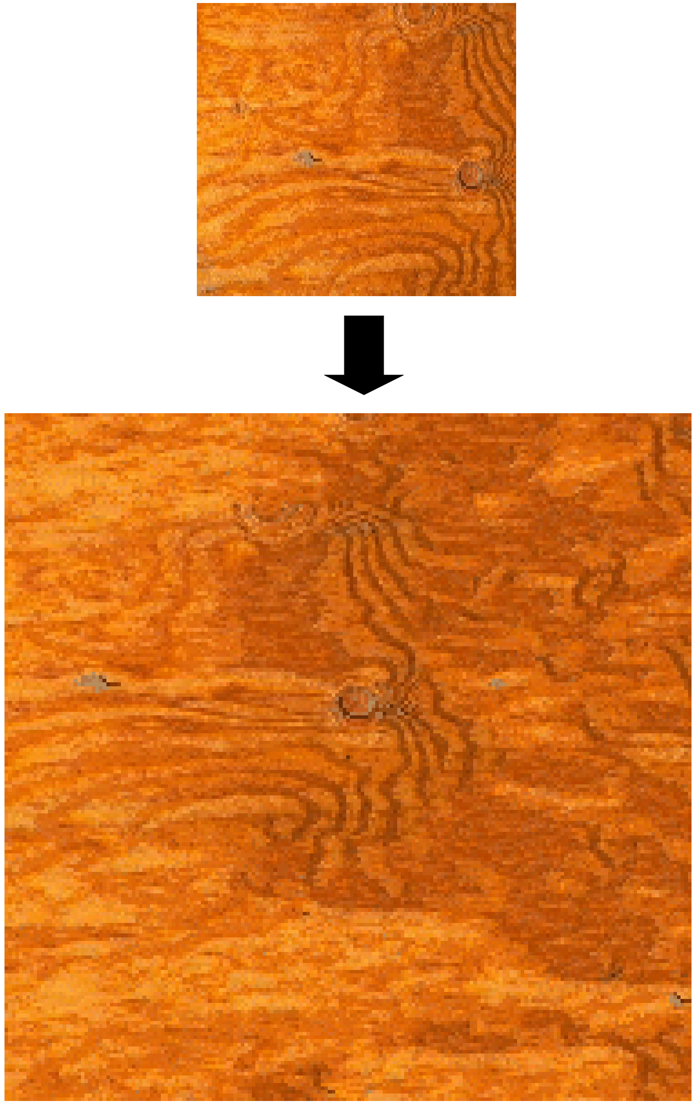


rafia weave

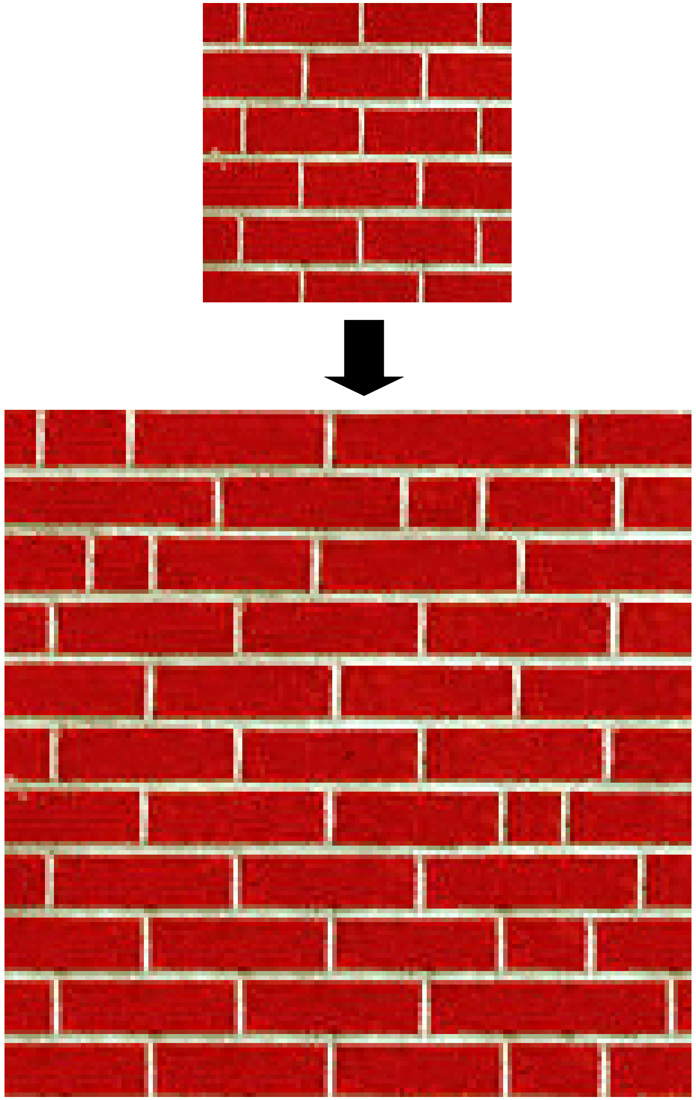


More results

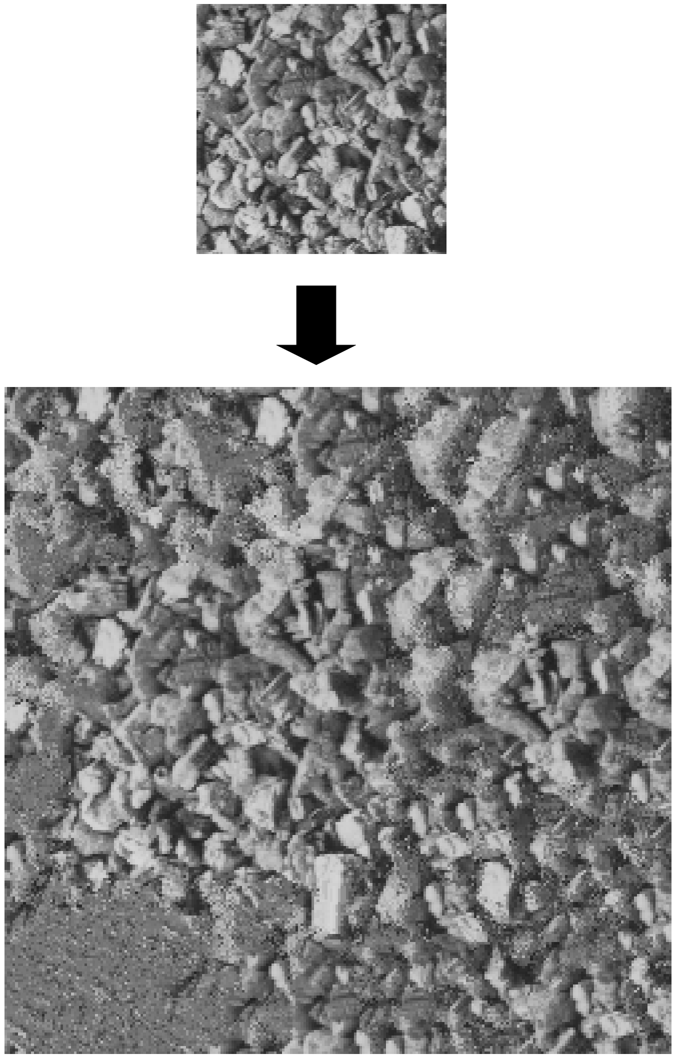
wood



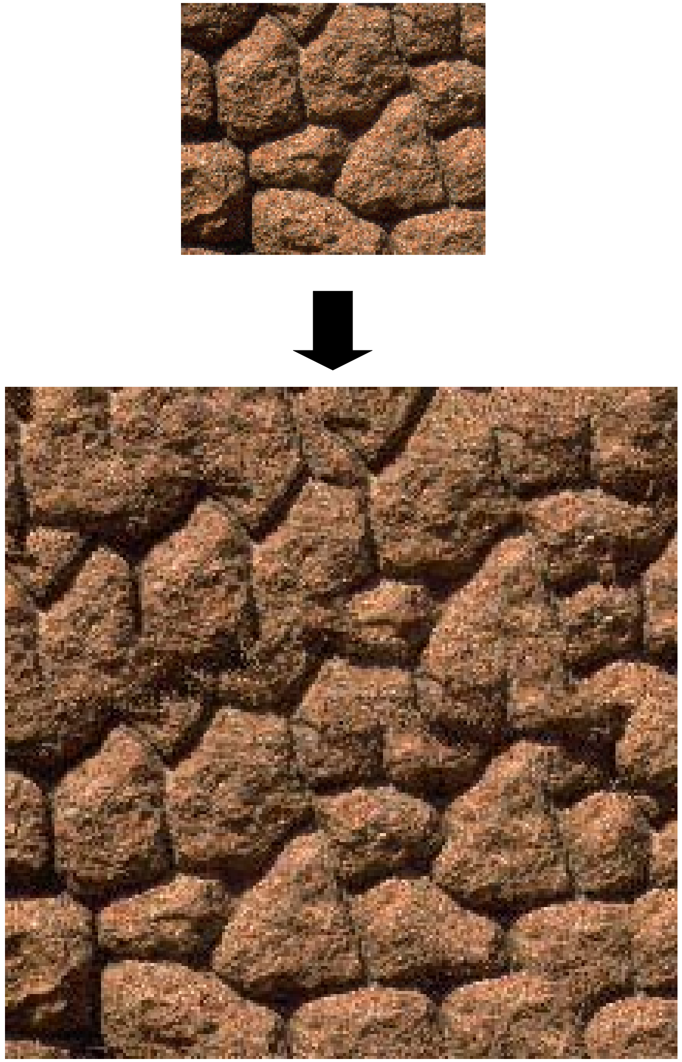
brick wall



Failure cases



Growing garbage

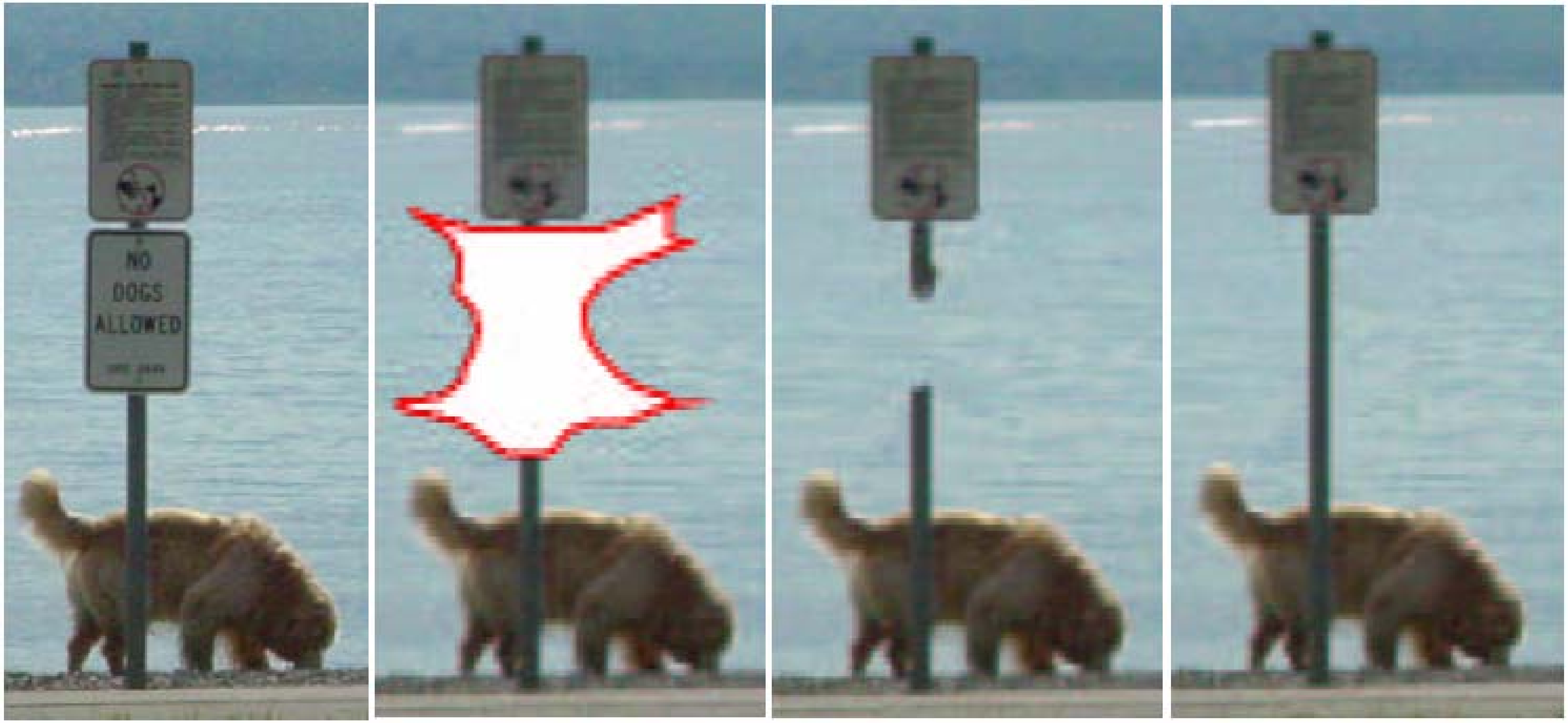


Verbatim copying

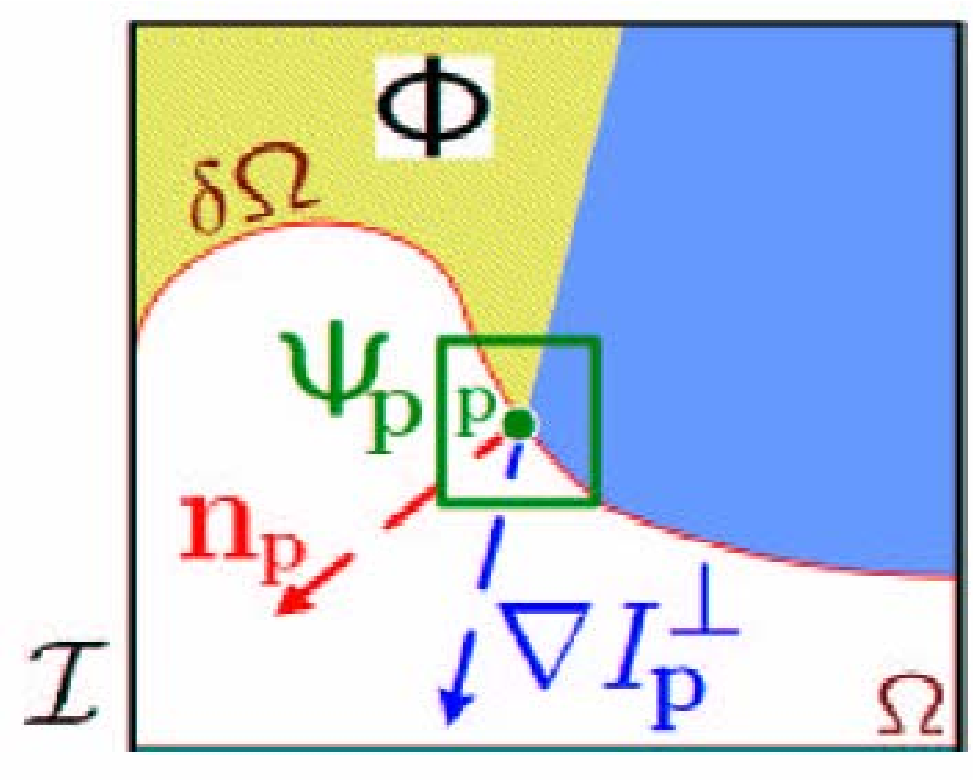
Inpainting

- Growing is in “onion peeling” order
  - within each “layer”, pixels with most neighbors are synthesized first

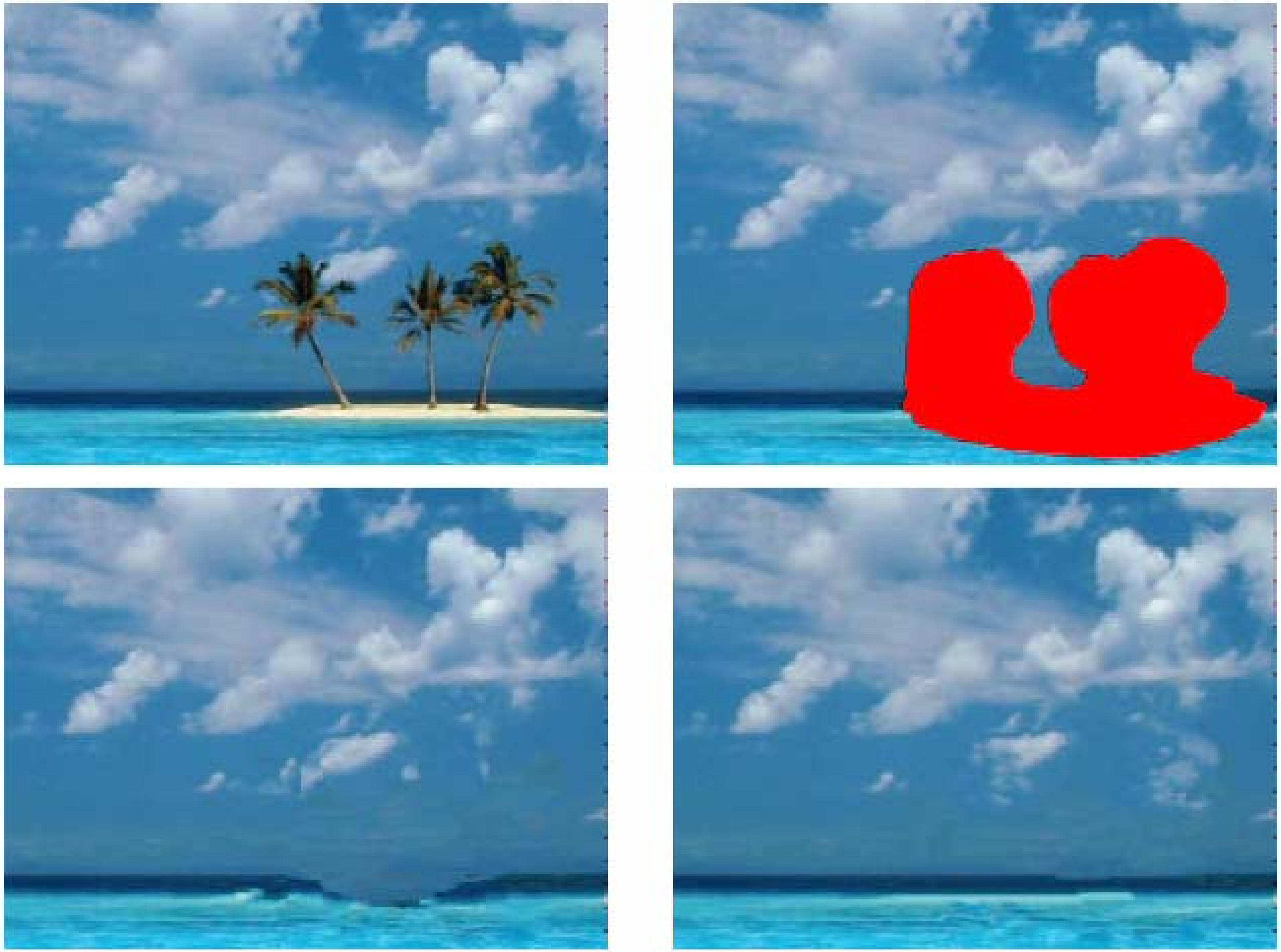
Inpainting



Inpainting

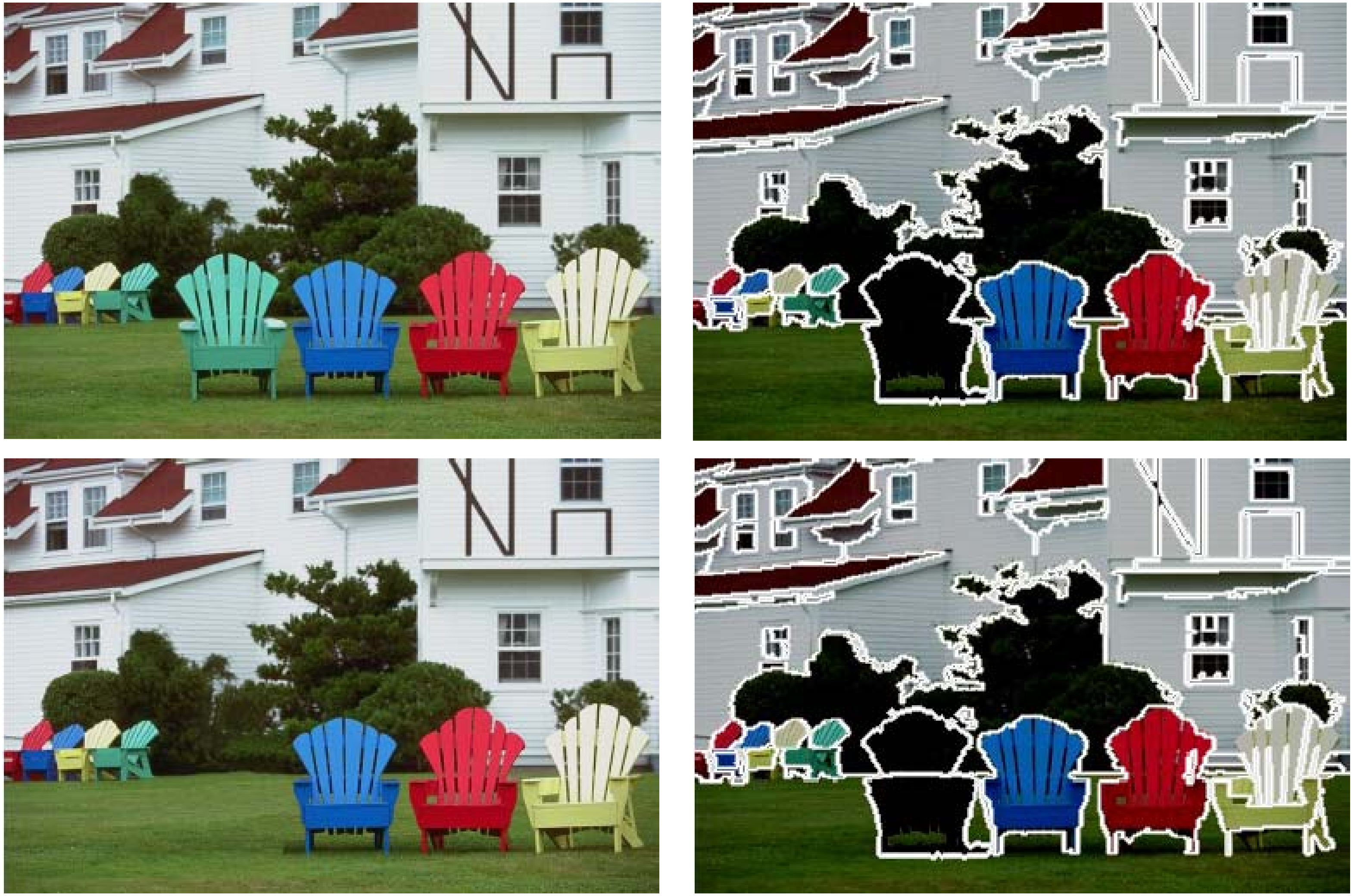


Results



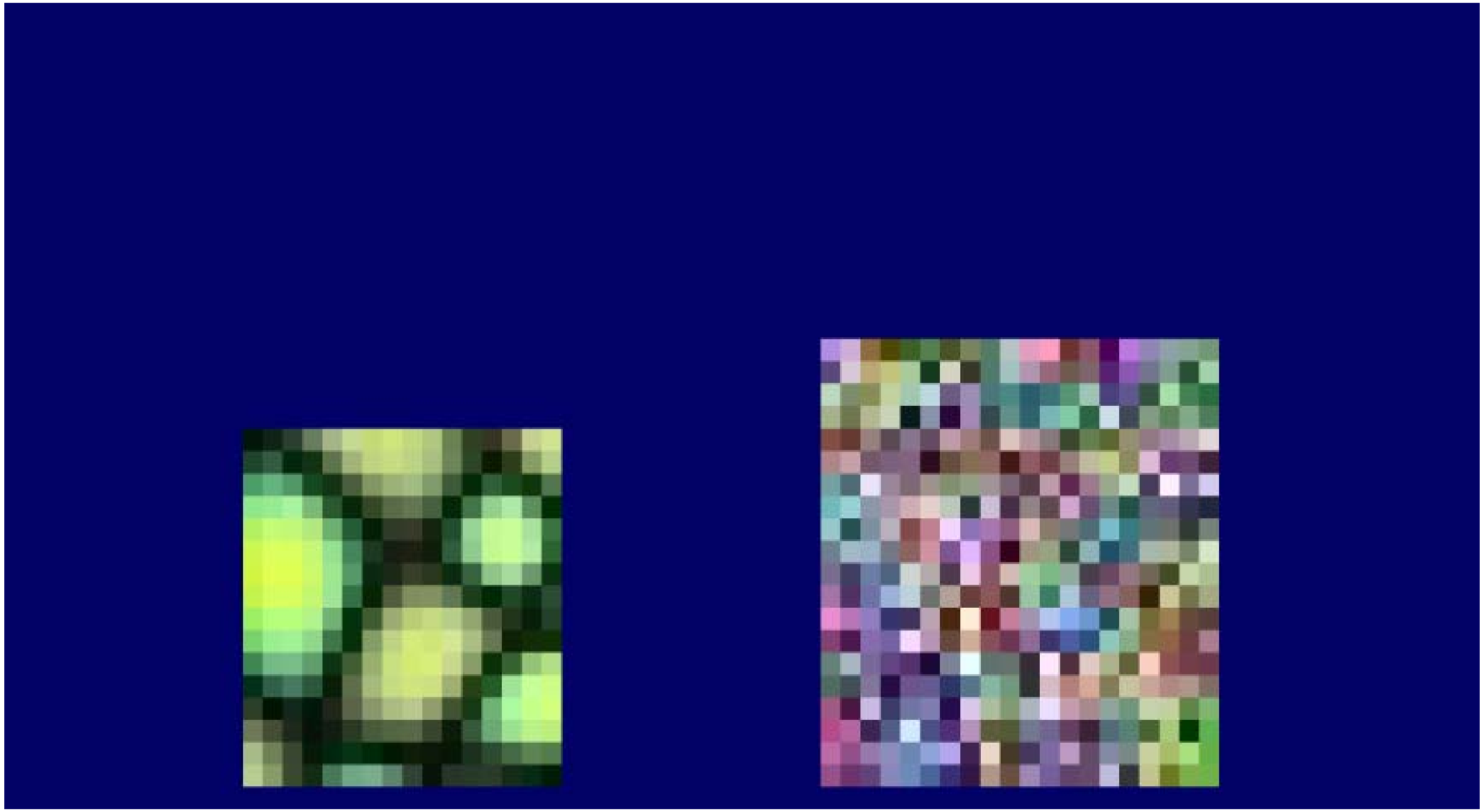
Recent inpaiting algorithms

Obtain structure first, add details by texture synthesis



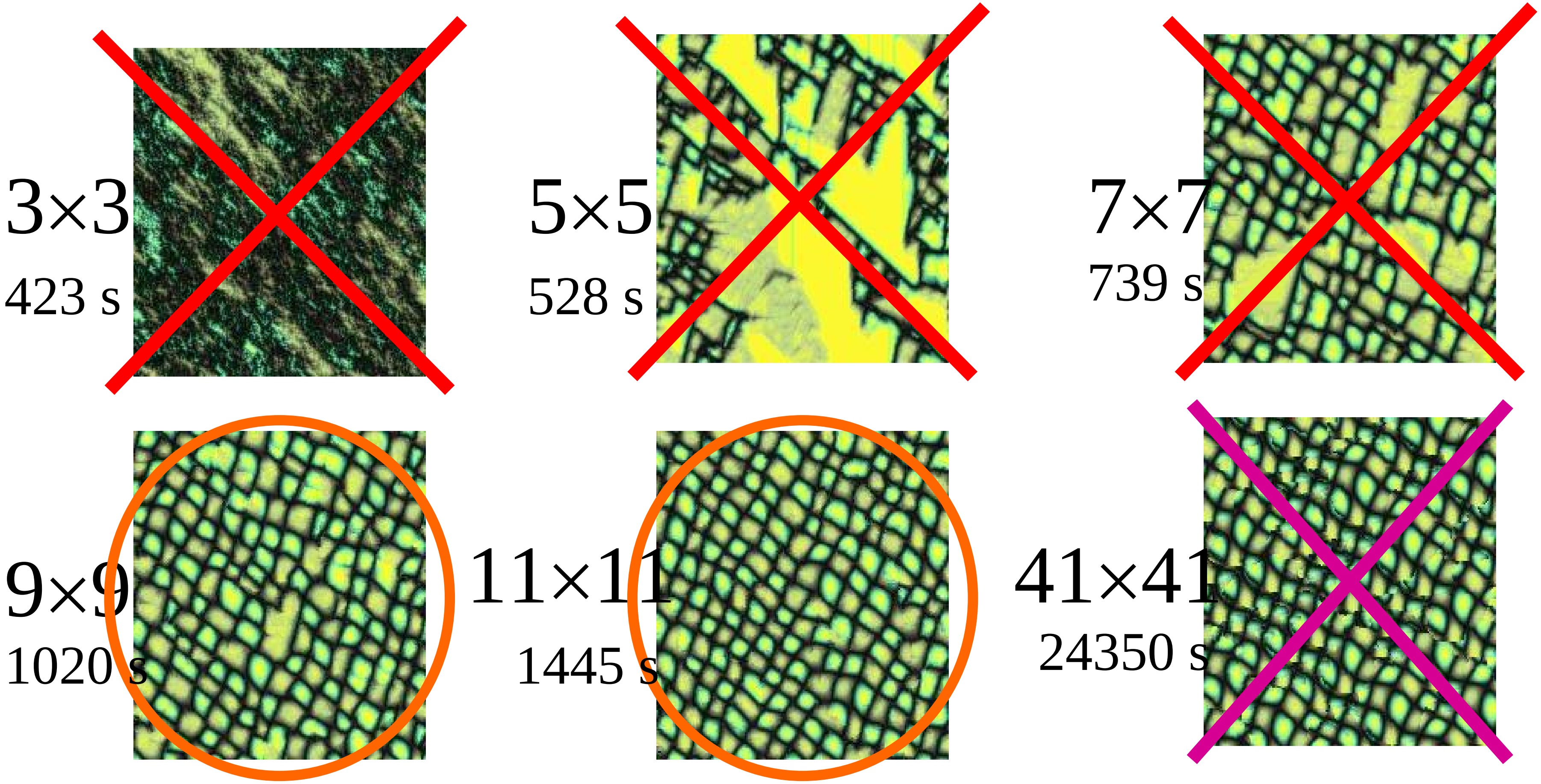
Summary of the basic algorithm

- Exhaustively search neighborhoods



Neighborhood

- Neighborhood size determines the quality & cost



Summary

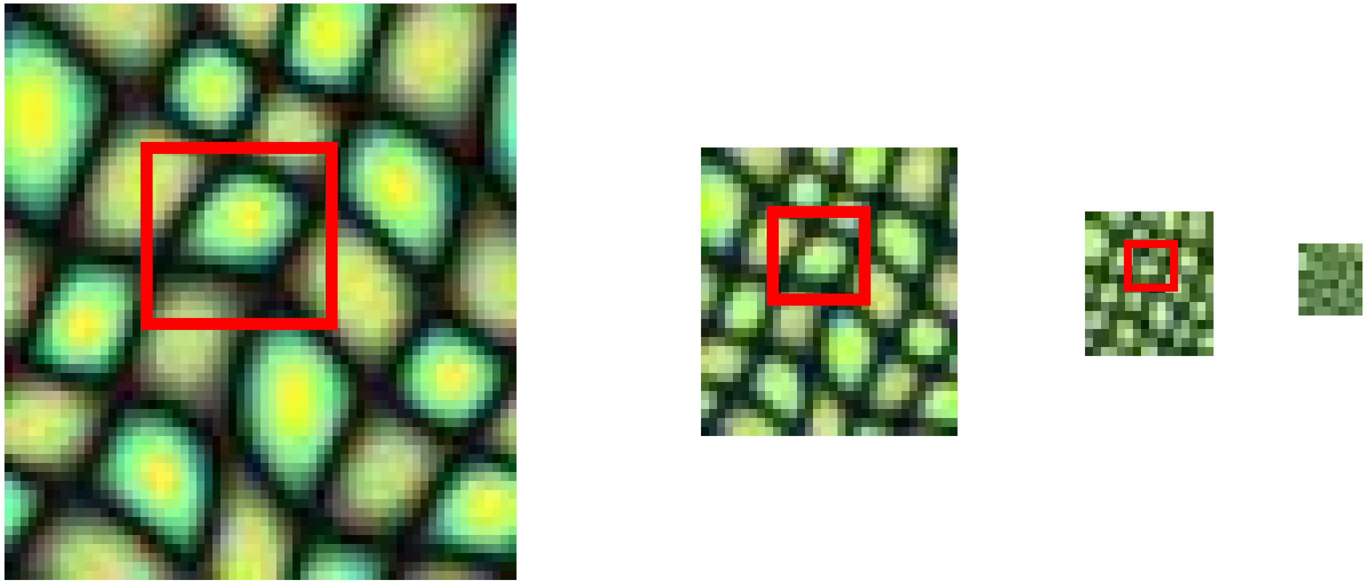
- Advantages:
  - conceptually simple
  - models a wide range of real-world textures
  - naturally does hole-filling
- Disadvantages:
  - it's slow
  - it's a heuristic

Acceleration by Wei & Levoy

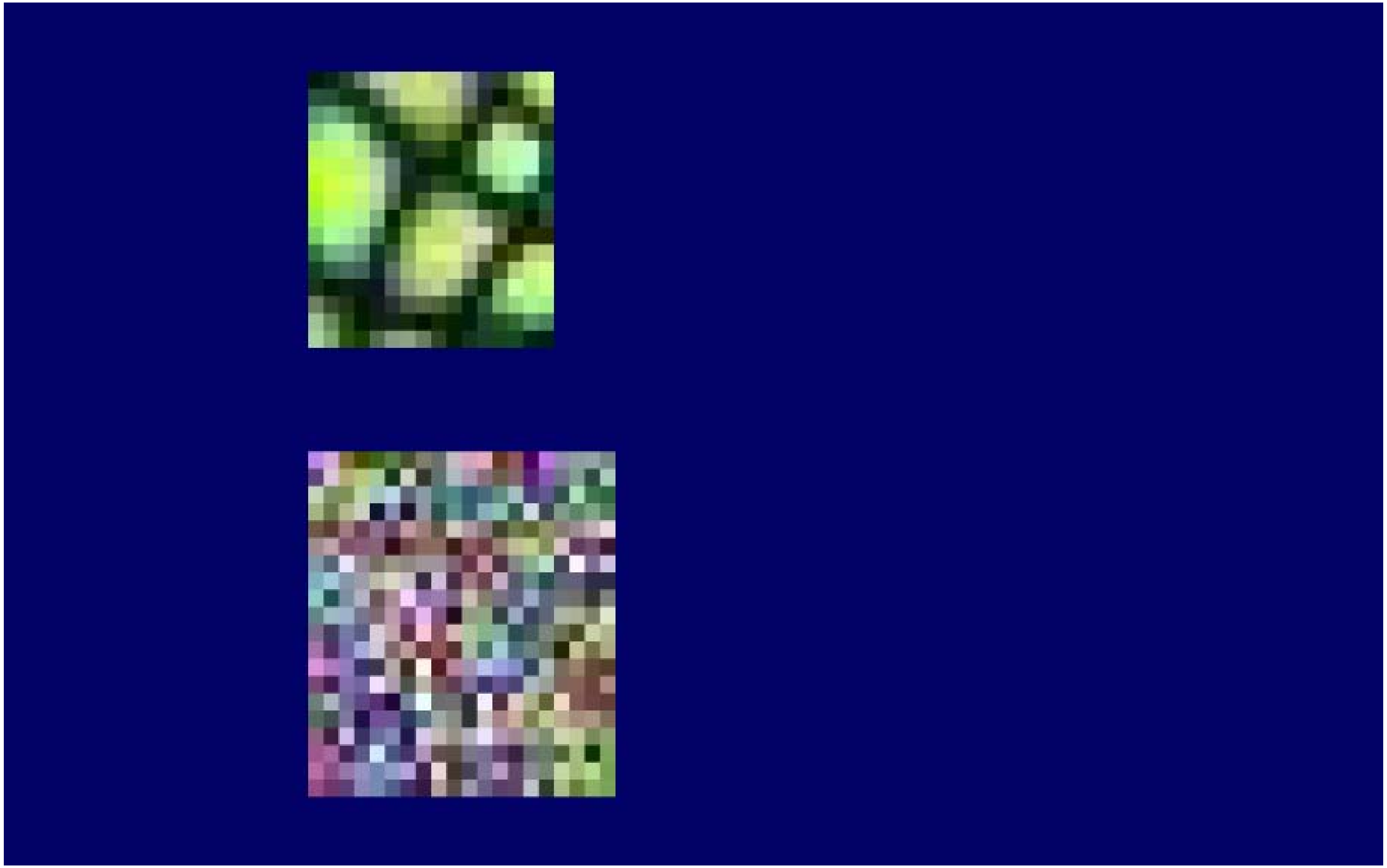
- Multi-resolution
- Tree-structure

Multi-resolution pyramid

High resolution ⟷ Low resolution

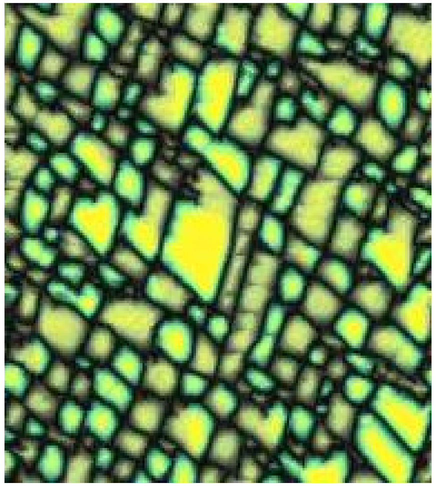


Multi-resolution algorithm

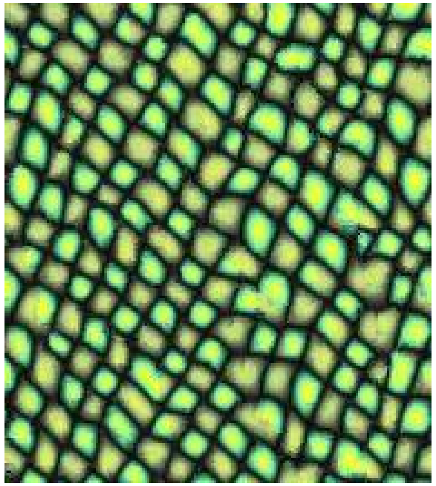


Benefits

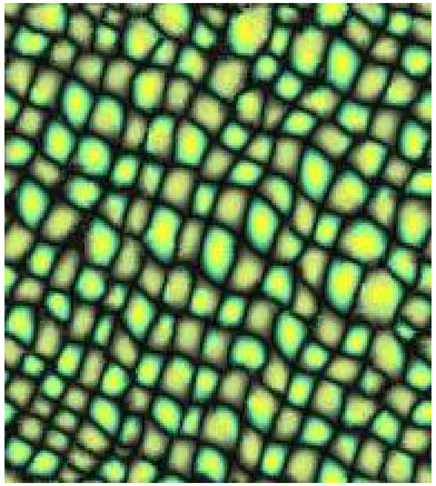
- Better image quality & faster computation



1 level
5×5

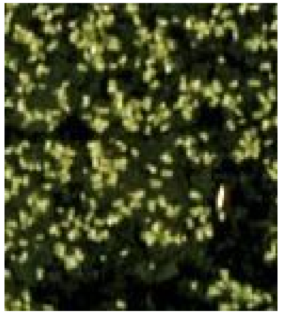


1 level
11×11

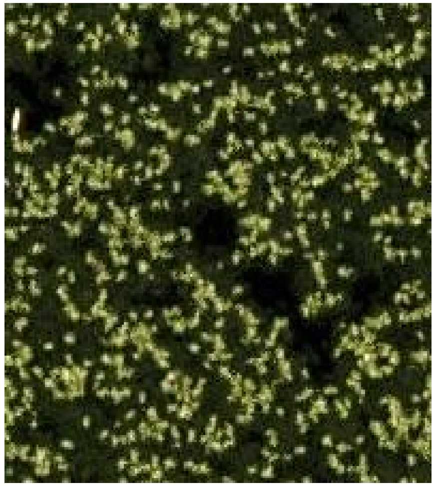


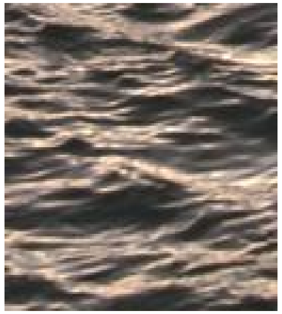
3 levels
5×5

Results

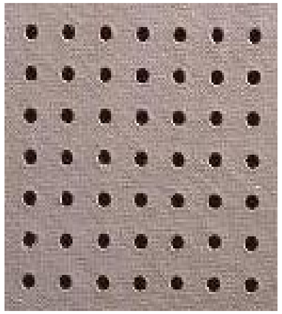


Random

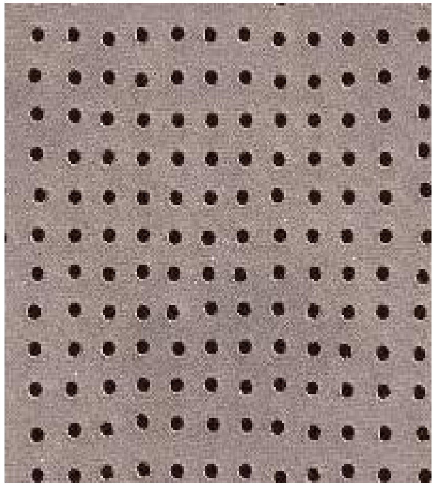




Oriented



Regular

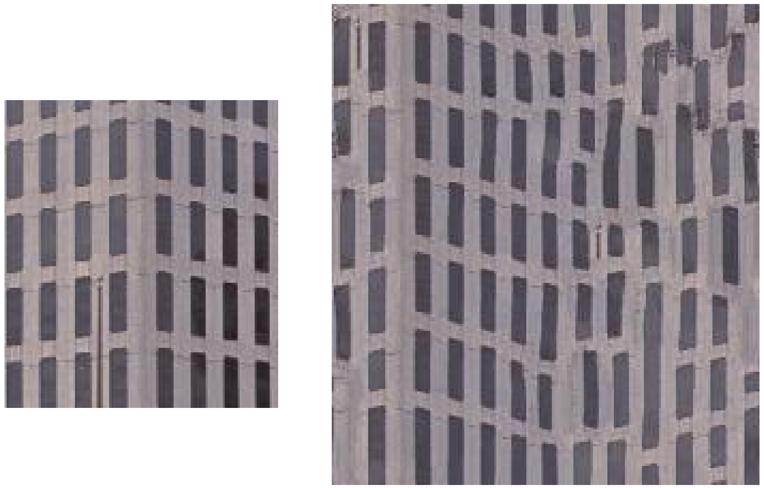


Semi-regular

Failures

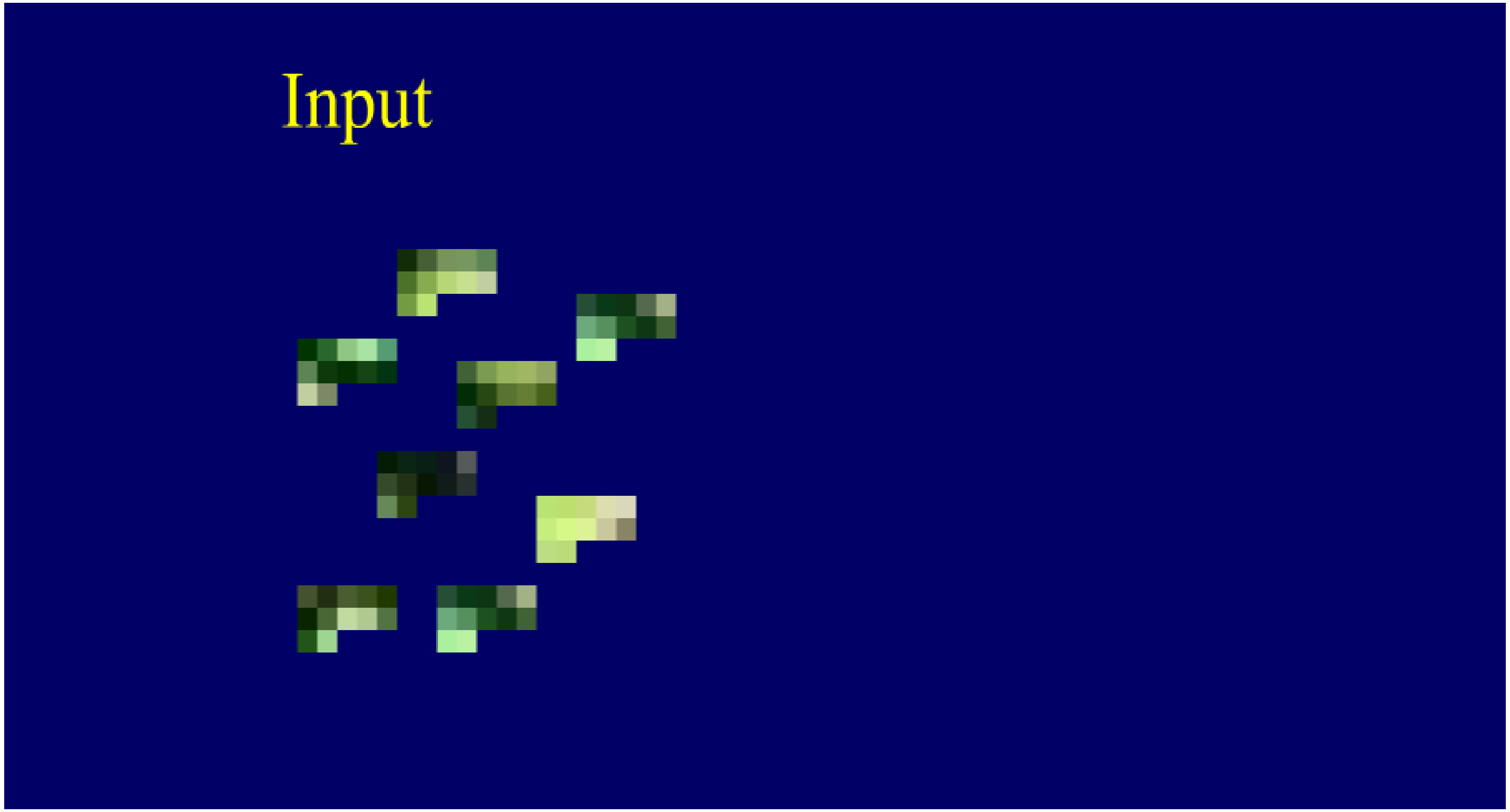
- Non-planar structures

- Global information



Acceleration

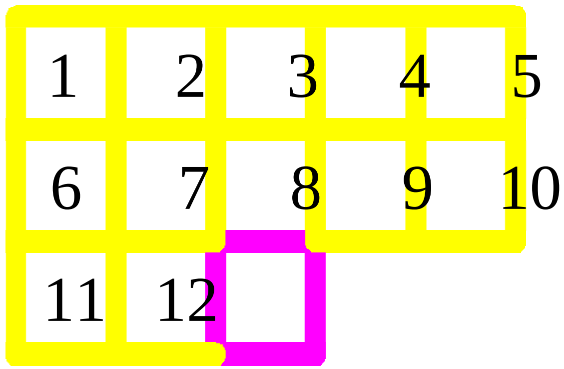
- Computation bottleneck: neighborhood search



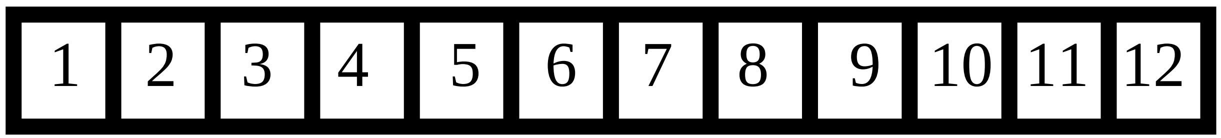
Nearest point search

- Treat neighborhoods as high dimensional points

Neighborhood



High dimensional point/vector

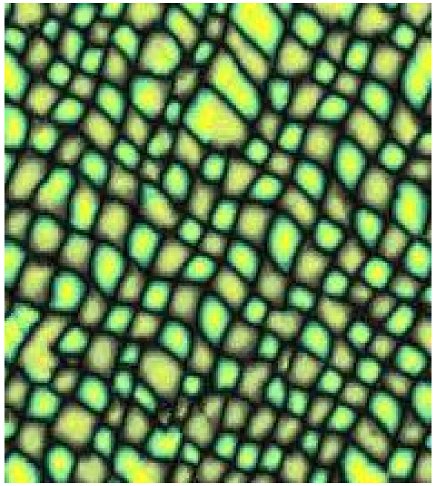


Tree-Structured Vector Quantization

Timing

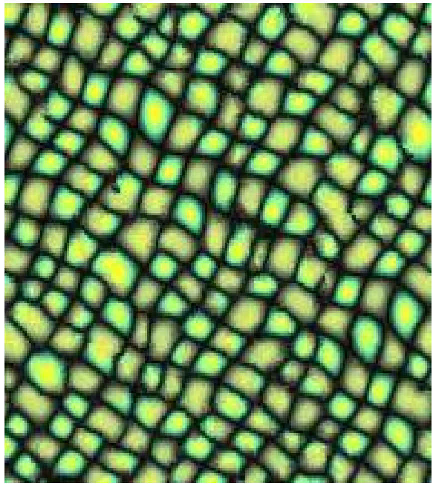
- Time complexity : O(log N) instead of O(N)

Efros 99



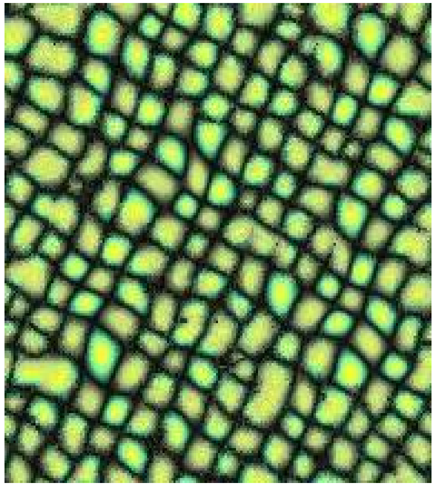
1941 secs

Full searching



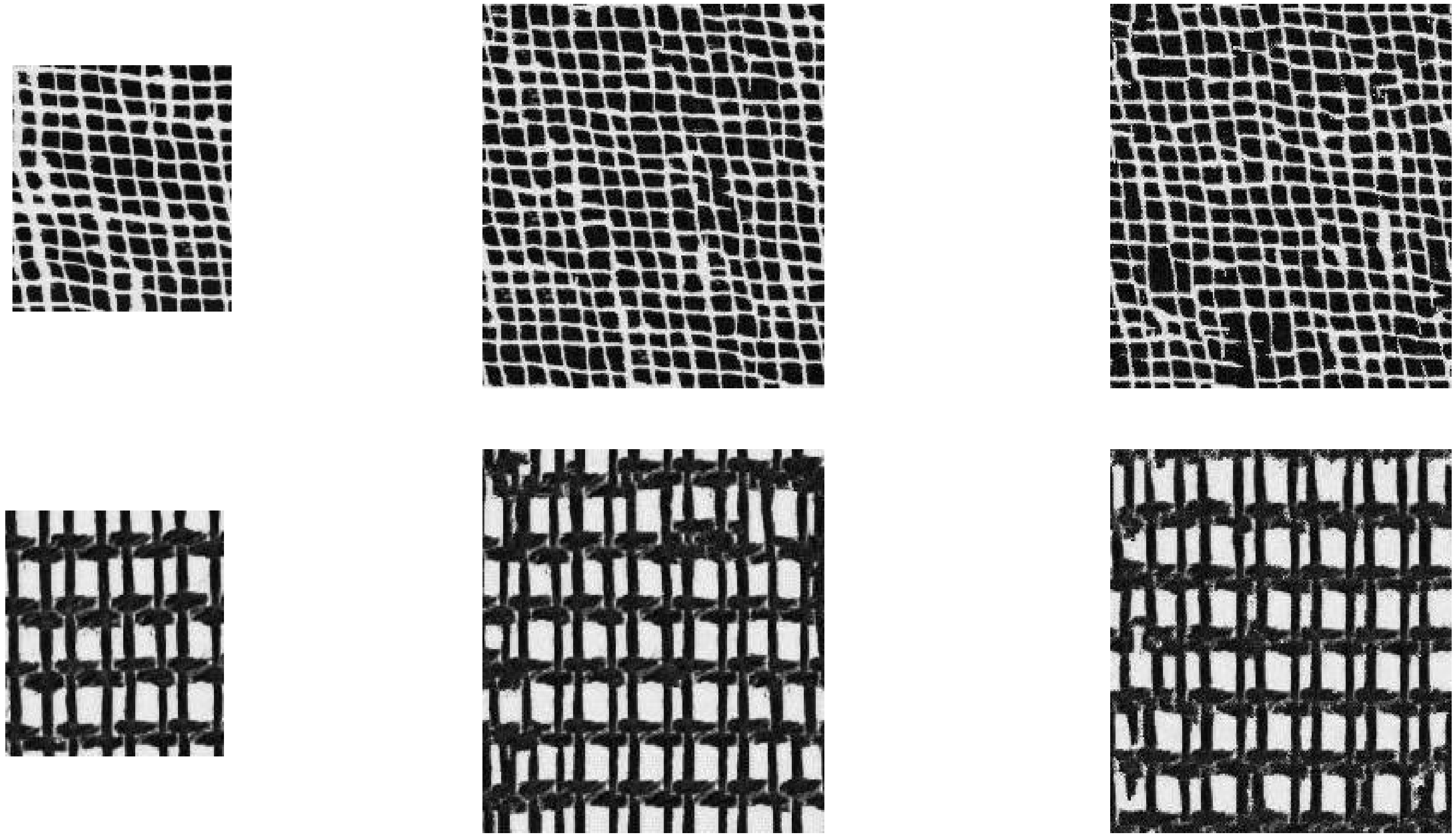
503 secs

TSVQ



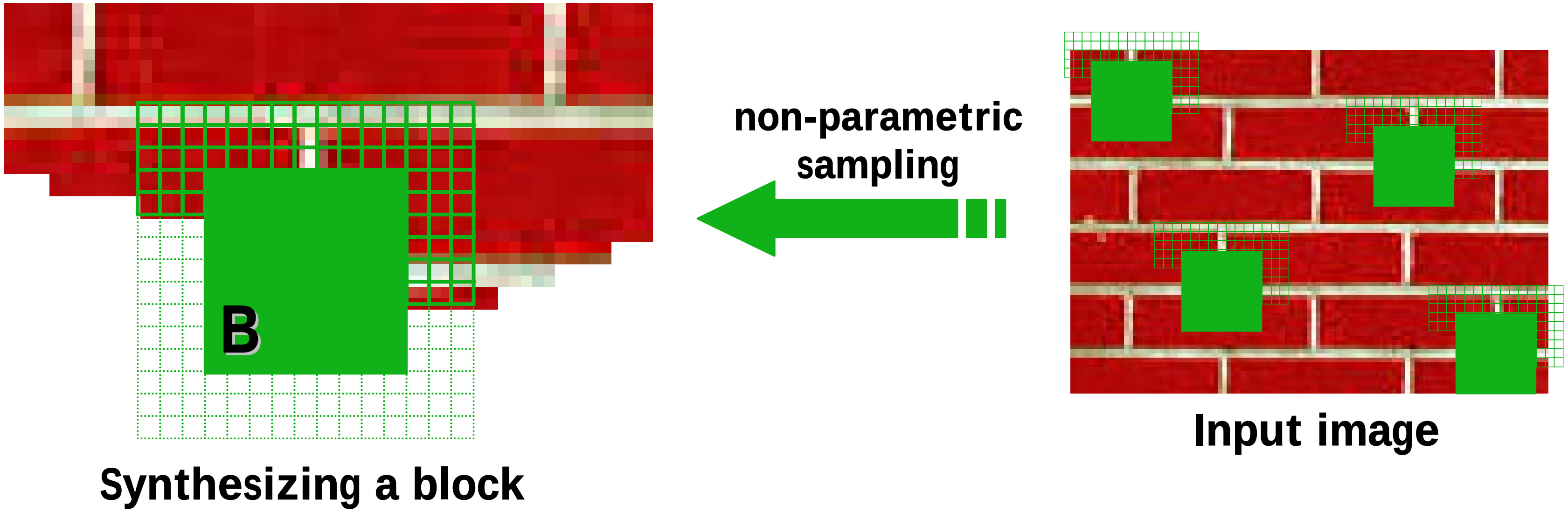
12 secs

Results



Input   Exhaustive: 360 s   TSVQ: 7.5 s

Patch-based methods



- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

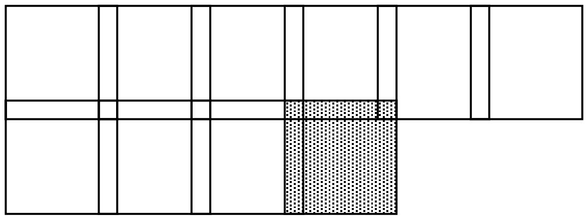
- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once

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

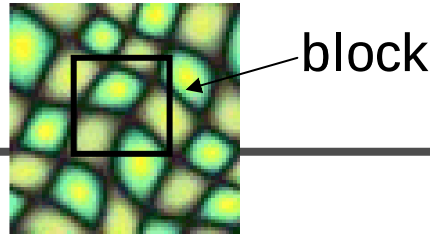
Algorithm

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

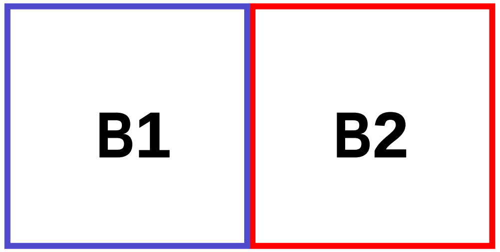


- Search input texture for block that satisfies overlap constraints (above and left)
- Paste new block into resulting texture
  - blending
  - use dynamic programming to compute minimal error boundary cut

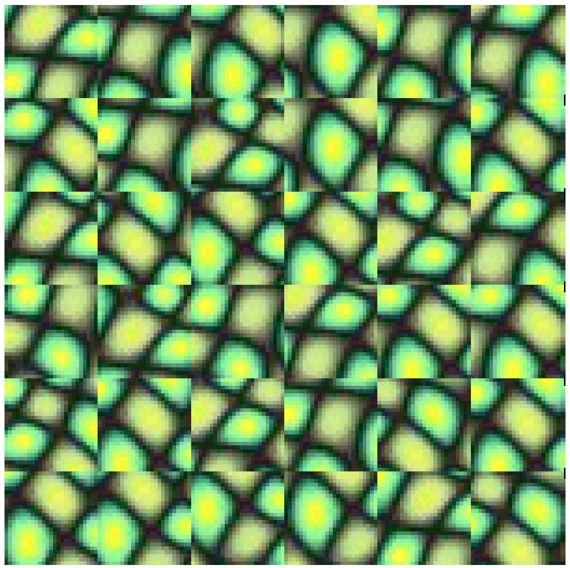
DigiVFX

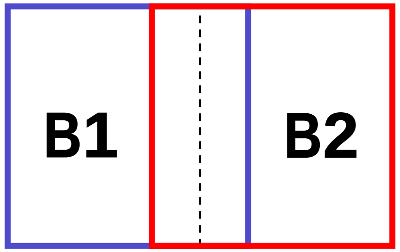


Input texture

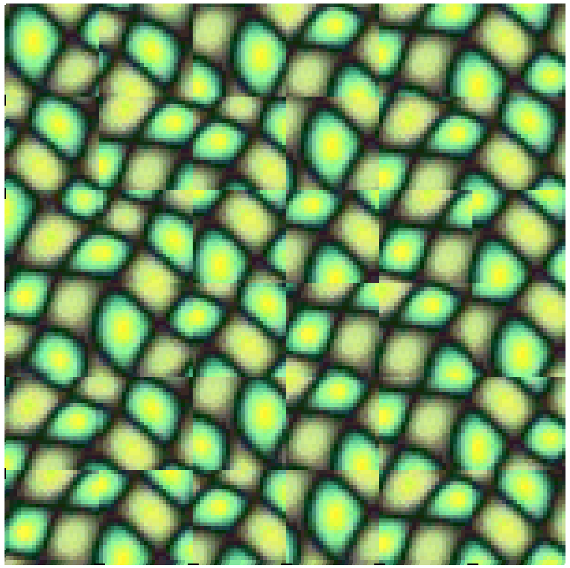


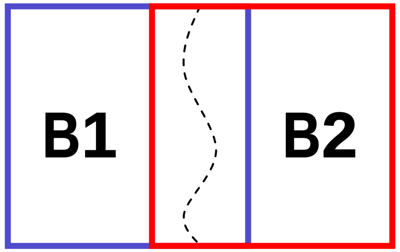
Random placement of blocks



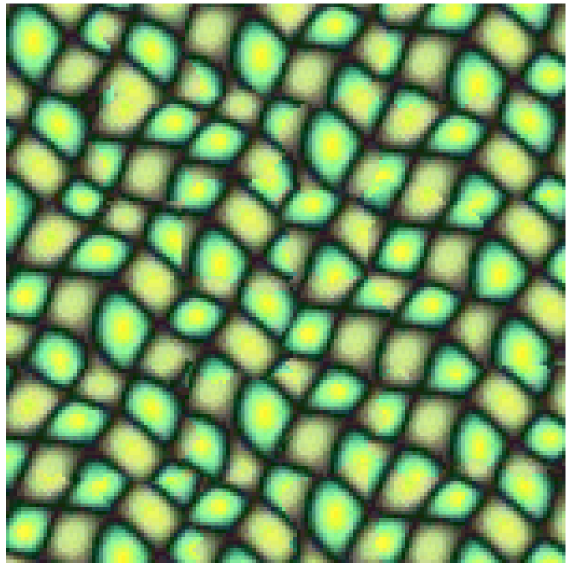


Neighboring blocks constrained by overlap

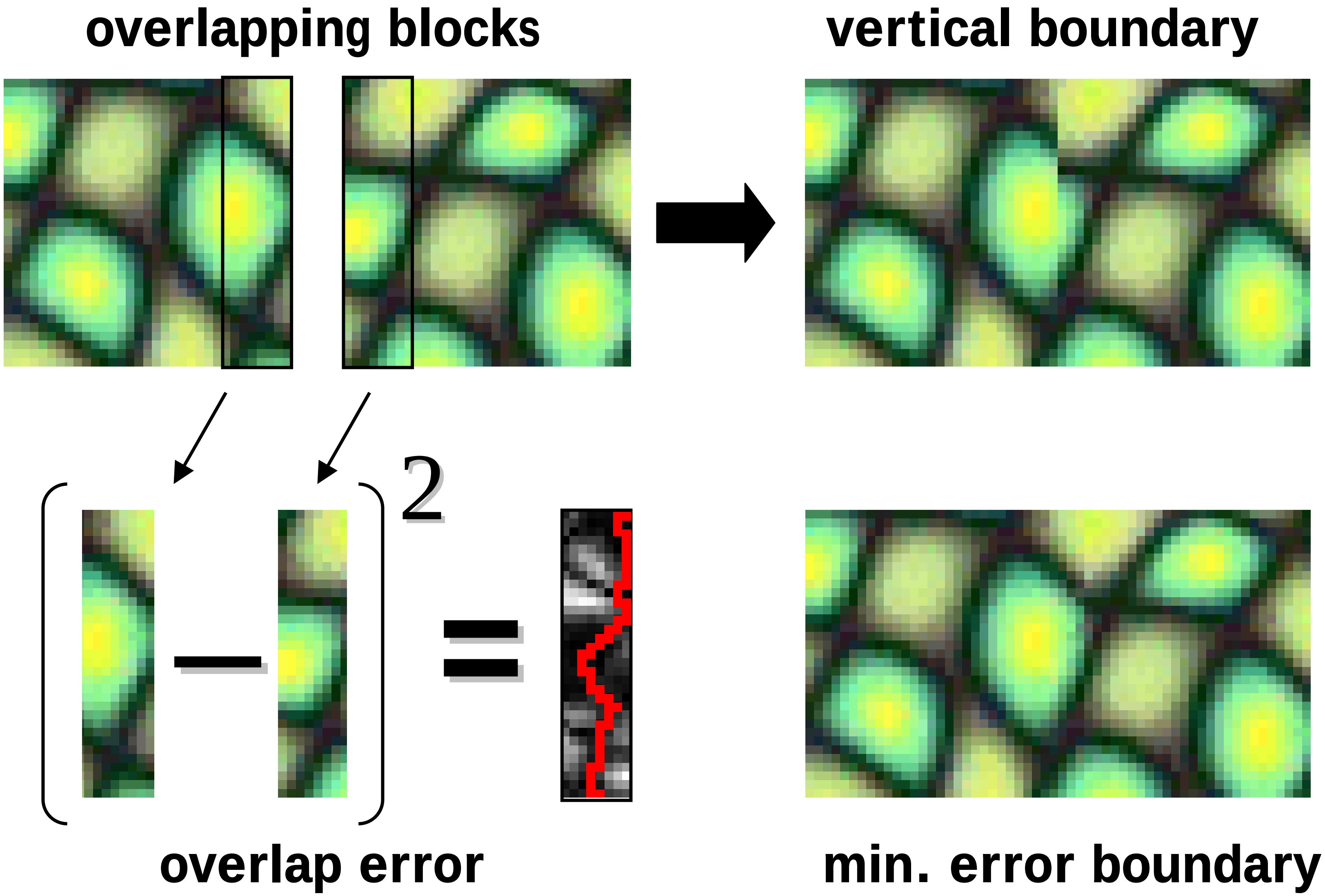




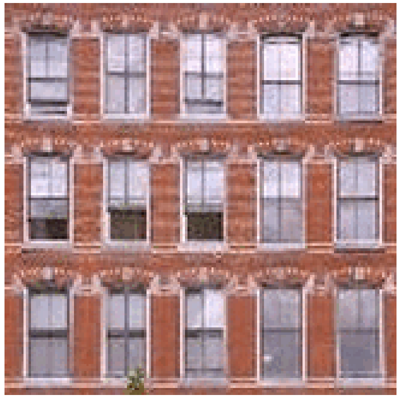
Minimal error boundary cut



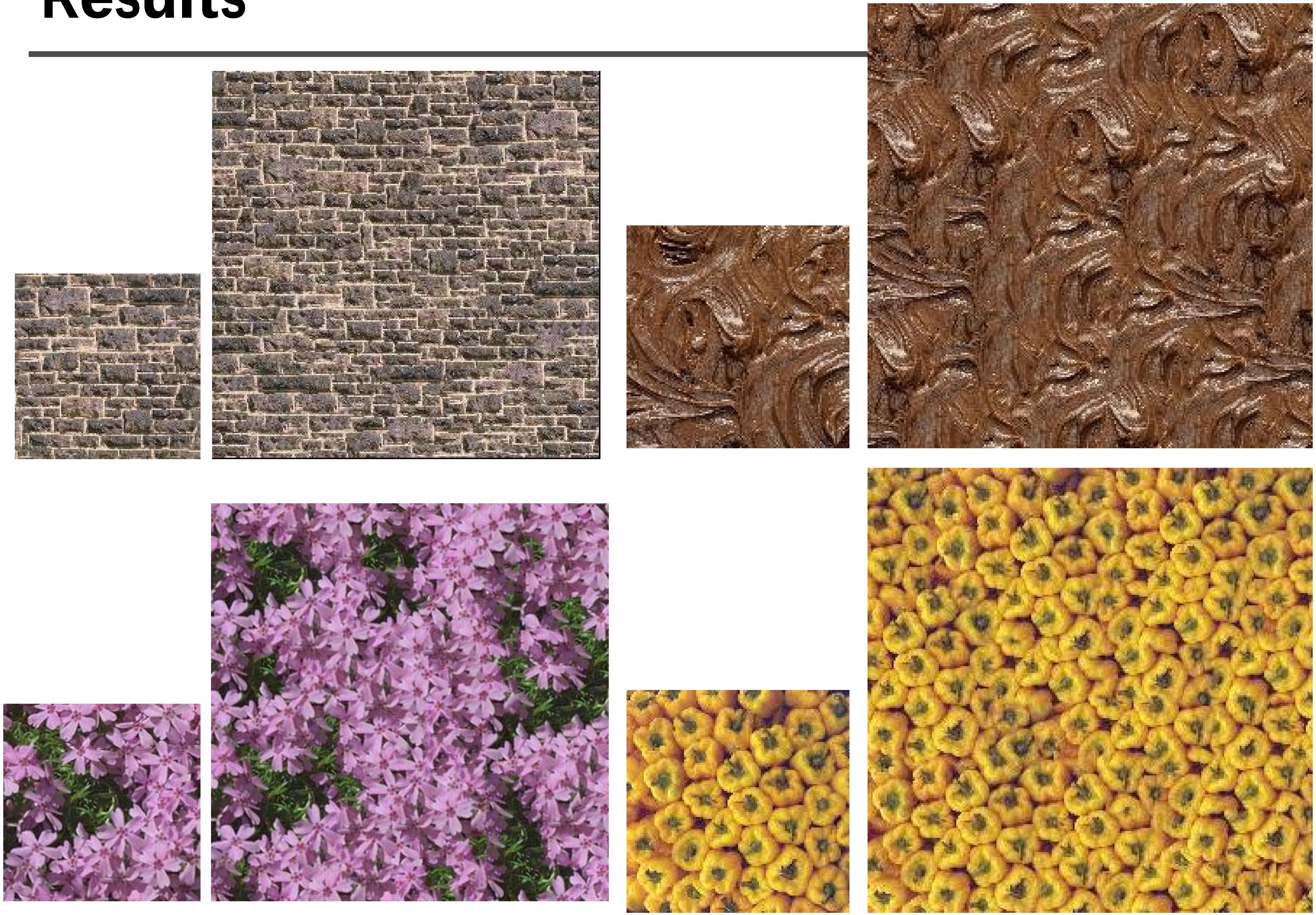
Minimal error boundary



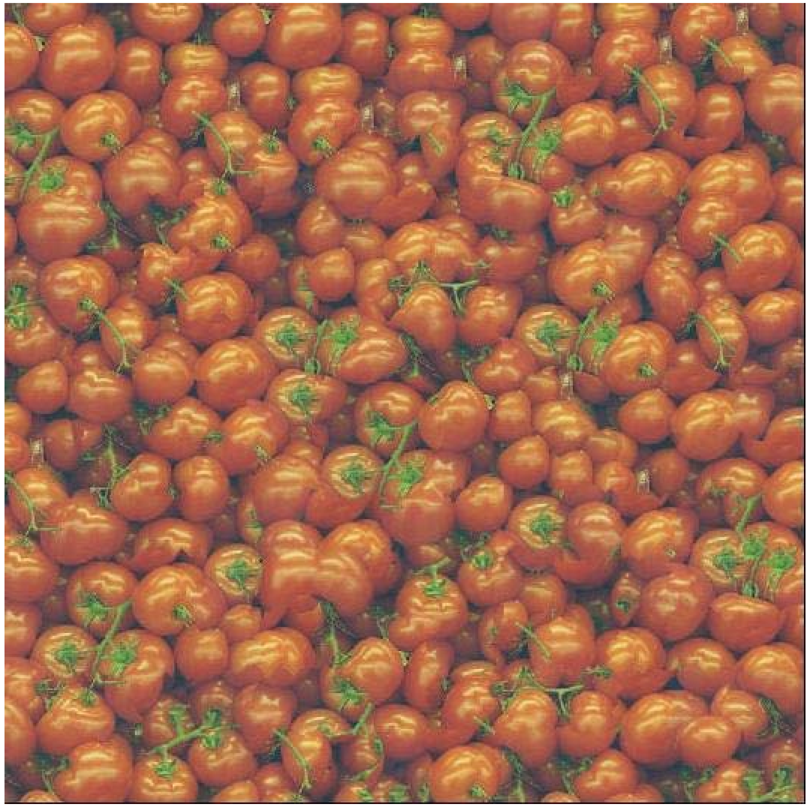
Results

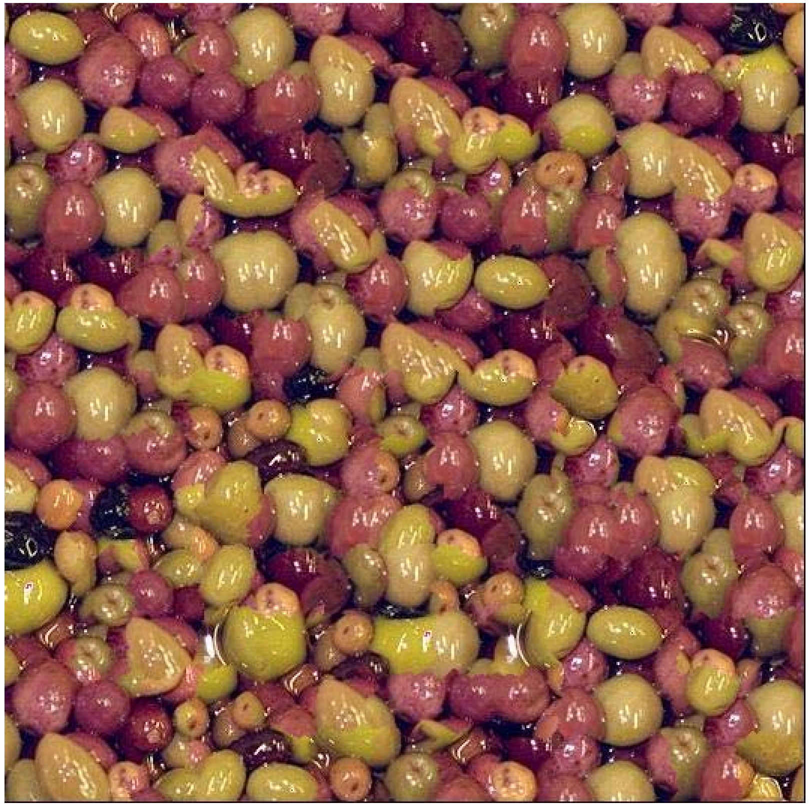


Results

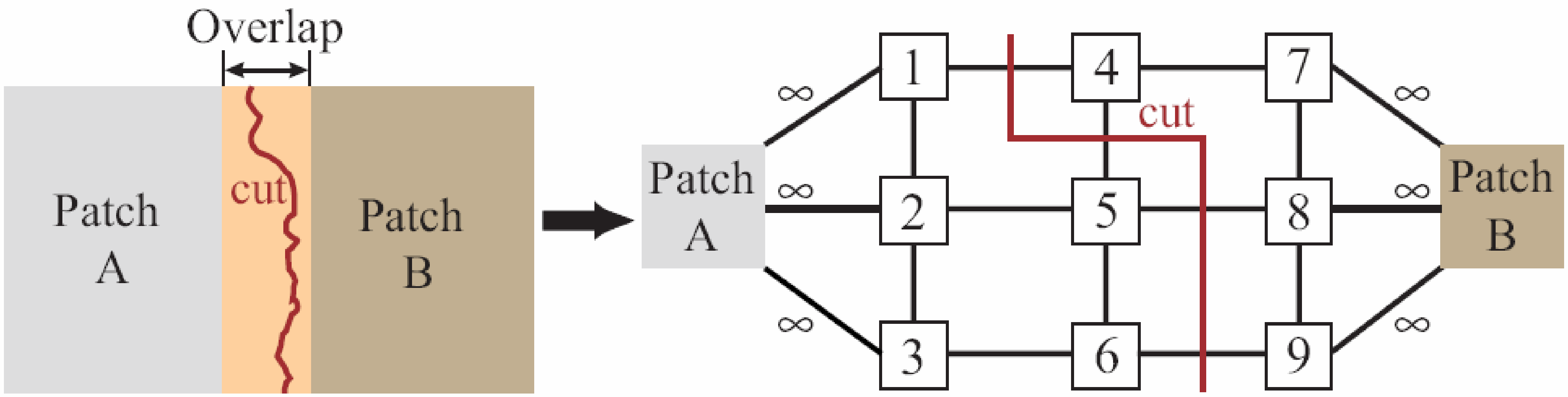


Failure cases





GraphCut textures

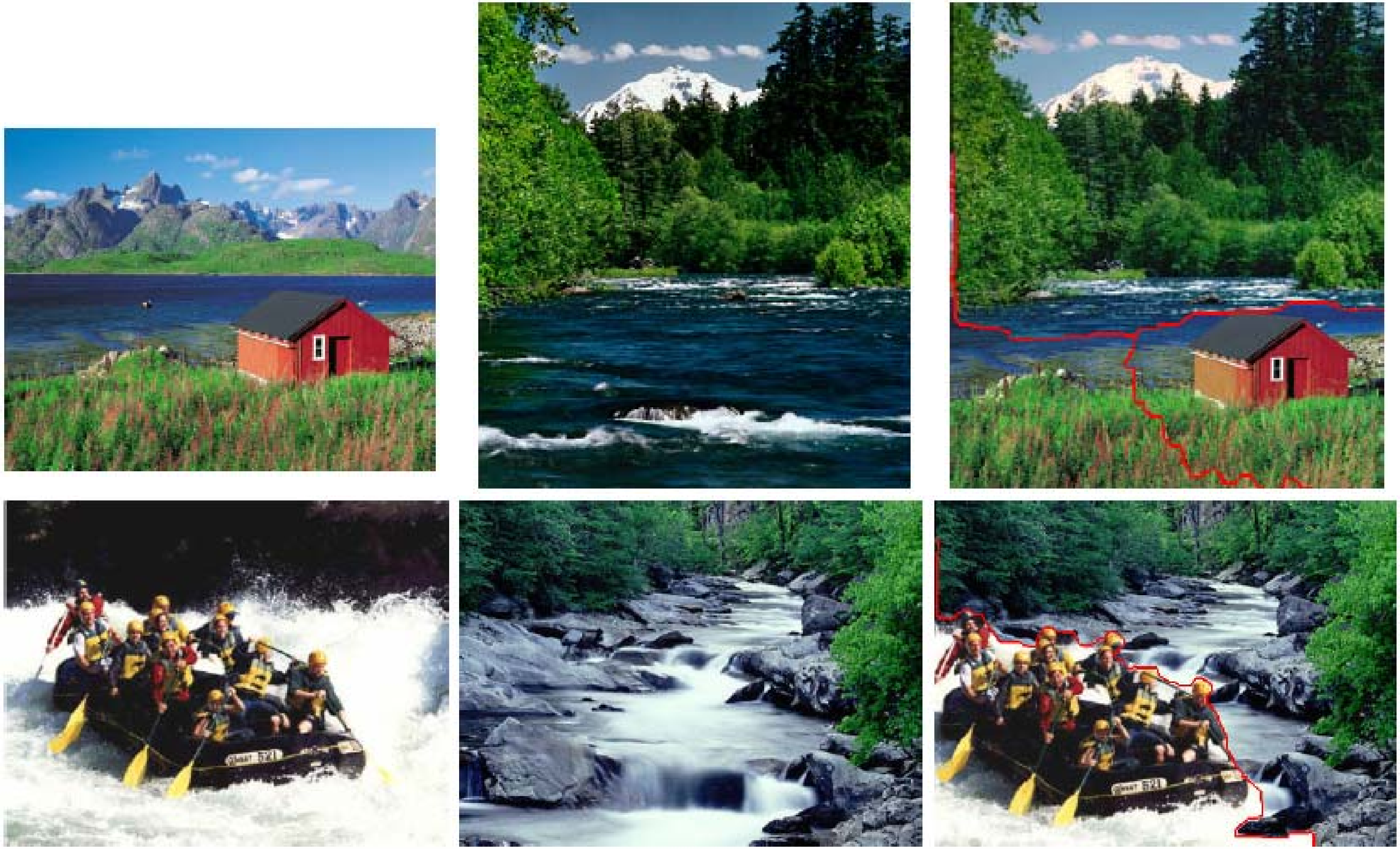


Input

Image Quilting

Graph cut

GraphCut textures



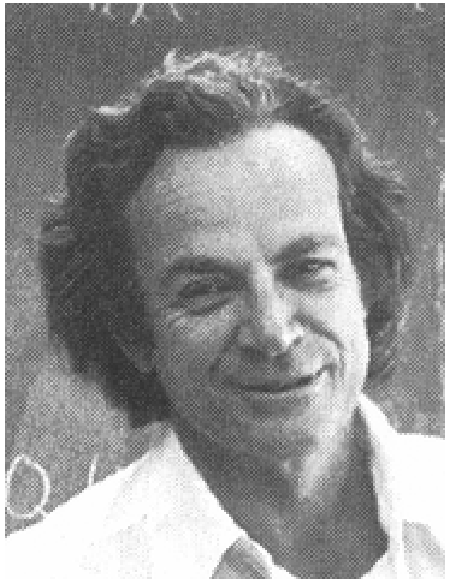
Photomontage

Photomontage

Photomontage

Texture transfer

Source texture



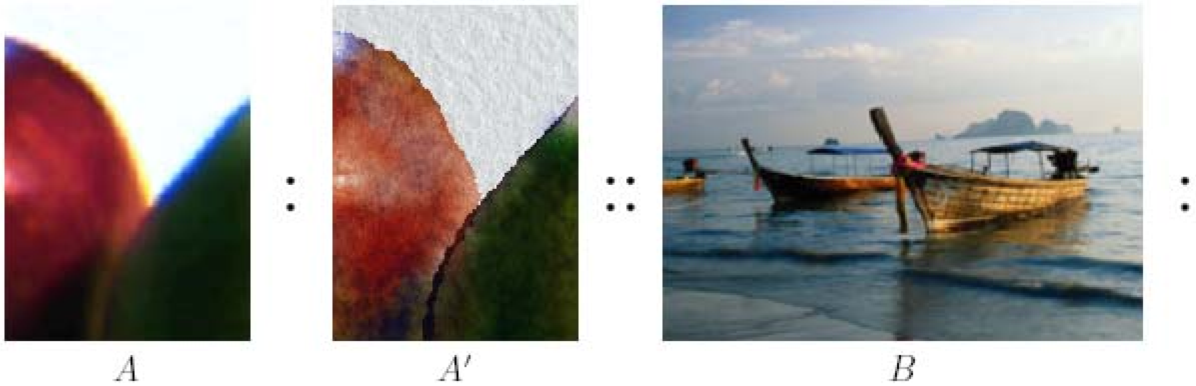
Target image

Source correspondence image

Target correspondence image

Texture transfer

Image Analogies



Coherence search

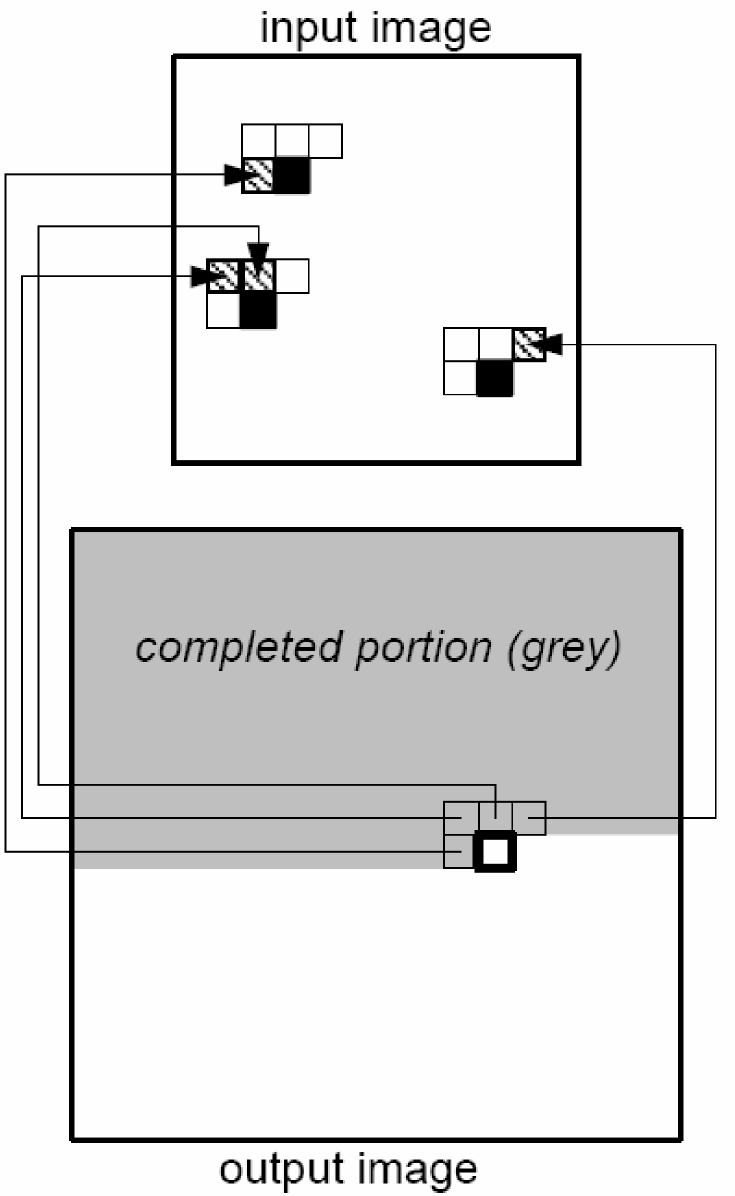


Image Analogies Implementation

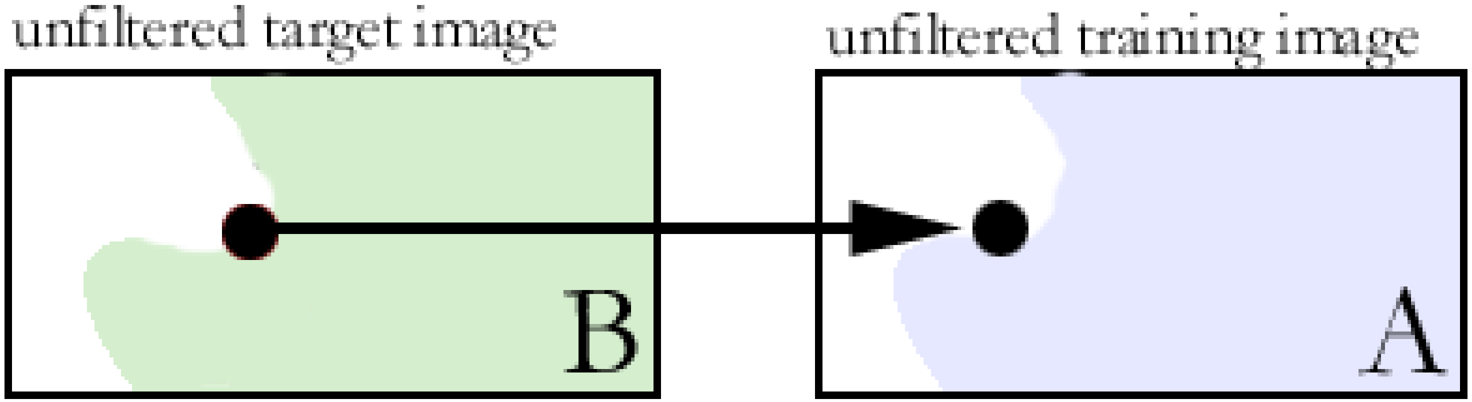


Image Analogies Implementation

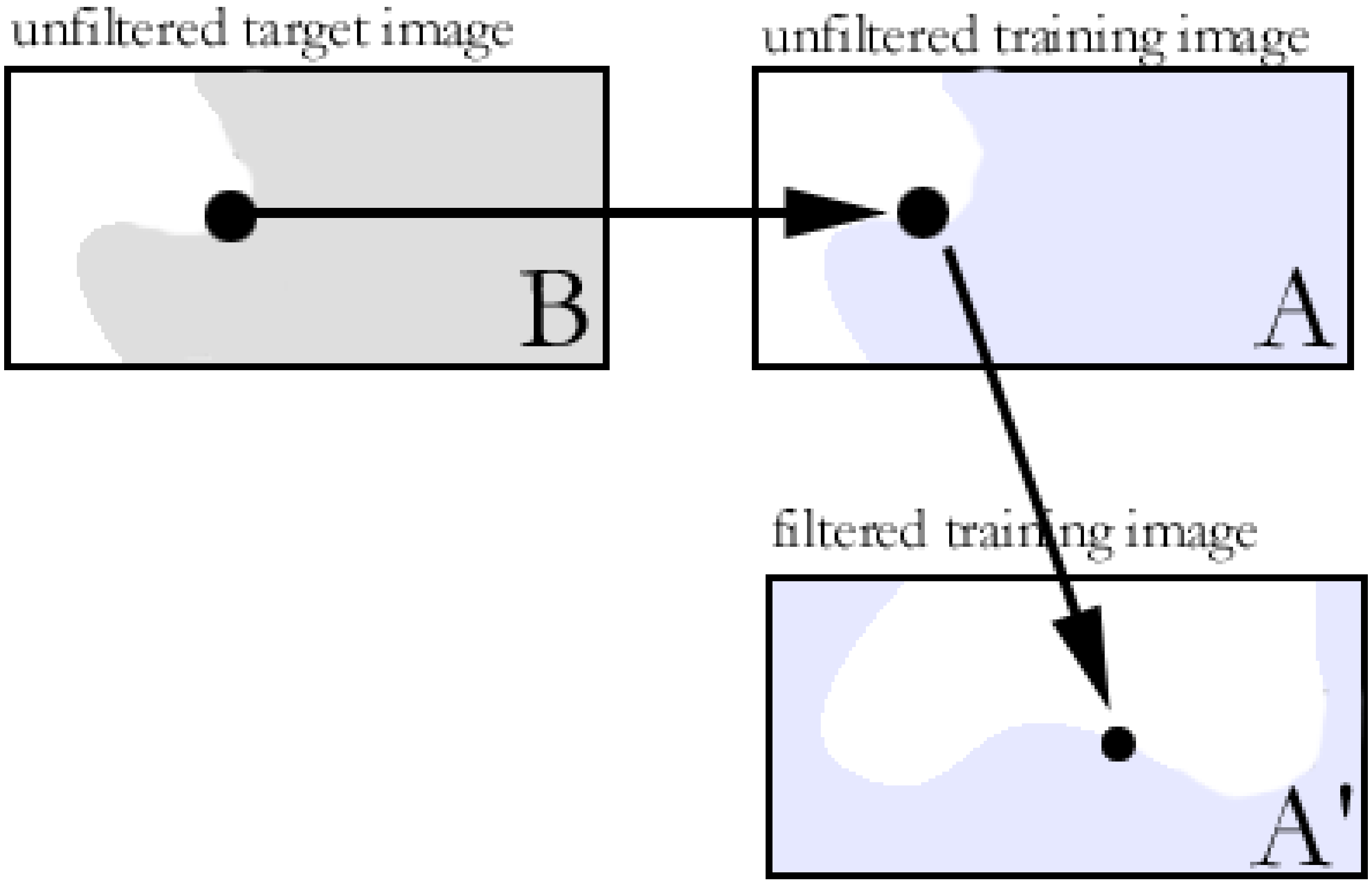
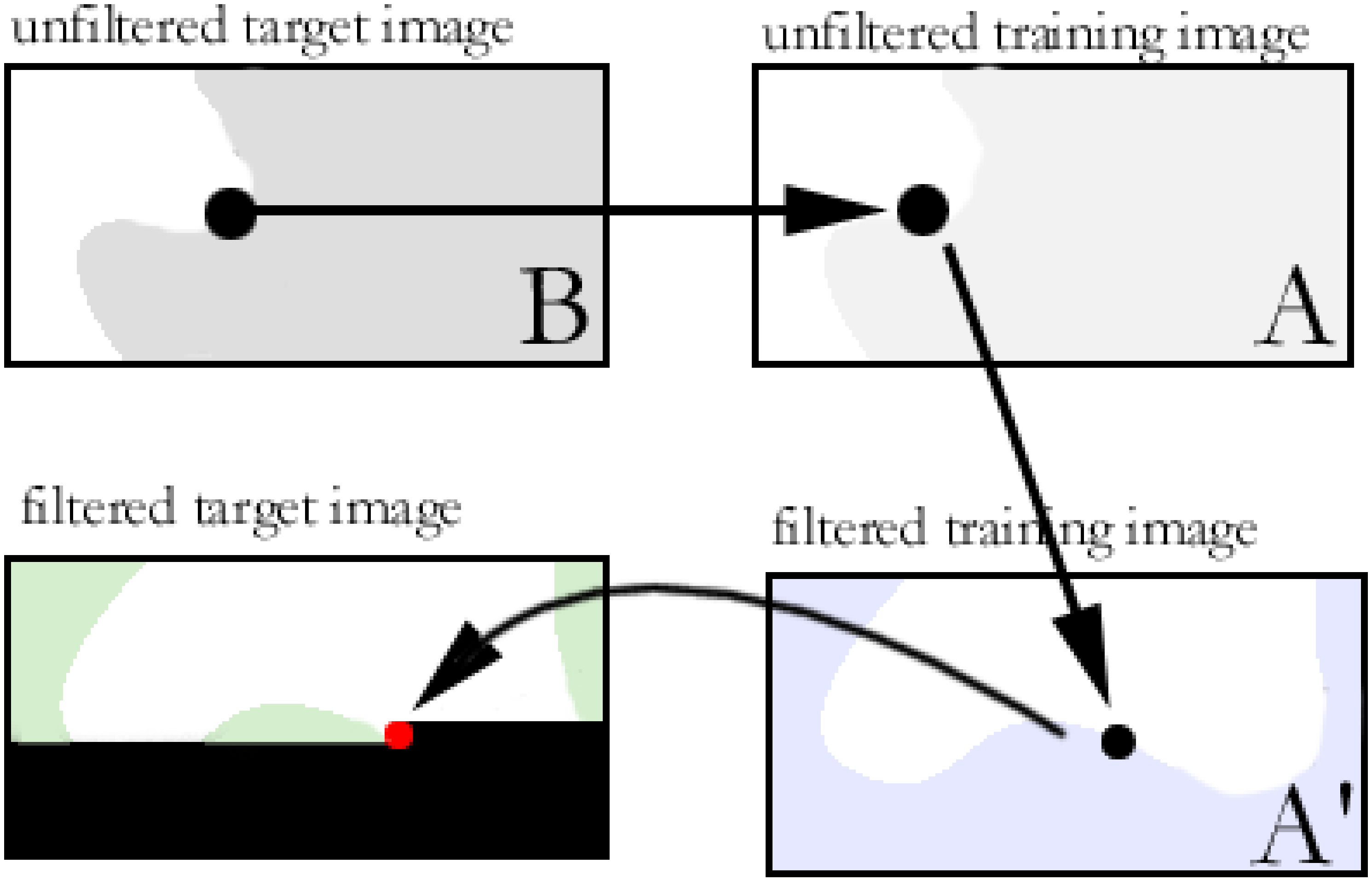


Image Analogies Implementation



Balance between approximate and coherence searches

function $\textsc{BestMatch}(A, A', B, B', s, \ell, q)$:

$p_{\text{app}} \leftarrow \textsc{BestApproximateMatch}(A, A', B, B', \ell, q)$

$p_{\text{coh}} \leftarrow \textsc{BestCoherenceMatch}(A, A', B, B', s, \ell, q)$

$d_{\text{app}} \leftarrow \|F_\ell(p_{\text{app}}) - F_\ell(q)\|^2$

$d_{\text{coh}} \leftarrow \|F_\ell(p_{\text{coh}}) - F_\ell(q)\|^2$

if $d_{\text{coh}} \leq d_{\text{app}}(1 + 2^{\ell - L}\kappa)$ **then**

  return p_{coh}

else

  return p_{app}

Learn to blur

Unfiltered source (A)

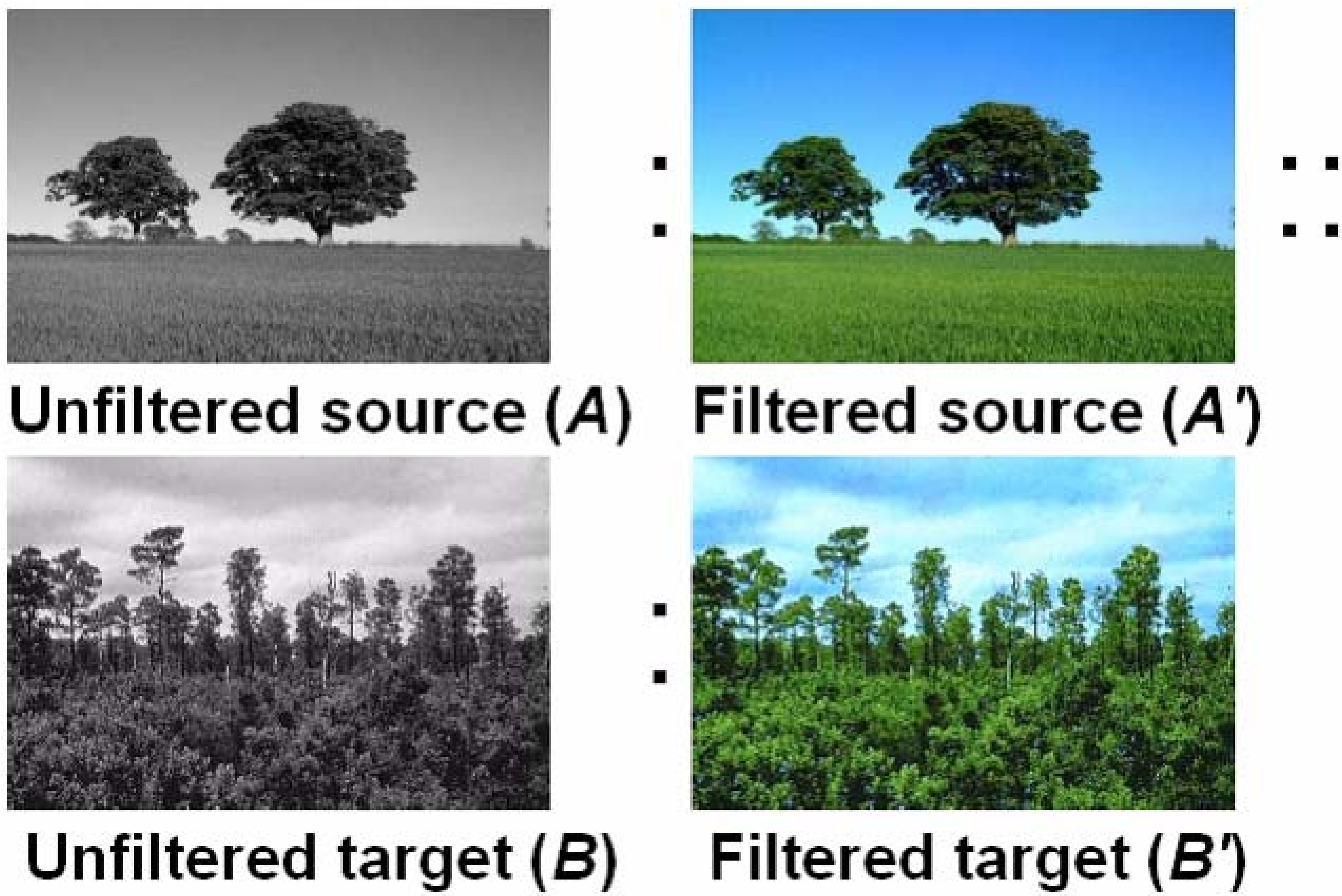
Filtered source (A')

Unfiltered target (B)

Filtered target (B')

Super-resolution


Colorization



Artistic filters



Unfiltered source (A) :: Filtered source (A') ::

::  : 

B

B’

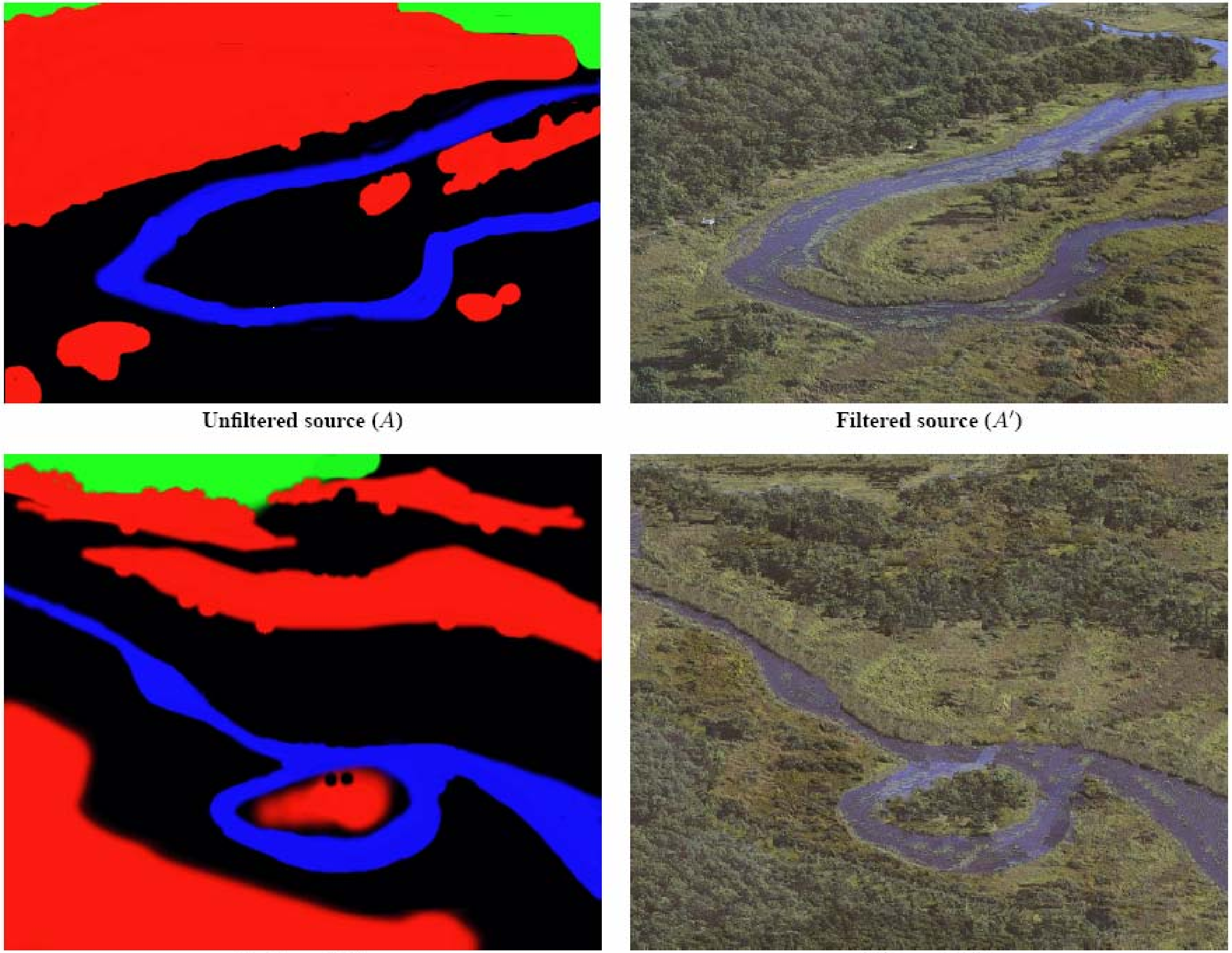
::

:

B

B’

Texture by numbers



Unfiltered source (A) Filtered source (A')

Unfiltered (B) Filtered (B')

Texture by numbers

Image Analogies

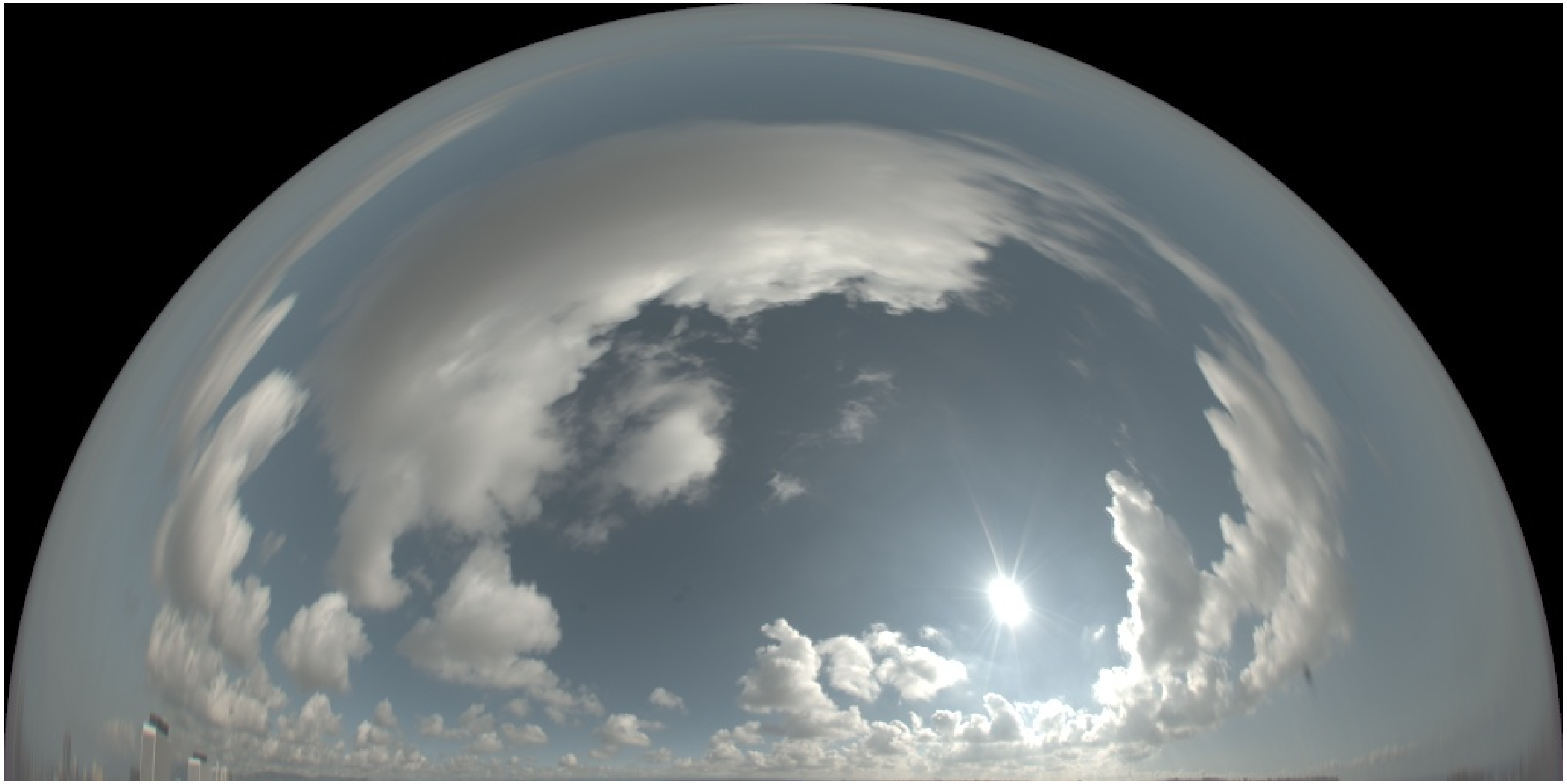
Aaron Hertzmann
Charles Jacobs
Nuria Oliver
Brian Curless
David Salesin

Image-based lighting

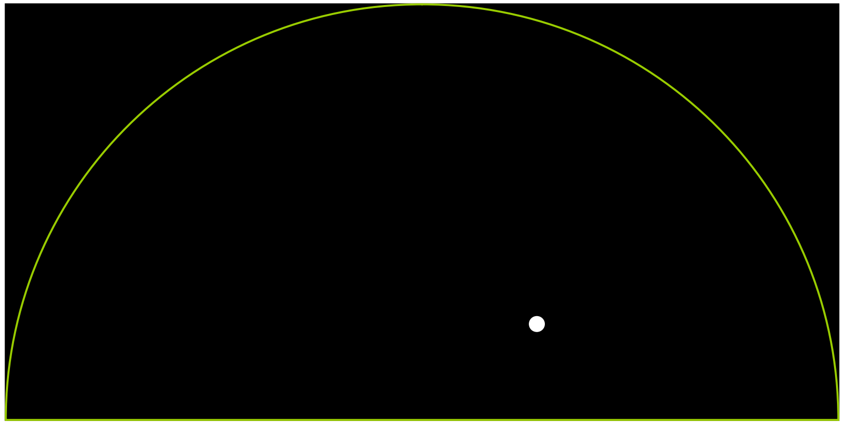
Rendering

- Rendering is a function of geometry, reflectance, lighting and viewing.
- To synthesize CGI into real scene, we have to match the above four factors.
- Viewing can be obtained from *calibration* or *structure from motion*.
- Geometry can be captured using *3D photography* or made by hands.
- How to capture lighting and reflectance?

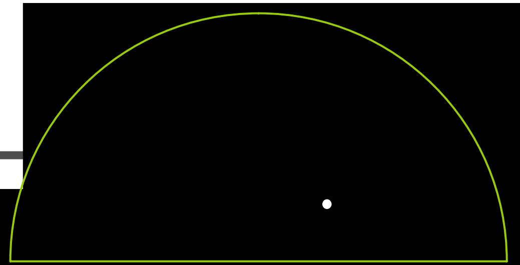
HDRI Sky Probe



Clipped Sky + Sun Source



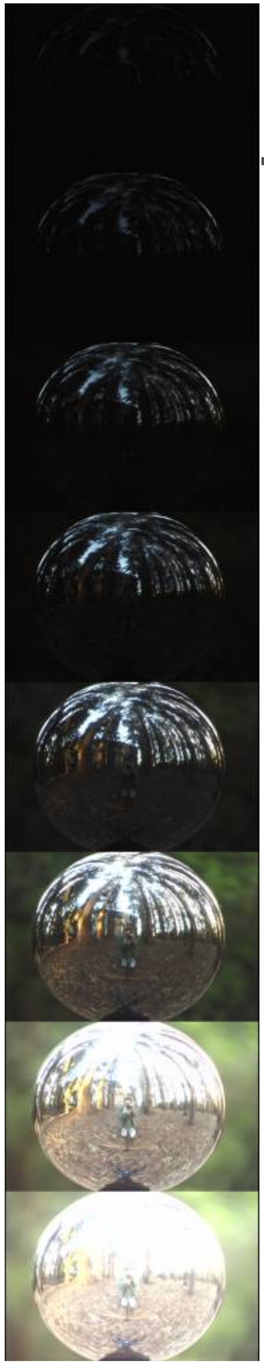
Lit by sun only

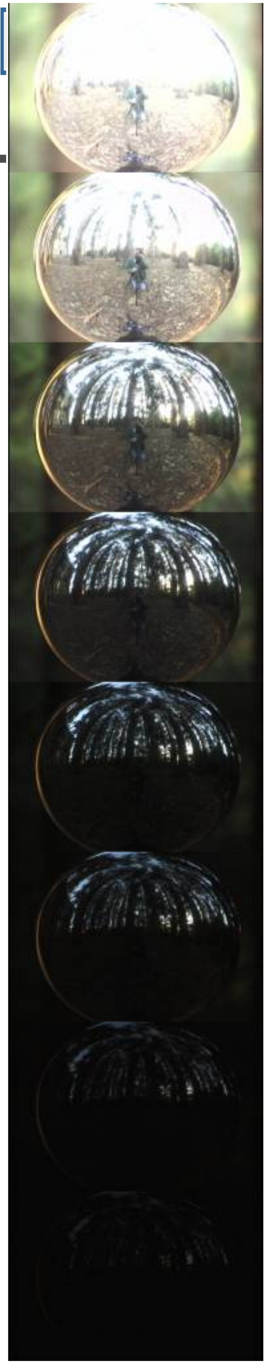


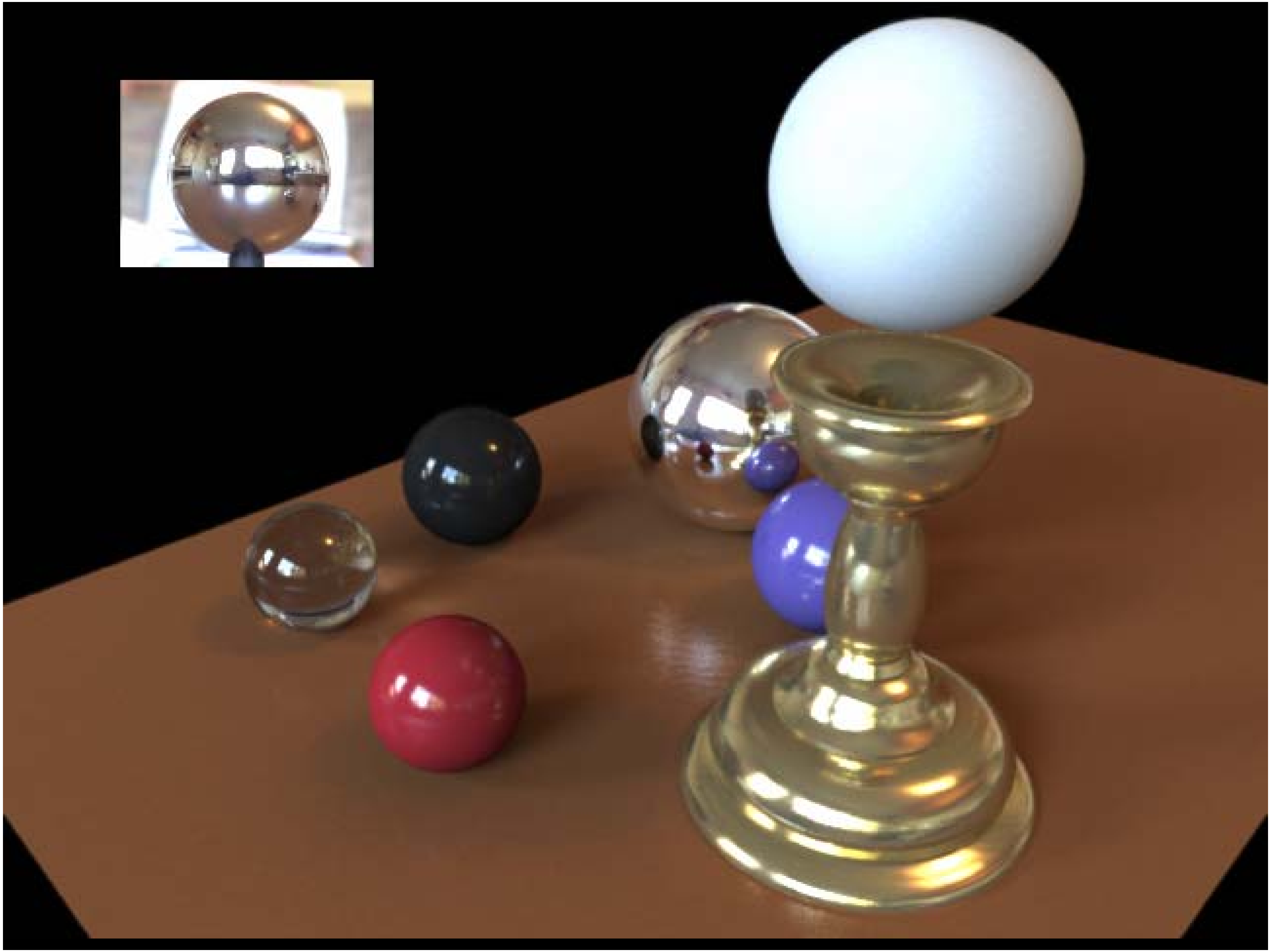
Lit by sky only

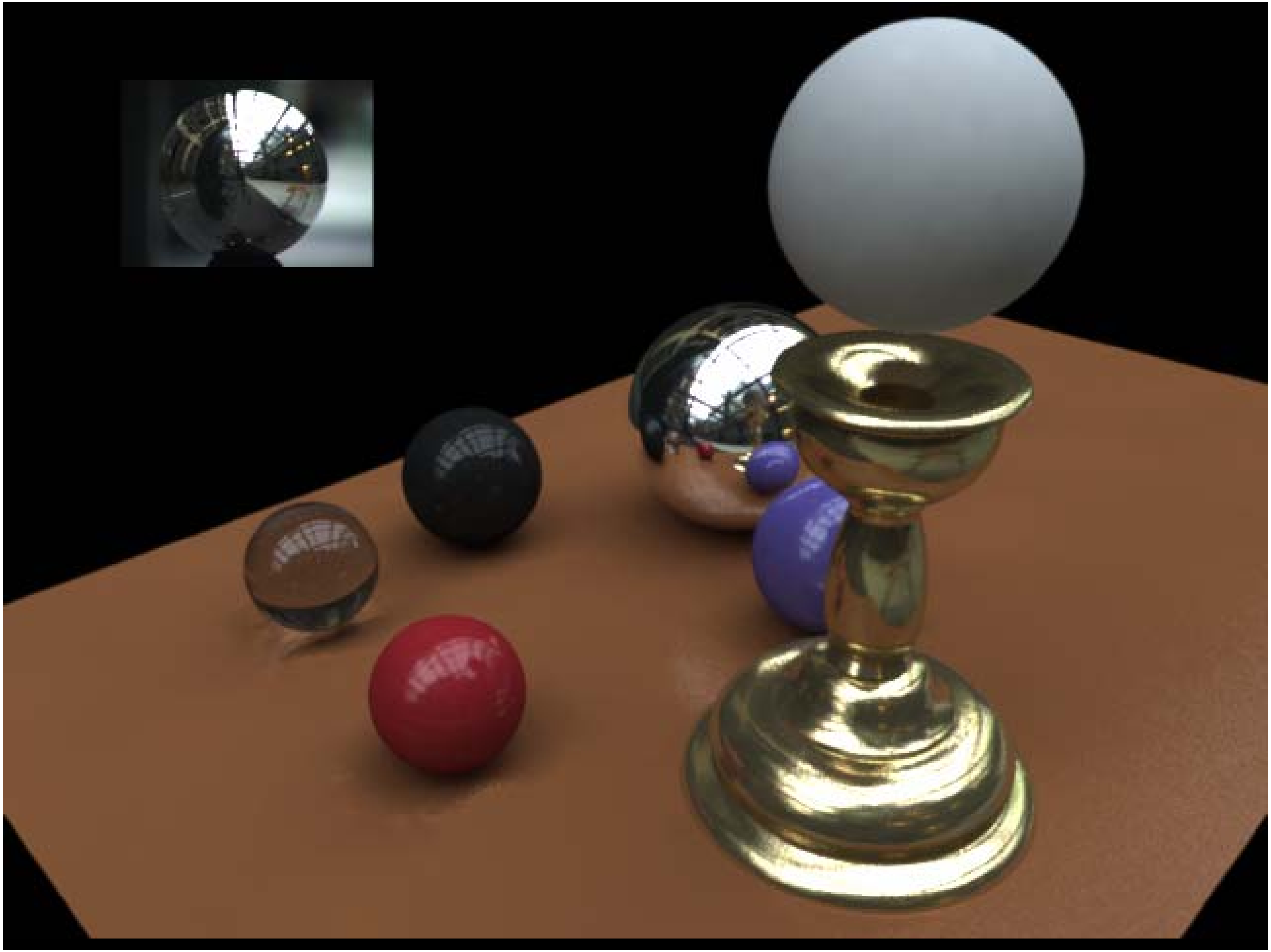
Lit by sun and sky

Acquiring the Light Probe





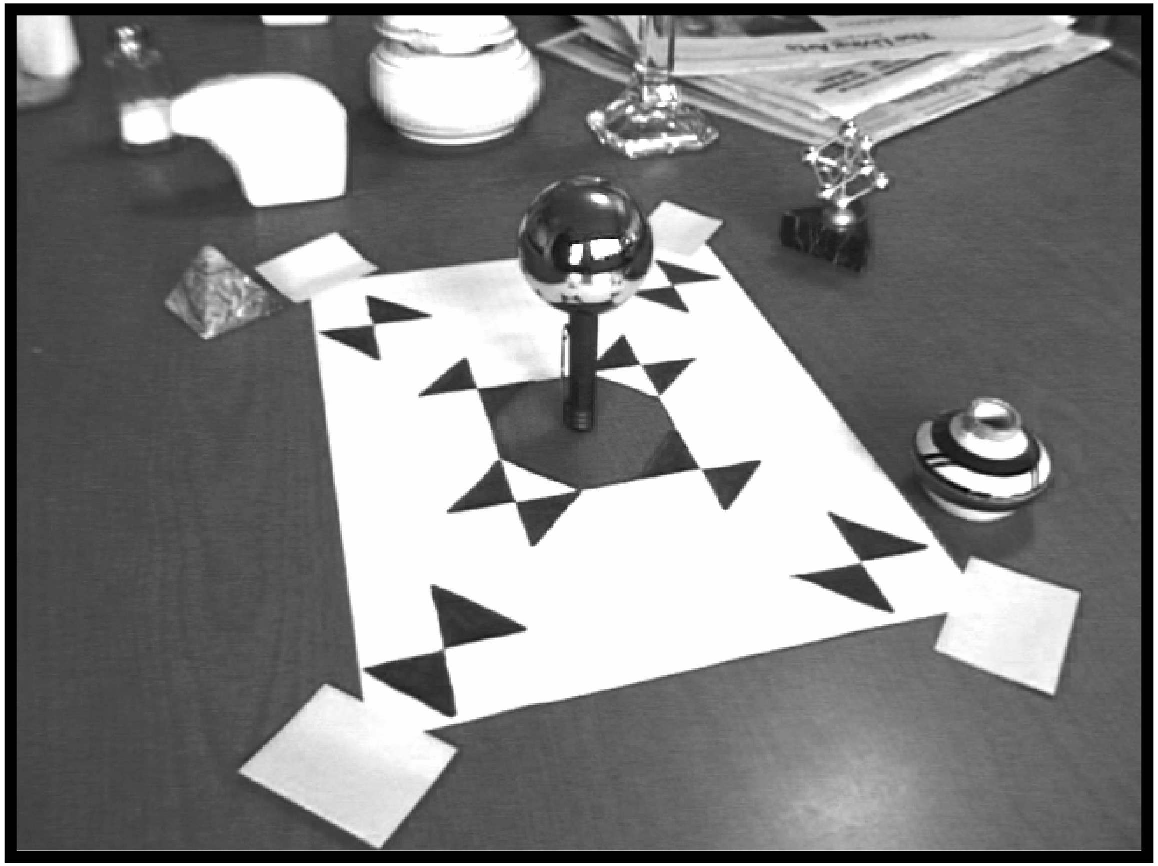




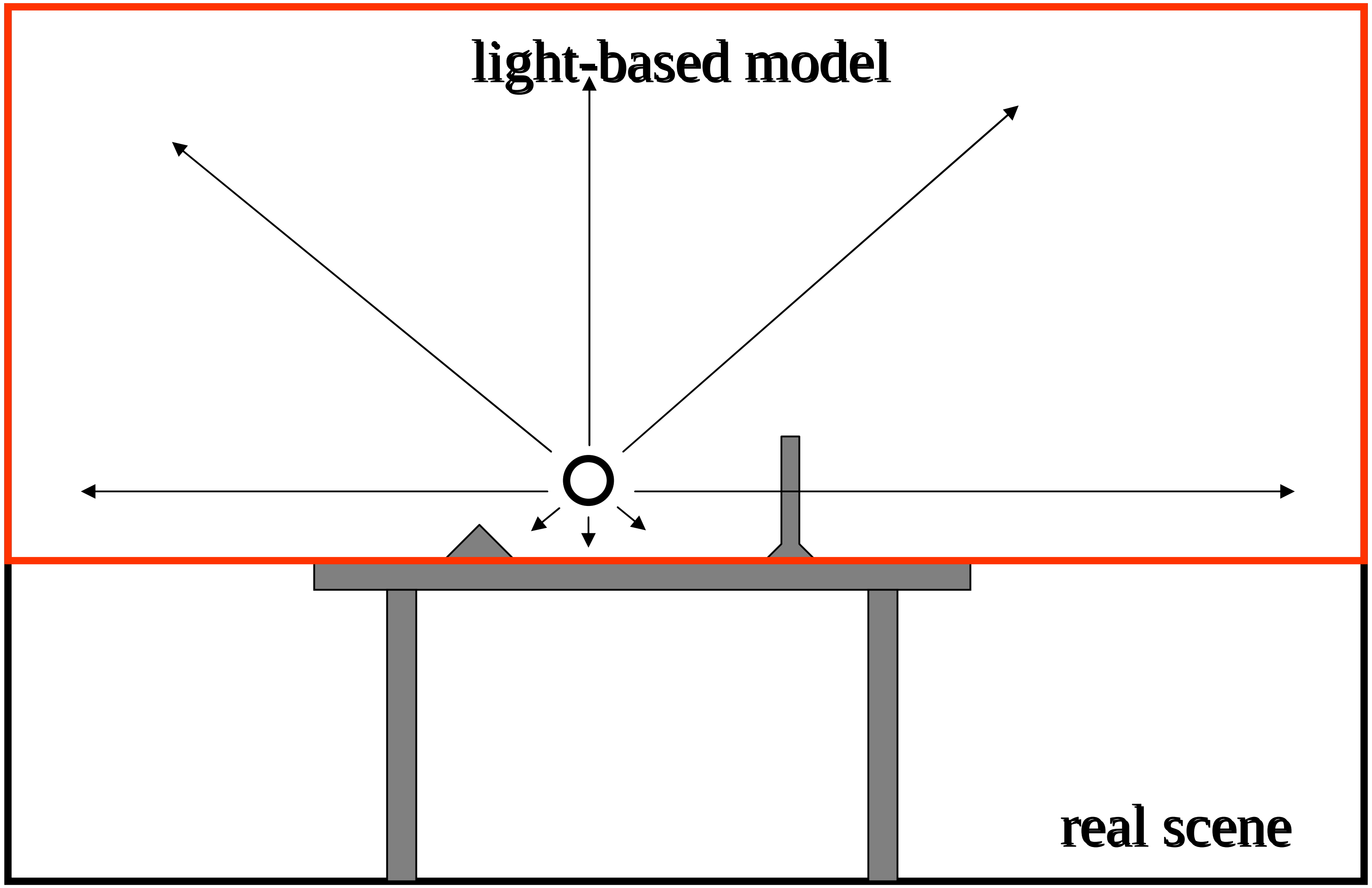
Real Scene Example

- Goal: place synthetic objects on table

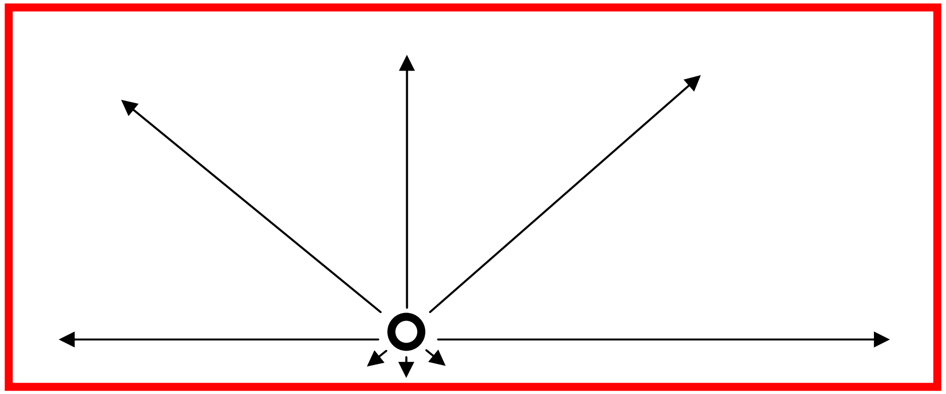
Light Probe / Calibration Grid



Modeling the Scene



The *Light-Based* Room Model



Rendering into the Scene

- Background Plate

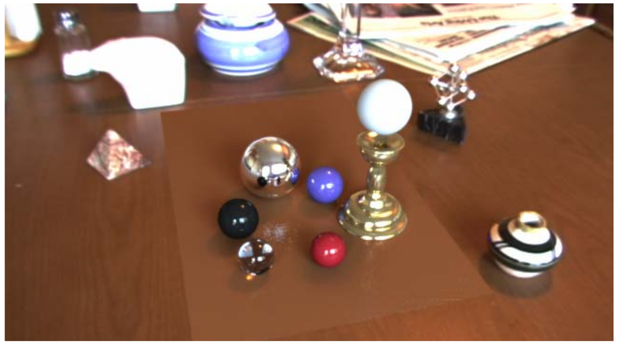
Rendering into the scene

- Objects and Local Scene matched to Scene

Differential rendering

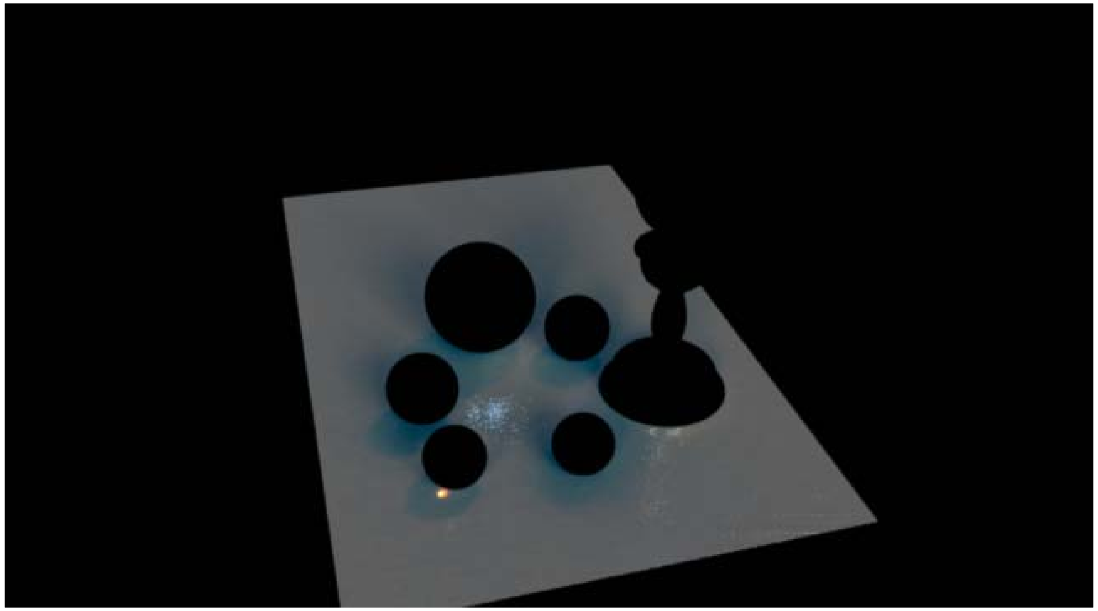
- Local scene w/o objects, illuminated by model

Differential rendering



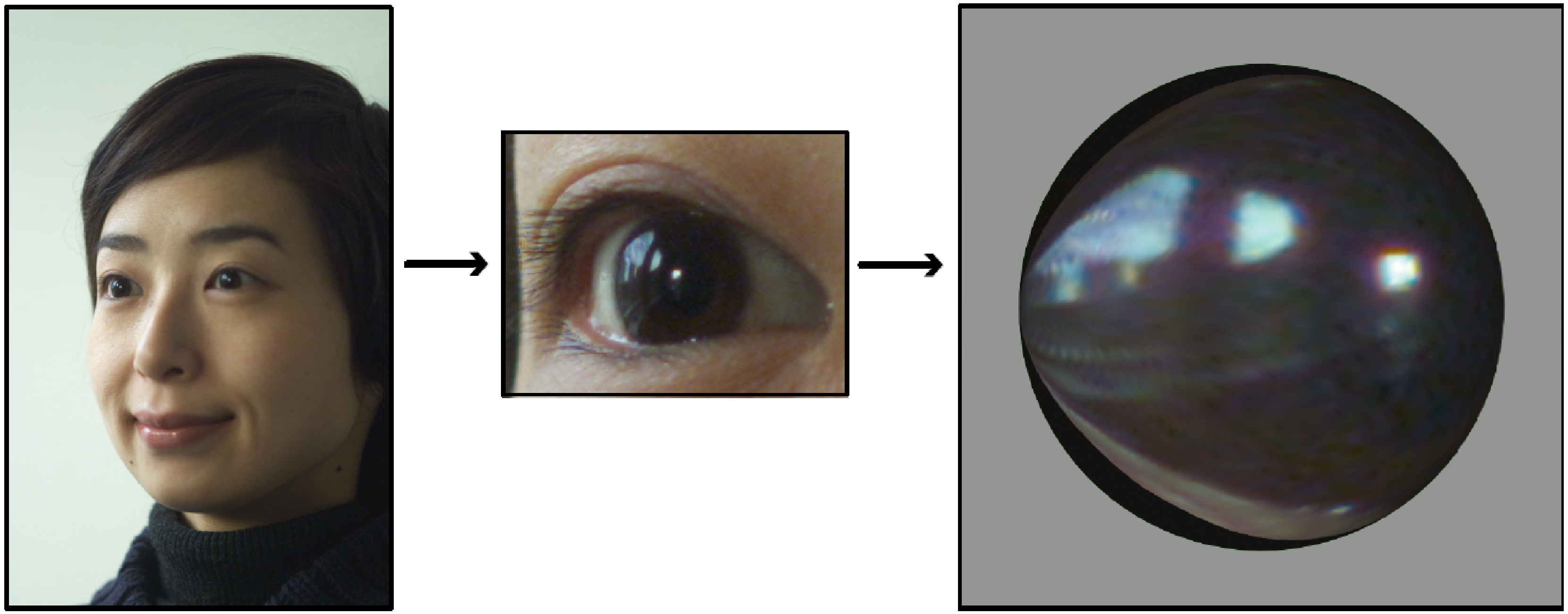
−

=



Environment map from single image?

Eye as light probe! (Nayar et al)



Cornea is an ellipsoid

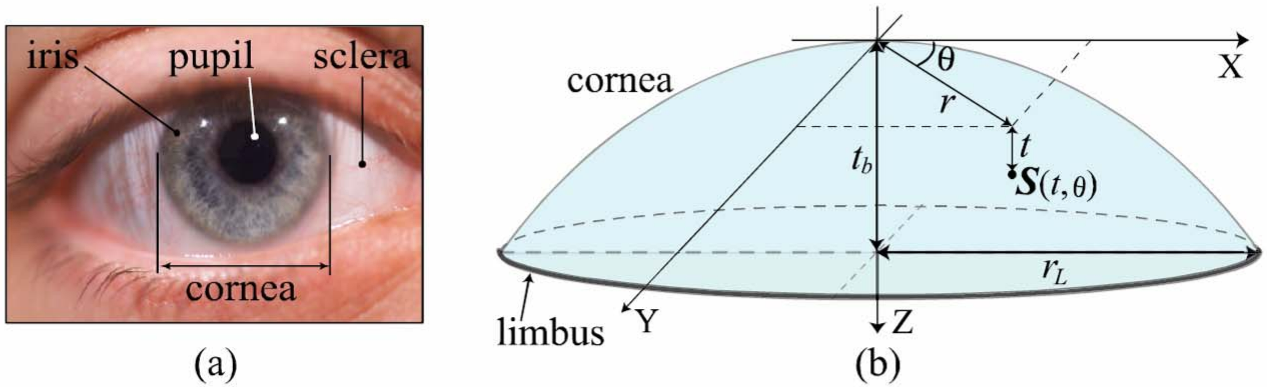
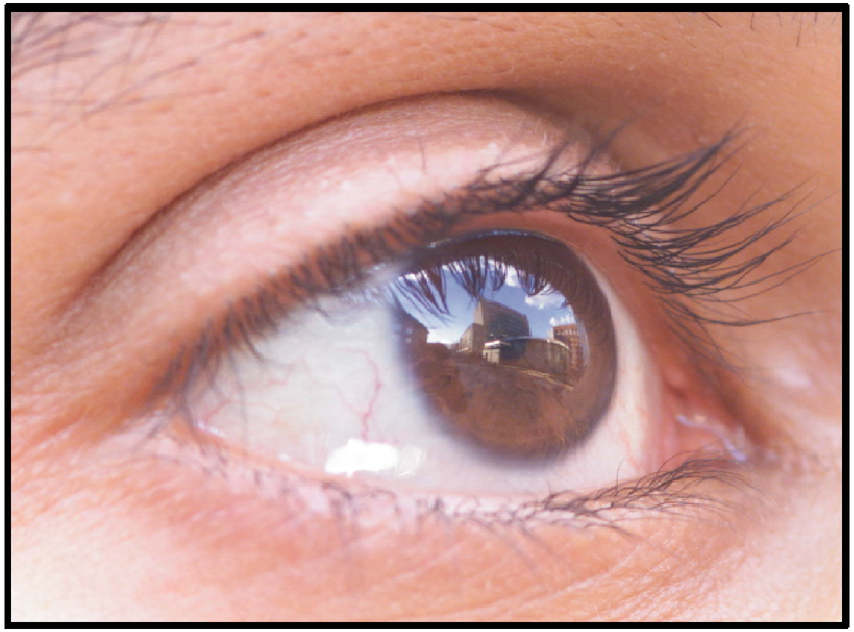


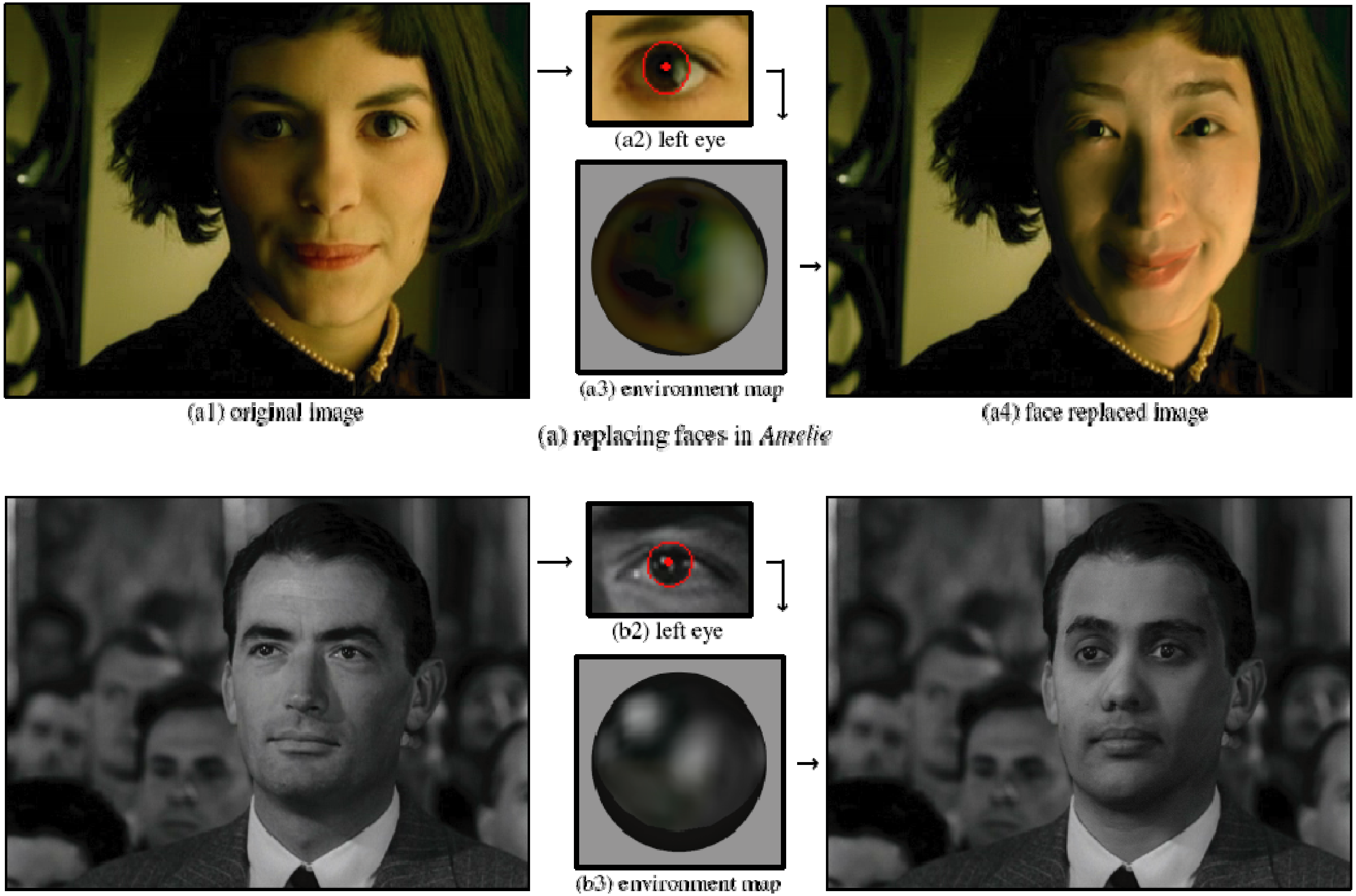
Figure 2: (a) An external view of the human eye. (b) A normal adult cornea can be modeled as an ellipsoid whose outer limit corresponds to the limbus. The eccentricity and radius of curvature at the apex can be assumed to be known.

Ellipsoid fitting



Results

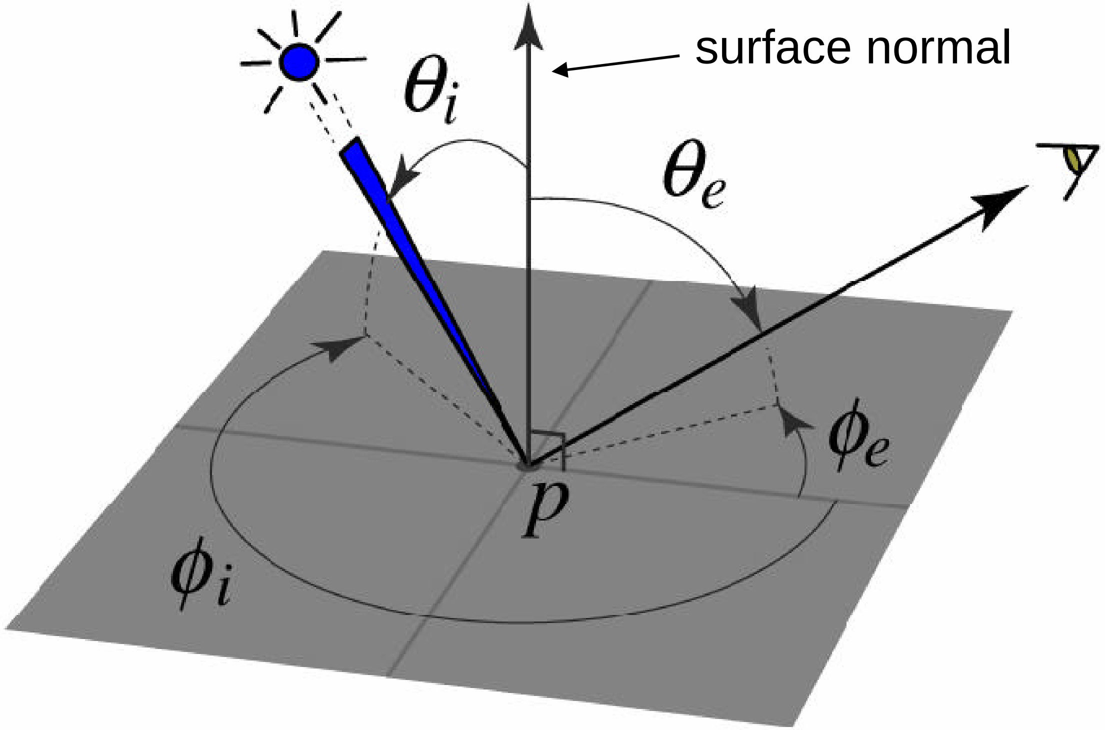
DigiVFX



(a) replacing faces in *Amelie*

Reflectance

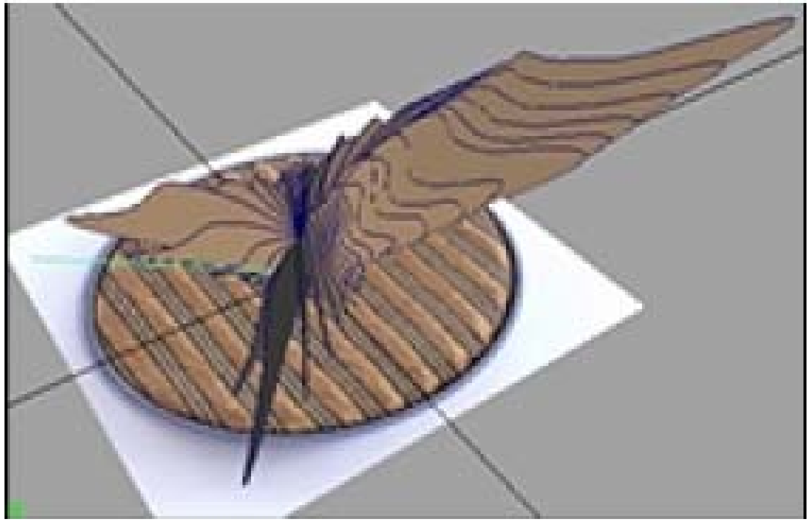
- The Bidirectional Reflection Distribution Function
  – Given an incoming ray (θ_i, ϕ_i) and outgoing ray (θ_e, ϕ_e) what proportion of the incoming light is reflected along out

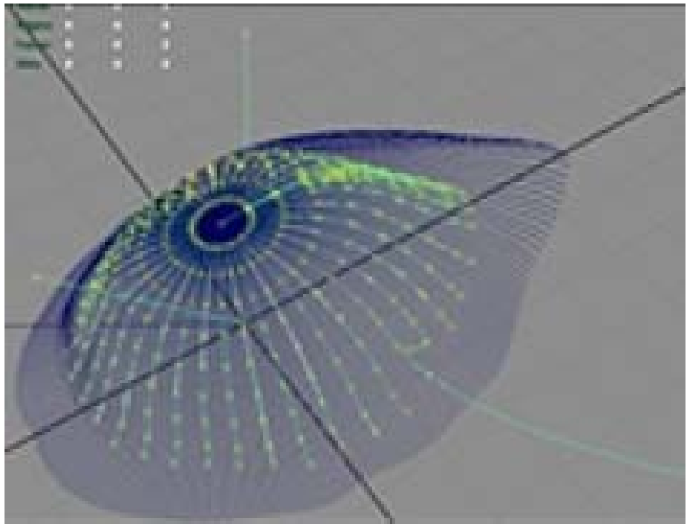


Answer given by the BRDF: $\rho(\theta_i, \phi_i, \theta_e, \phi_e)$

Capturing reflectance

DigiVFX





Application in "The Matrix Reloaded"

Reference

- Alexei A. Efros, Thomas K. Leung, Texture Synthesis by Non-parametric Sampling, ICCV 1999.
- Li-Yi Wei, Marc Levoy, Fast Texture Synthesis Using Tree-Structured Vector Quantization, SIGGRAPH 2000.
- Michael Ashikhmin, Synthesizing Natutal Textures, I3D 2001.
- Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, David H. Salesin, Image Analogies, SIGGRAPH 2001.
- Alexei A. Efros, William T. Freeman, Image Quilting for Texture Synthesis and Transfer, SIGGRAPH 2001.
- Vivek Kwatra, Arno Schodl, Irfan Essa, Greg Turk, Aaron Bobick, Graphcut Textures: Image and Video Texture Synthesis Using Graph Cuts, SIGGRAPH 2003.
- Michael F. Cohen, Jonathan Shade, Stefan Hiller, Oliver Deussen, Wang Tiles for Image and Texture Generation, SIGGRAPH 2003.
- A. Criminisi, P. Perez, K. Toyama, Object Removal by Examplar-Based Inpainting, CVPR 2003.
- Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David H. Salesin, Michael F. Cohen, Interactive Digital Photomontage, SIGGRAPH 2004.

Reference

- Paul Debevec, Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography, SIGGRAPH 1998.
- Haarm-Pieter Duiker, Lighting Reconstruction for "The Matrix Reloaded", SIGGRAPH 2003 Sketch and Applications.
- George Borshukov, Measured BRDF in Film Production - Realistic Cloth Appearance for "The Matrix Reloaded", SIGGRAPH 2003 Sketch and Applications.
- Ko Nishino, Shree K. Nayar, Eyes for Relighting, SIGGRAPH 2004.