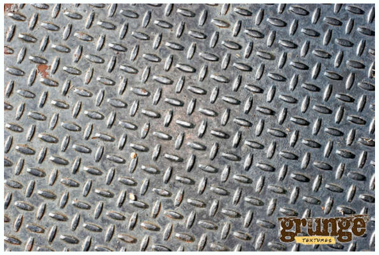
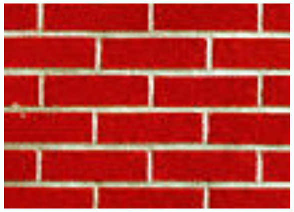
Texture

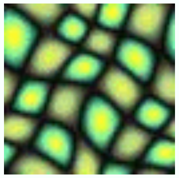
CS 419

Slides by Ali Farhadi

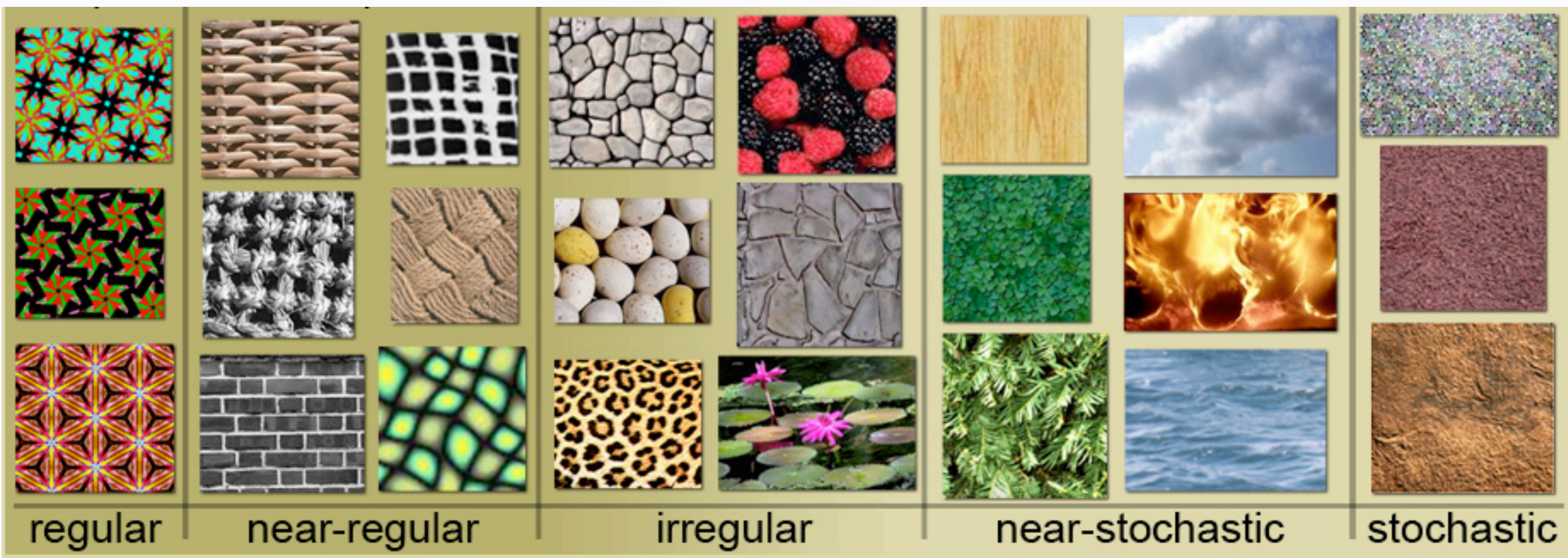
What is a Texture?





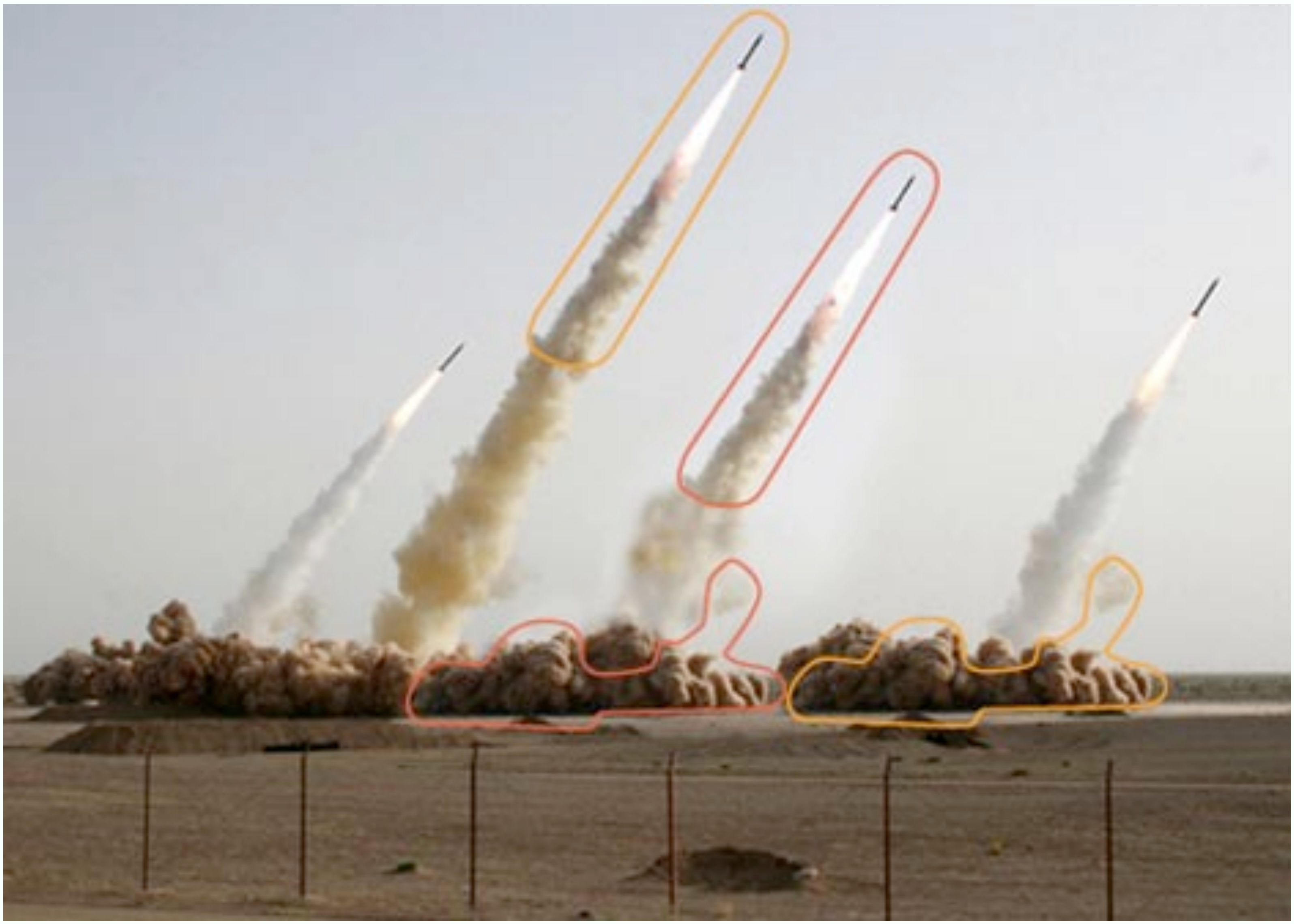


Texture Spectrum



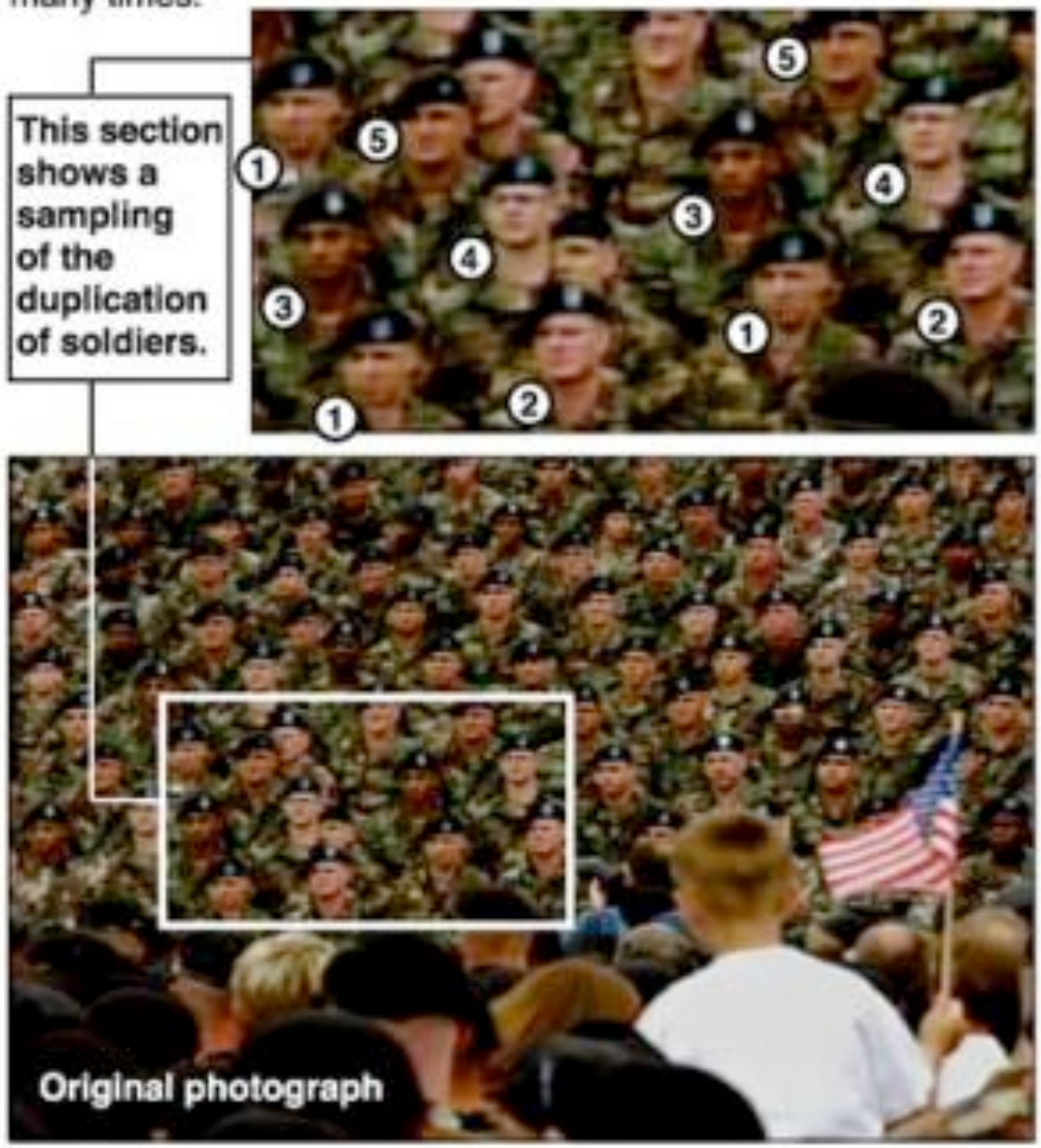
Steven Li, James Hays, Chenyu Wu, Vivek Kwatra, and Yanxi Liu, CVPR 06

Texture scandals!!



Bush campaign digitally altered TV ad

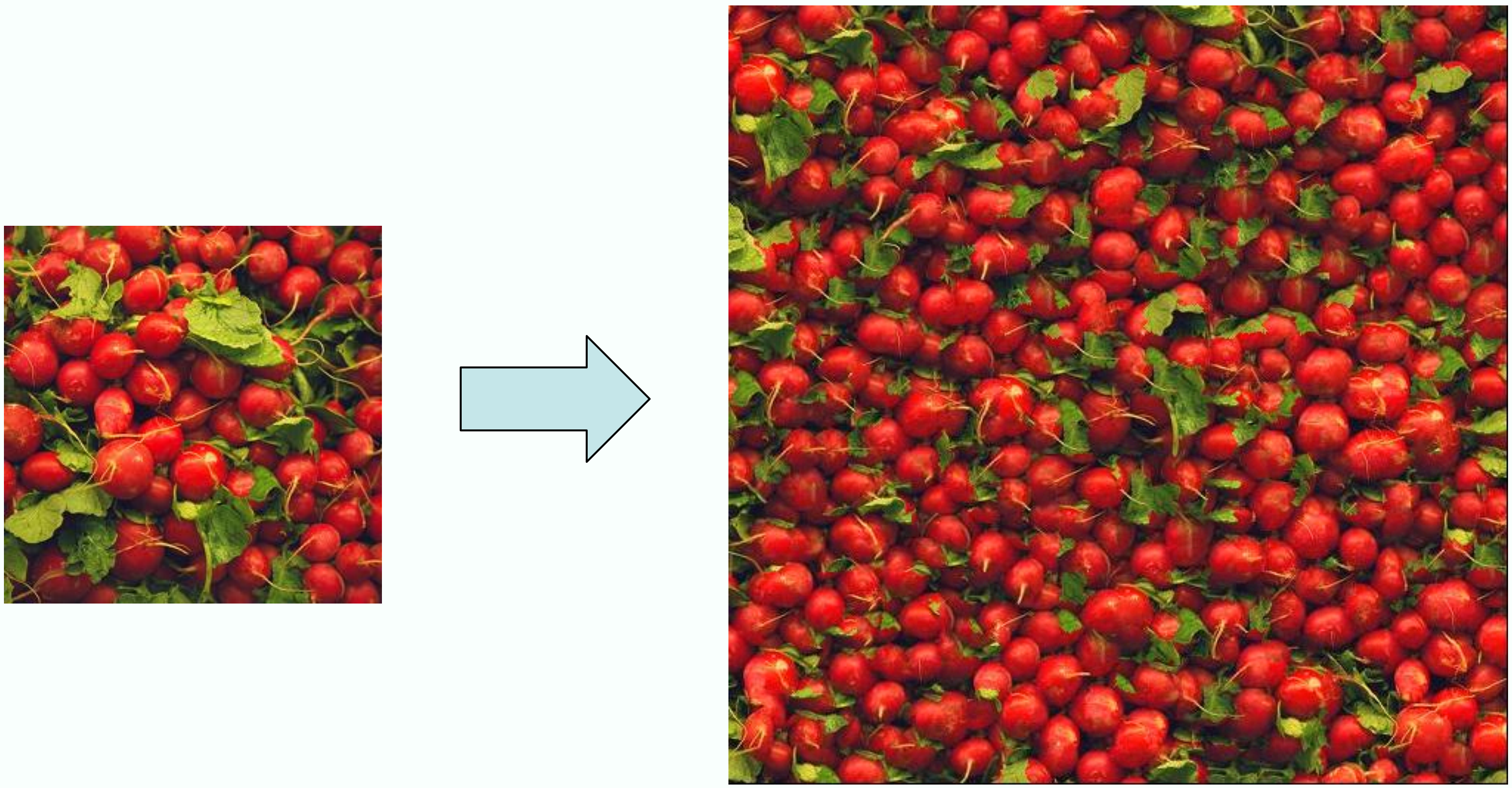
President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.



Two crucial algorithmic points

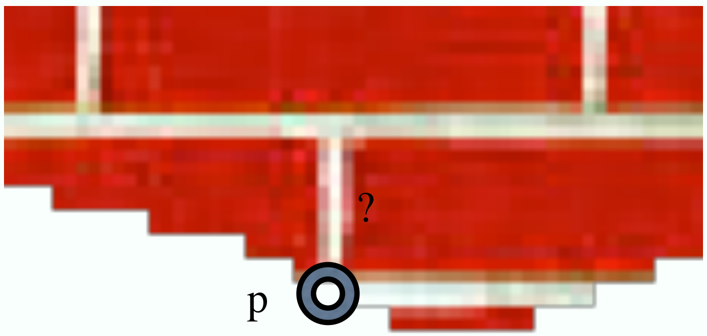
- Nearest neighbors
  - again and again and again
- Dynamic programming
  - likely new; we’ll use this again, too

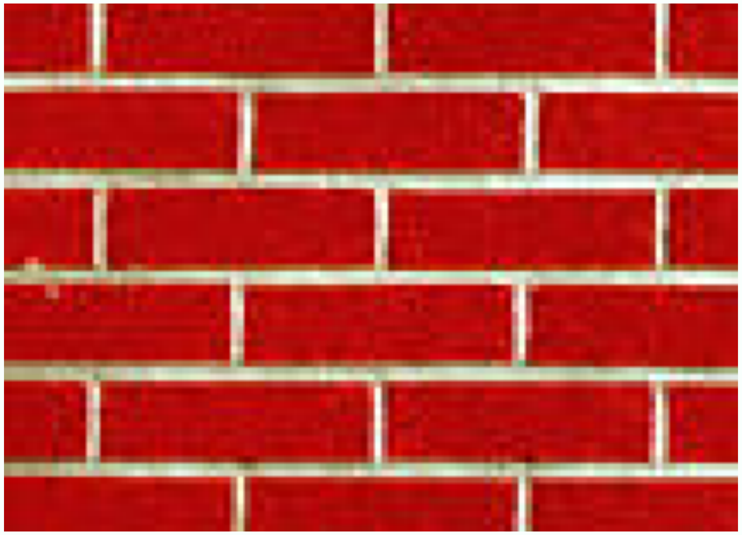
Texture Synthesis



Efros & Leung ICCV99

How to paint this pixel?

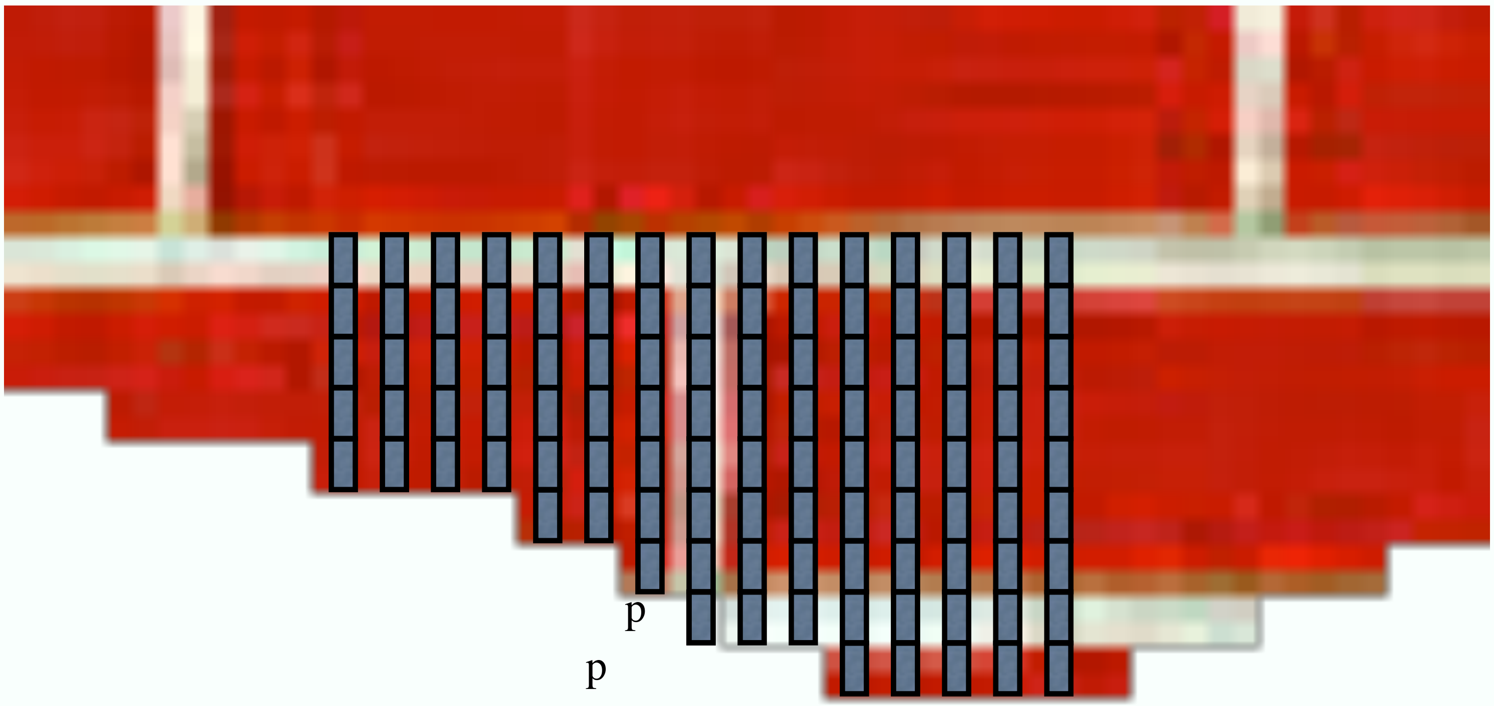




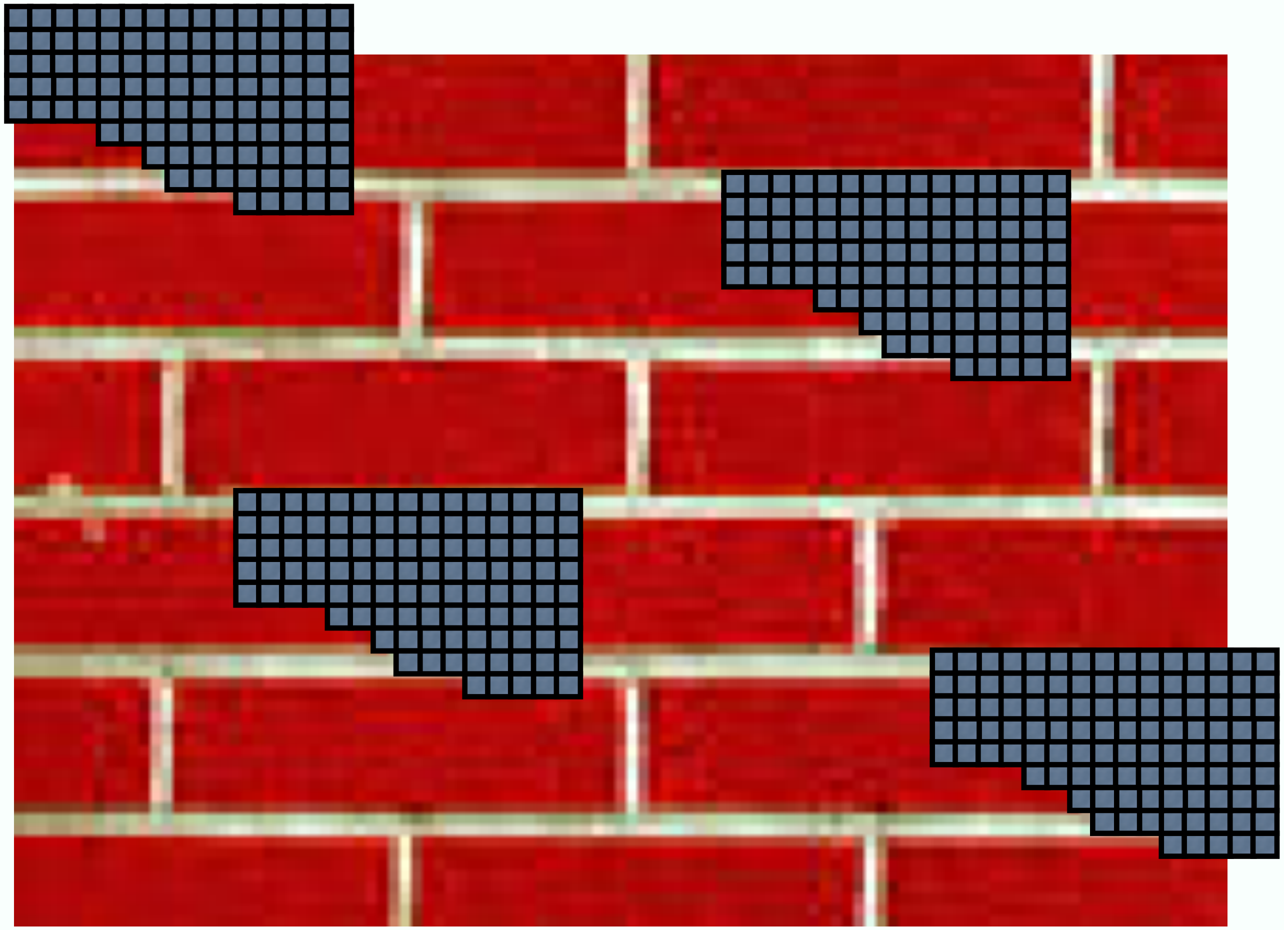
Input texture

Efros & Leung ICCV99

Ask Neighbors



- What is the conditional probability distribution of p, given it's neighbors?

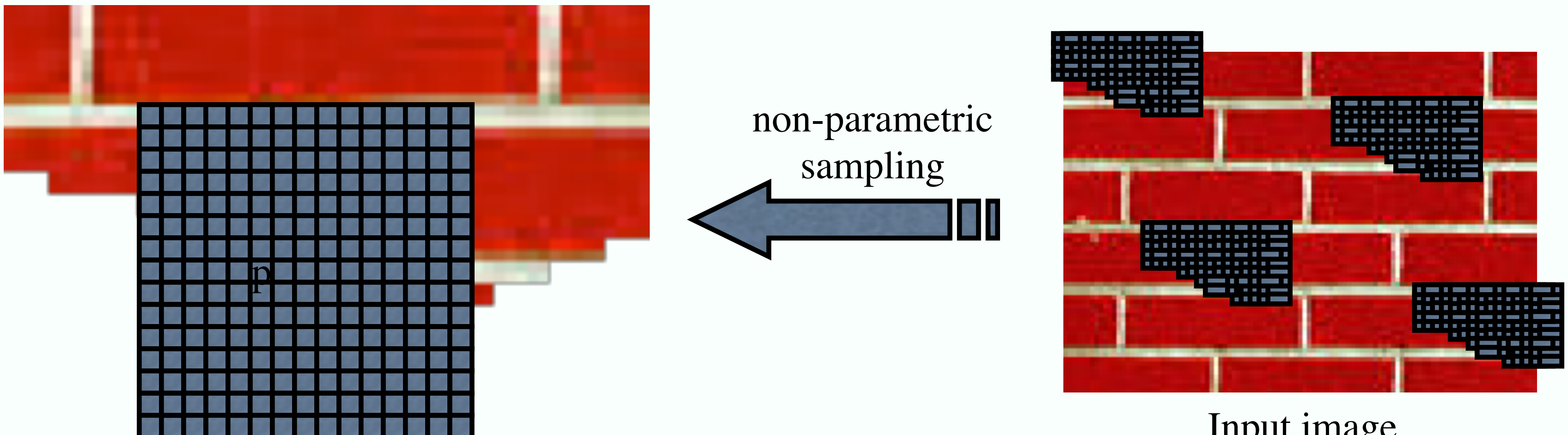


Input image

- Don’t bother to model the distribution
  - It’s already there, in the image

Efros & Leung ICCV99

Efros & Leung Algorithm



Synthesizing a pixel

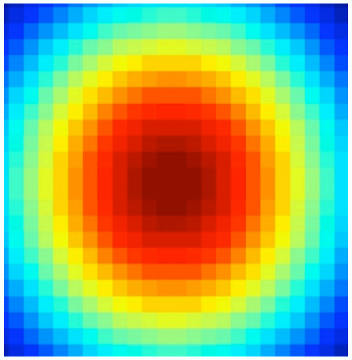
Input image

Concerns

- Distance metric
- Neighborhood size
- Order to paint

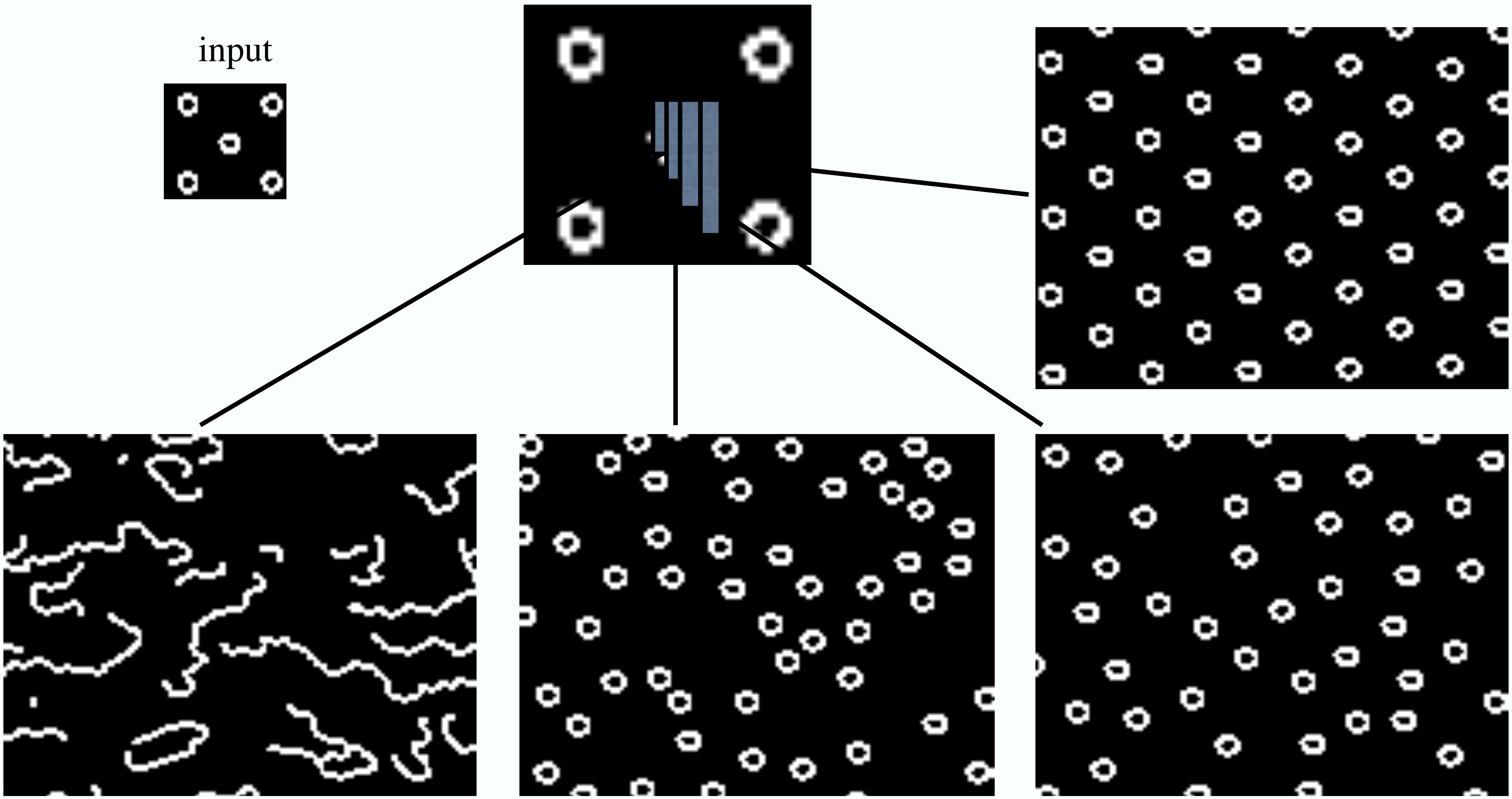
Distance metric

- Normalized sum of squared distances
- Not all the neighbors worth the same
  - Gaussian mask

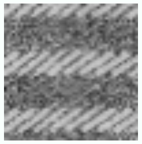


- Preserve the local structure
- Pick among reasonably similar neighborhoods

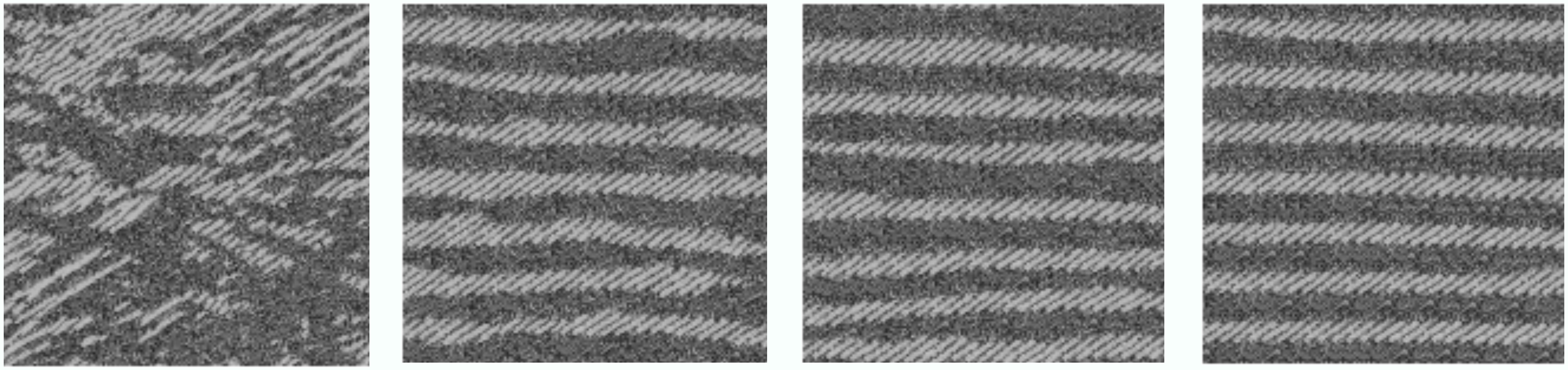
Neighborhood size

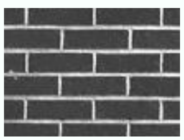


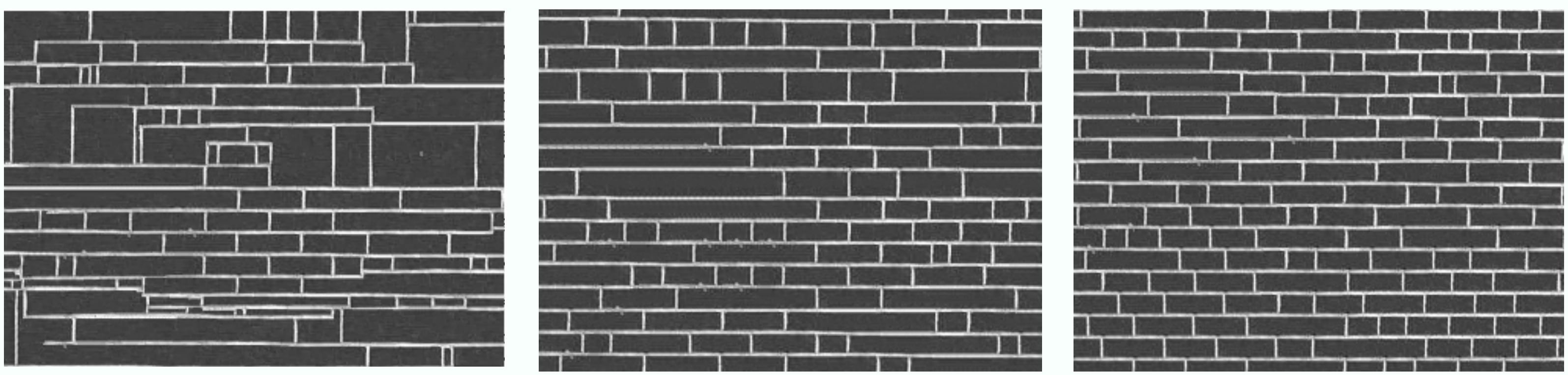
Efros & Leung ICCV99



Varying Window Size



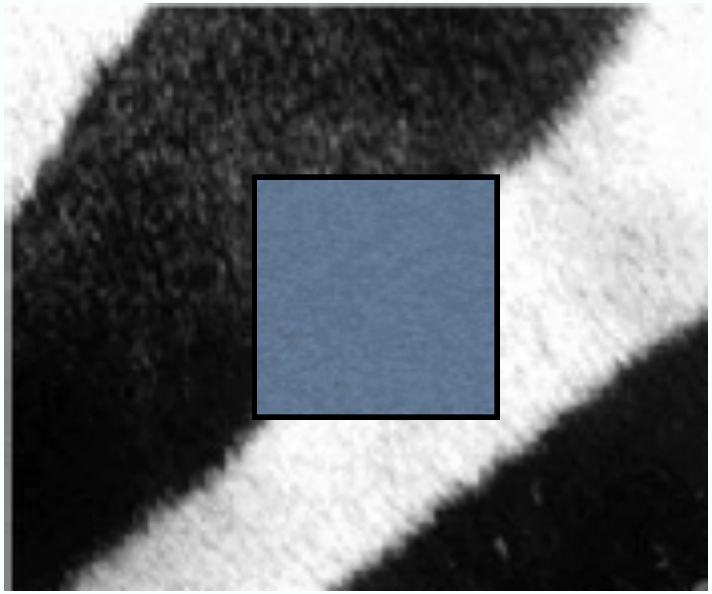




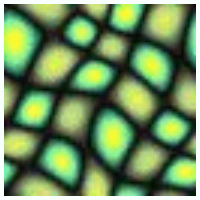
Increasing window size

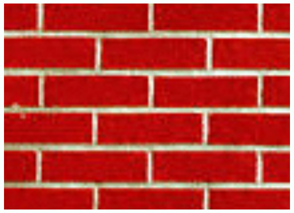
Efros & Leung ICCV99

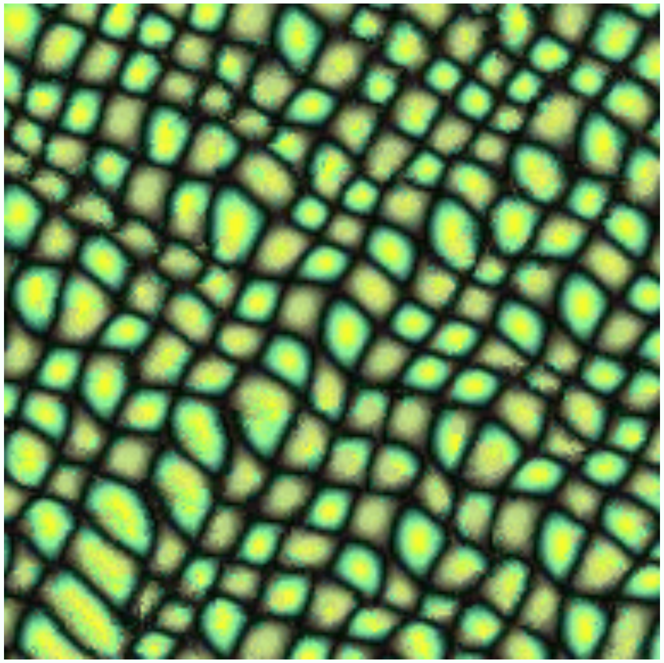
The Order matters

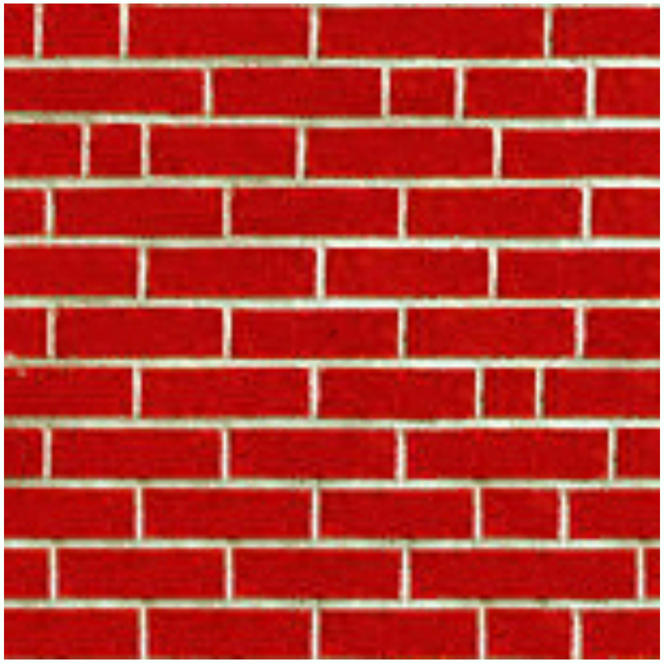


Some Results



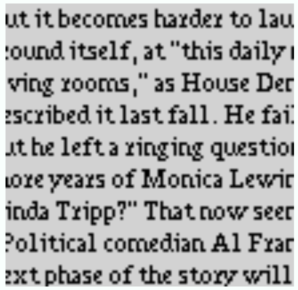


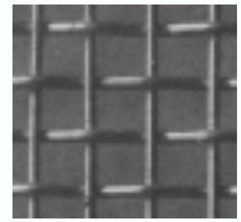


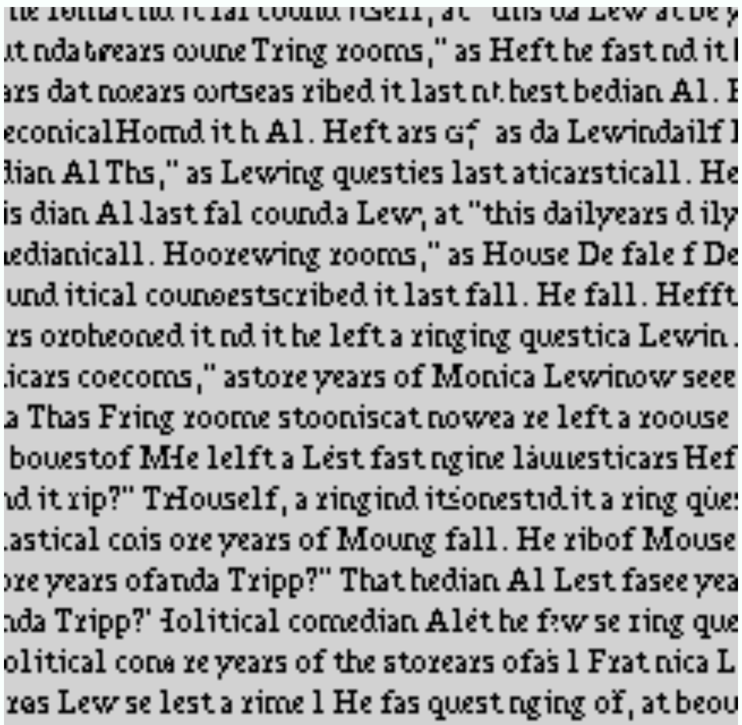


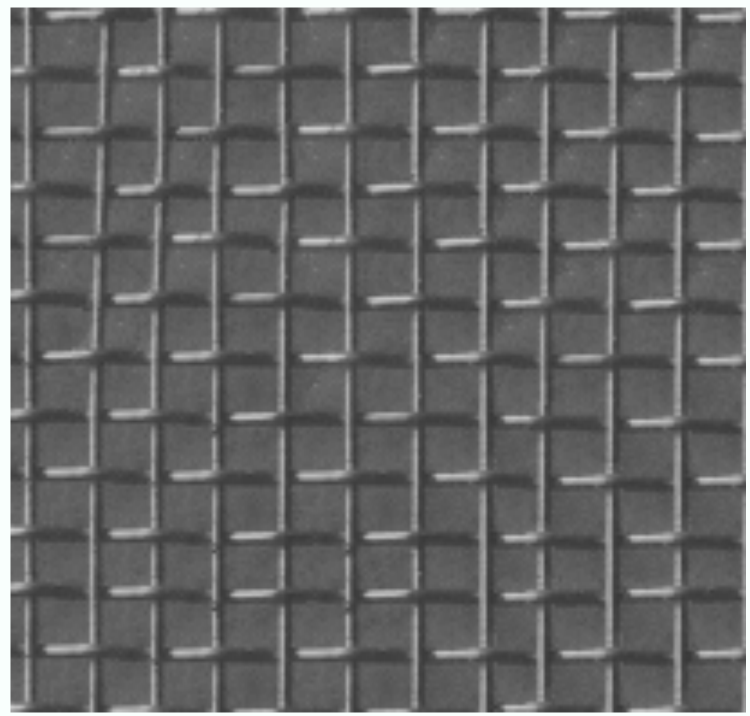
Efros & Leung ICCV99

More Results



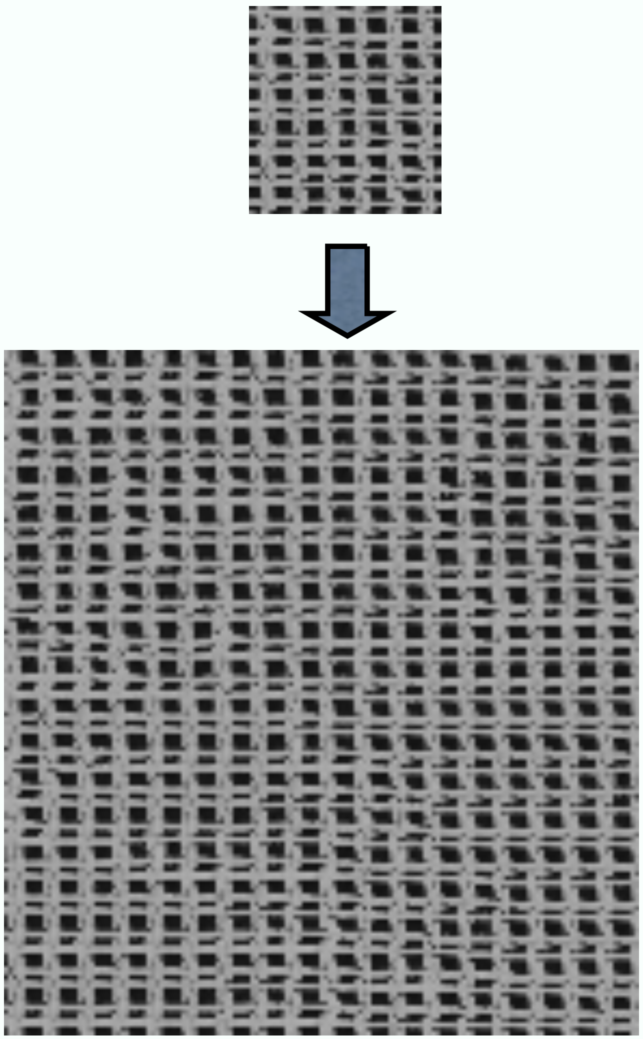




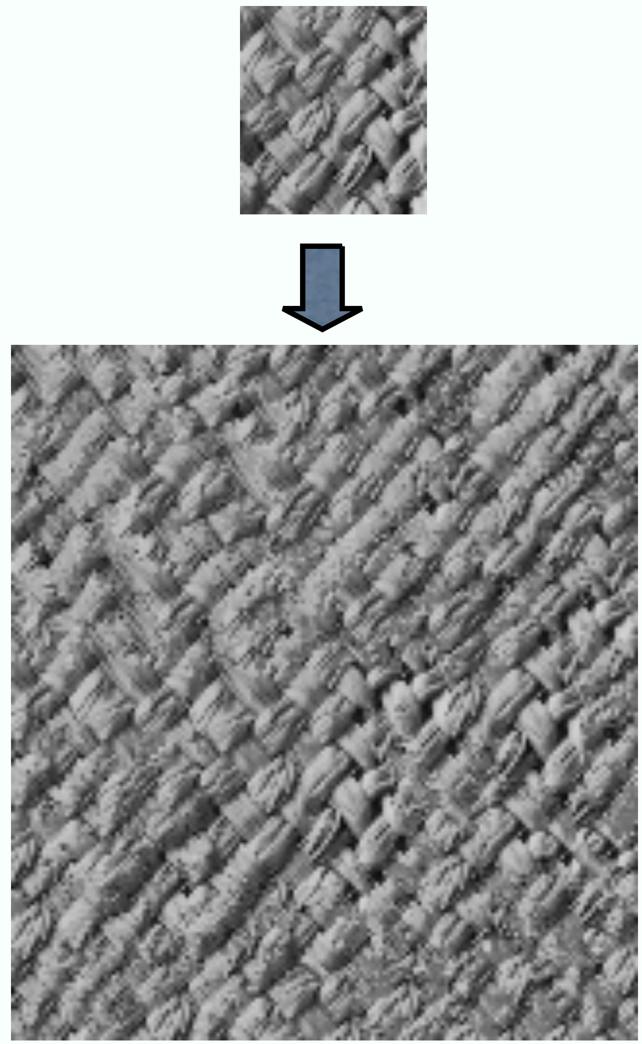


More Results

french canvas



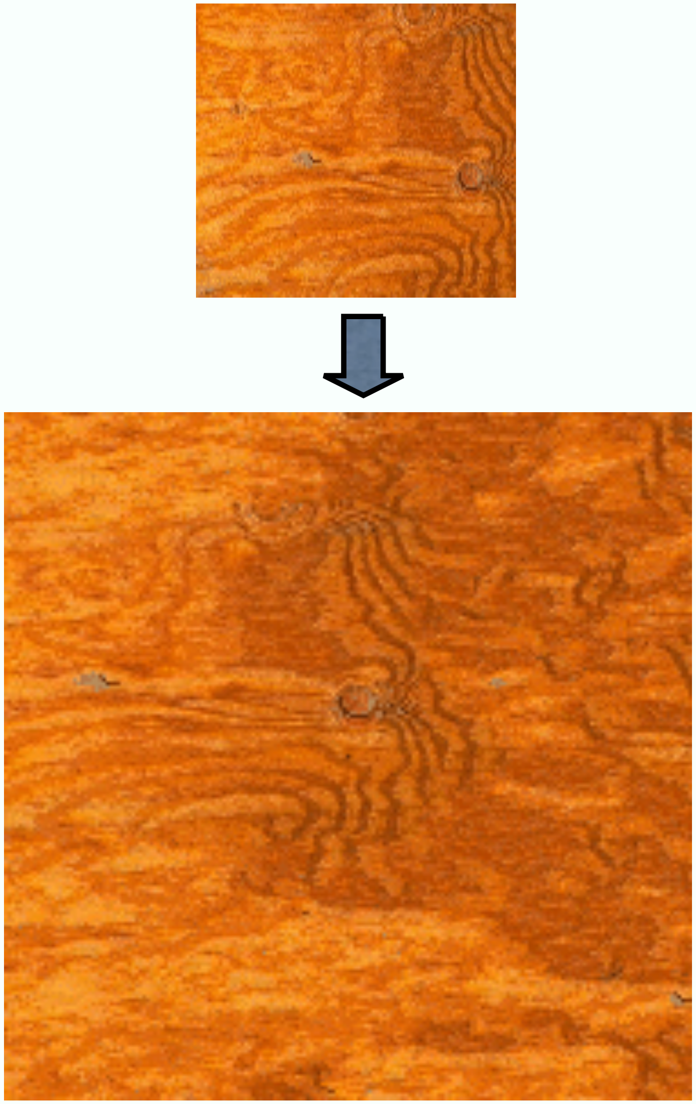
rafia weave



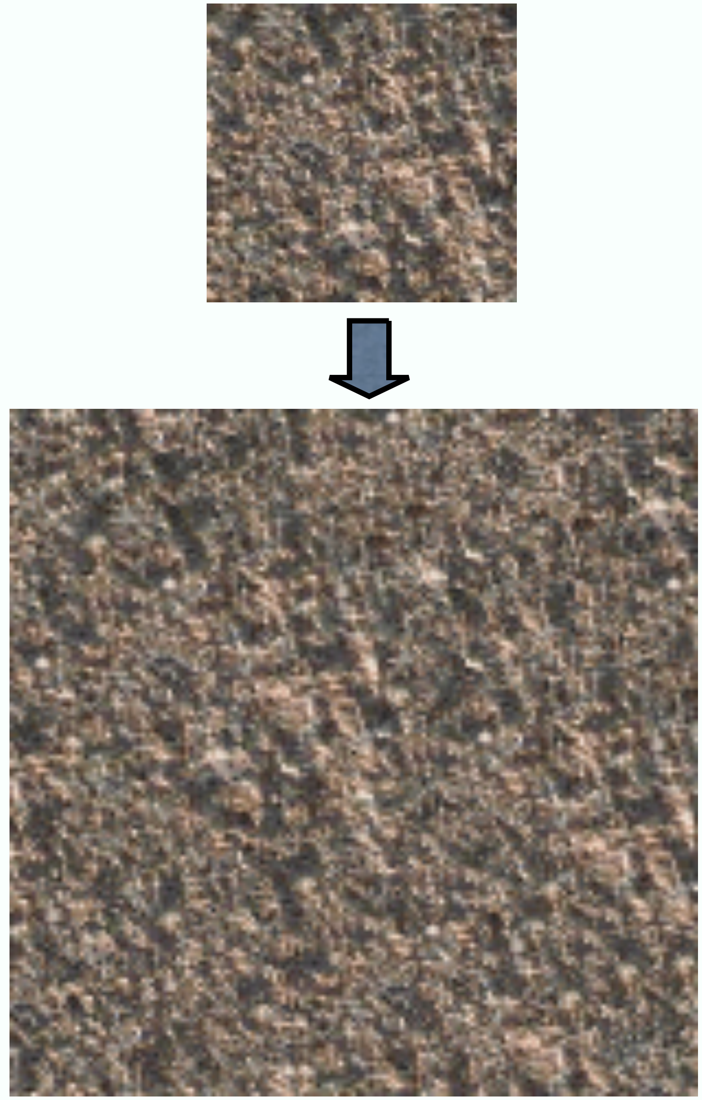
Efros & Leung ICCV99

More Results

wood



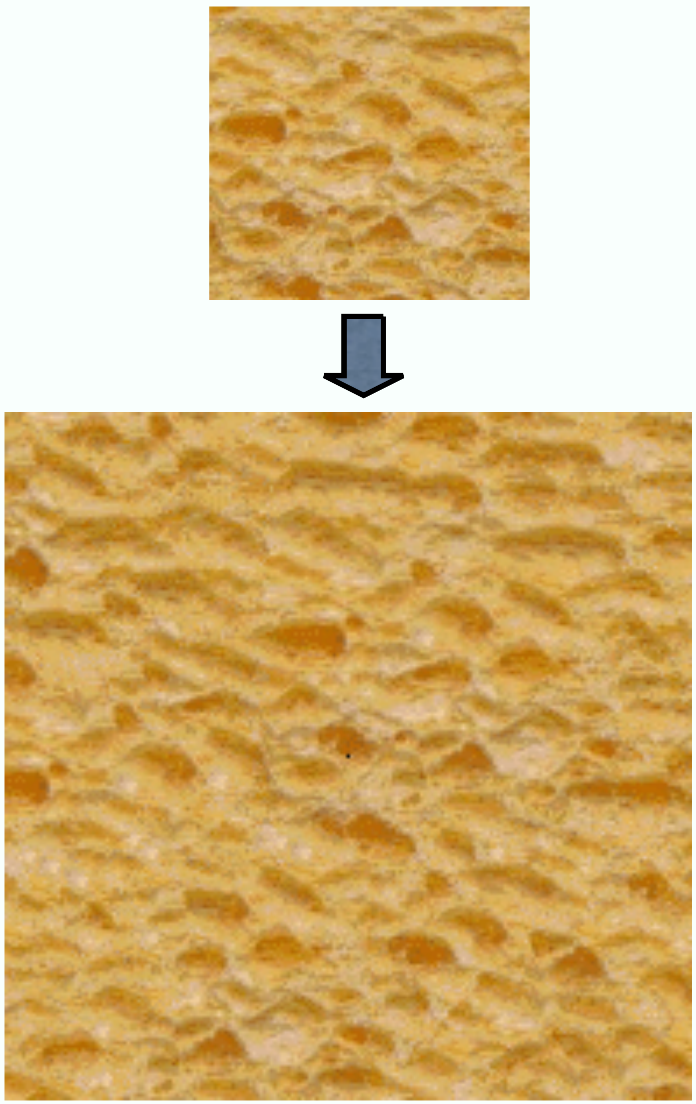
granite



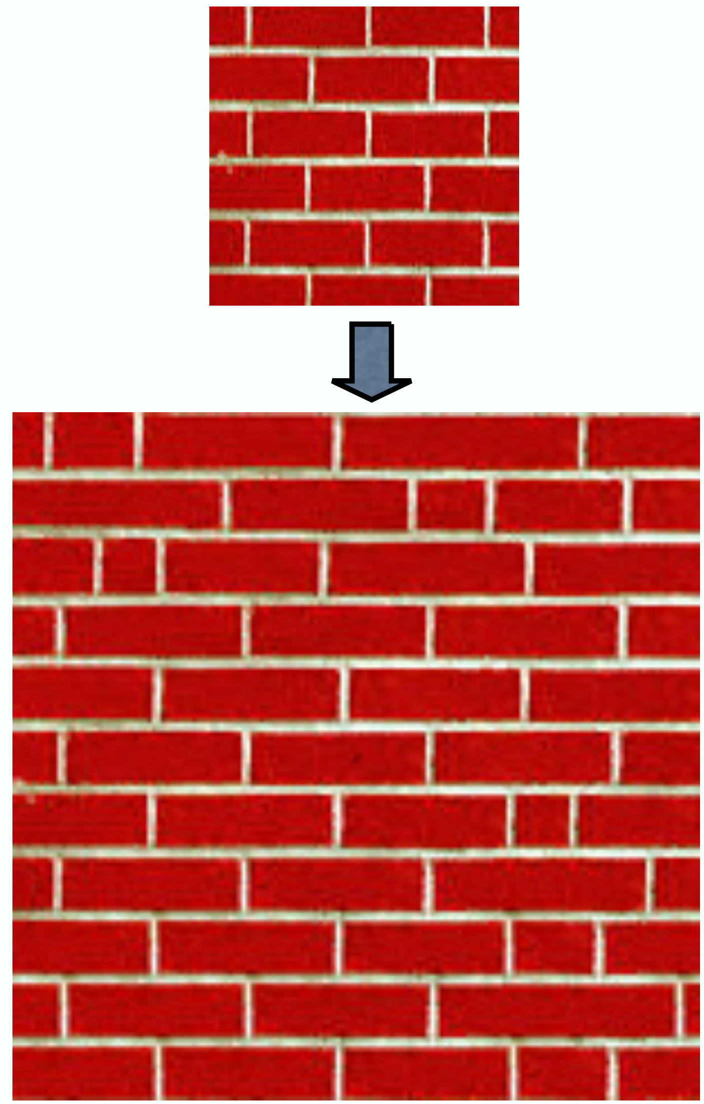
Efros & Leung ICCV99

More Results

white bread

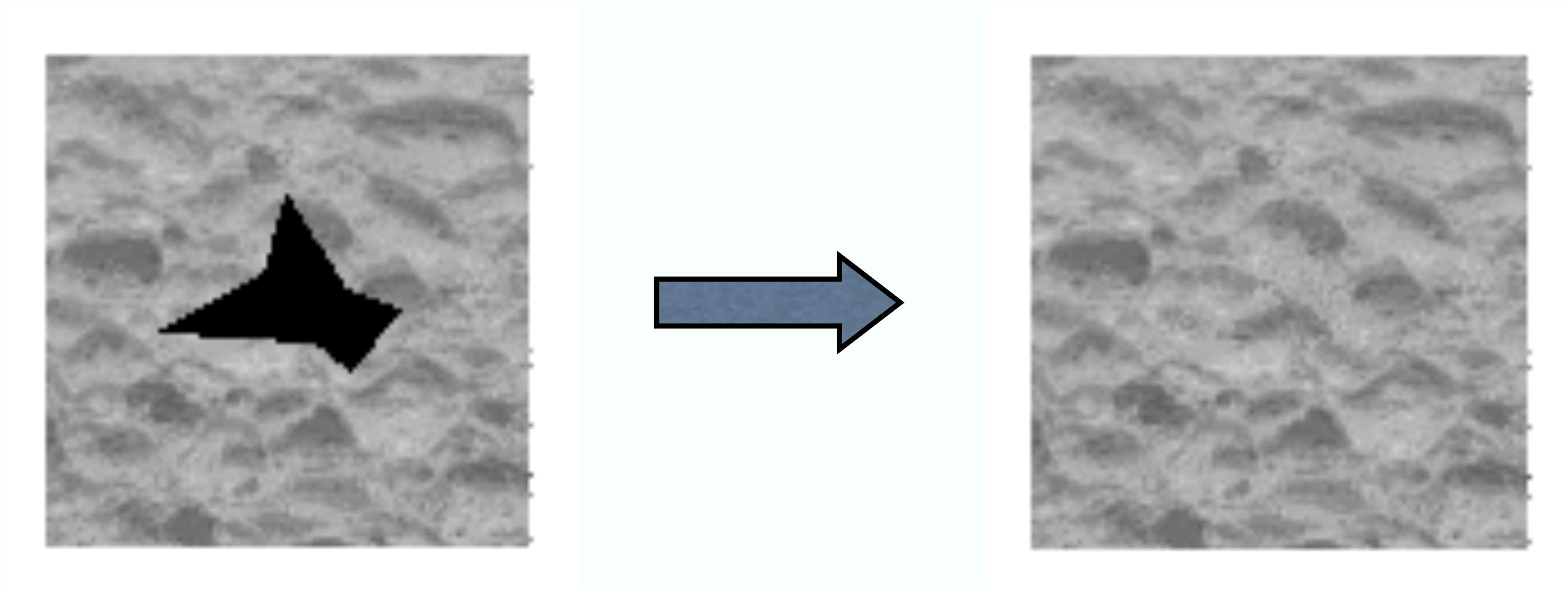


brick wall



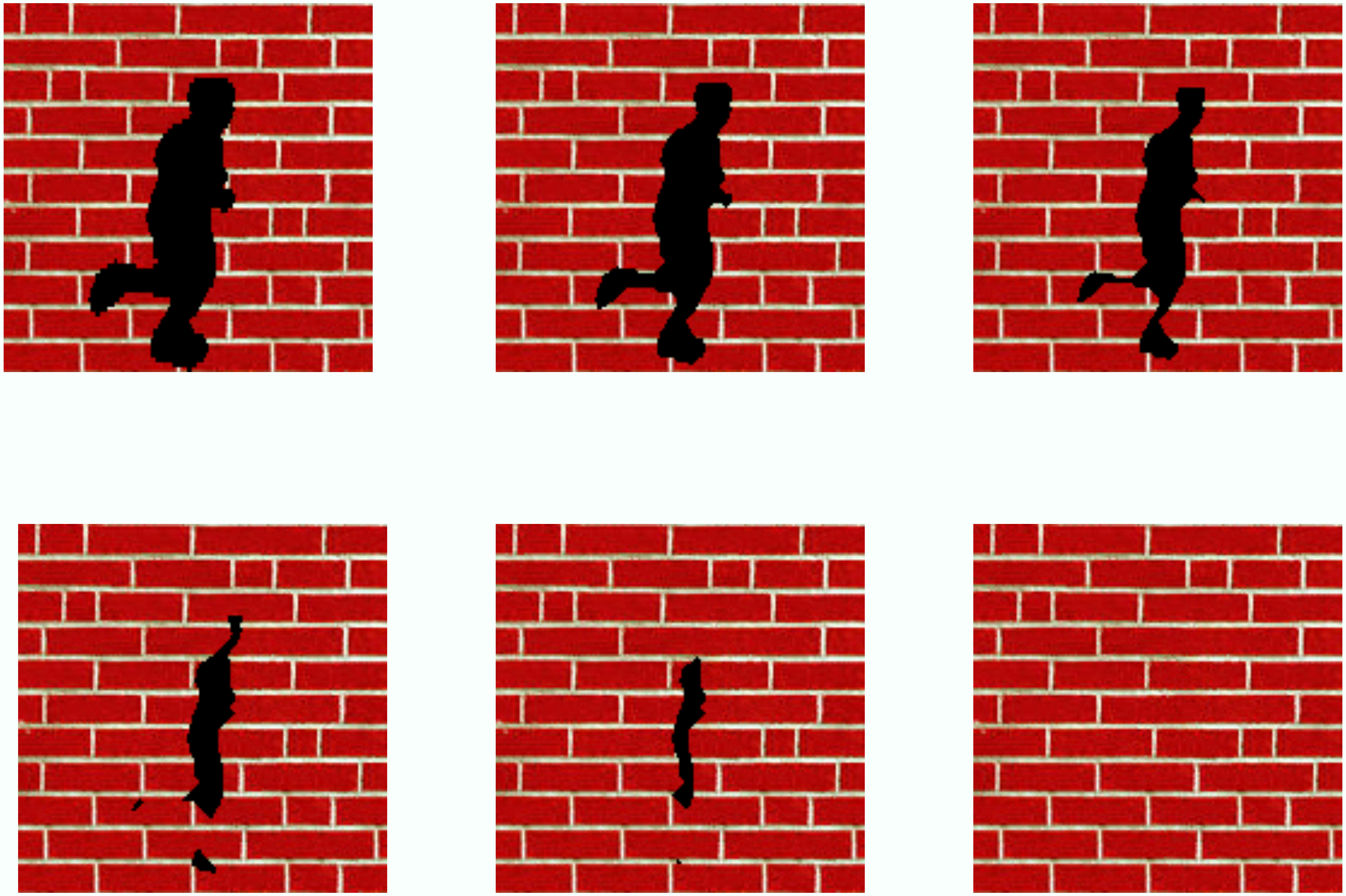
Efros & Leung ICCV99

Growing Regions
Hole Filling



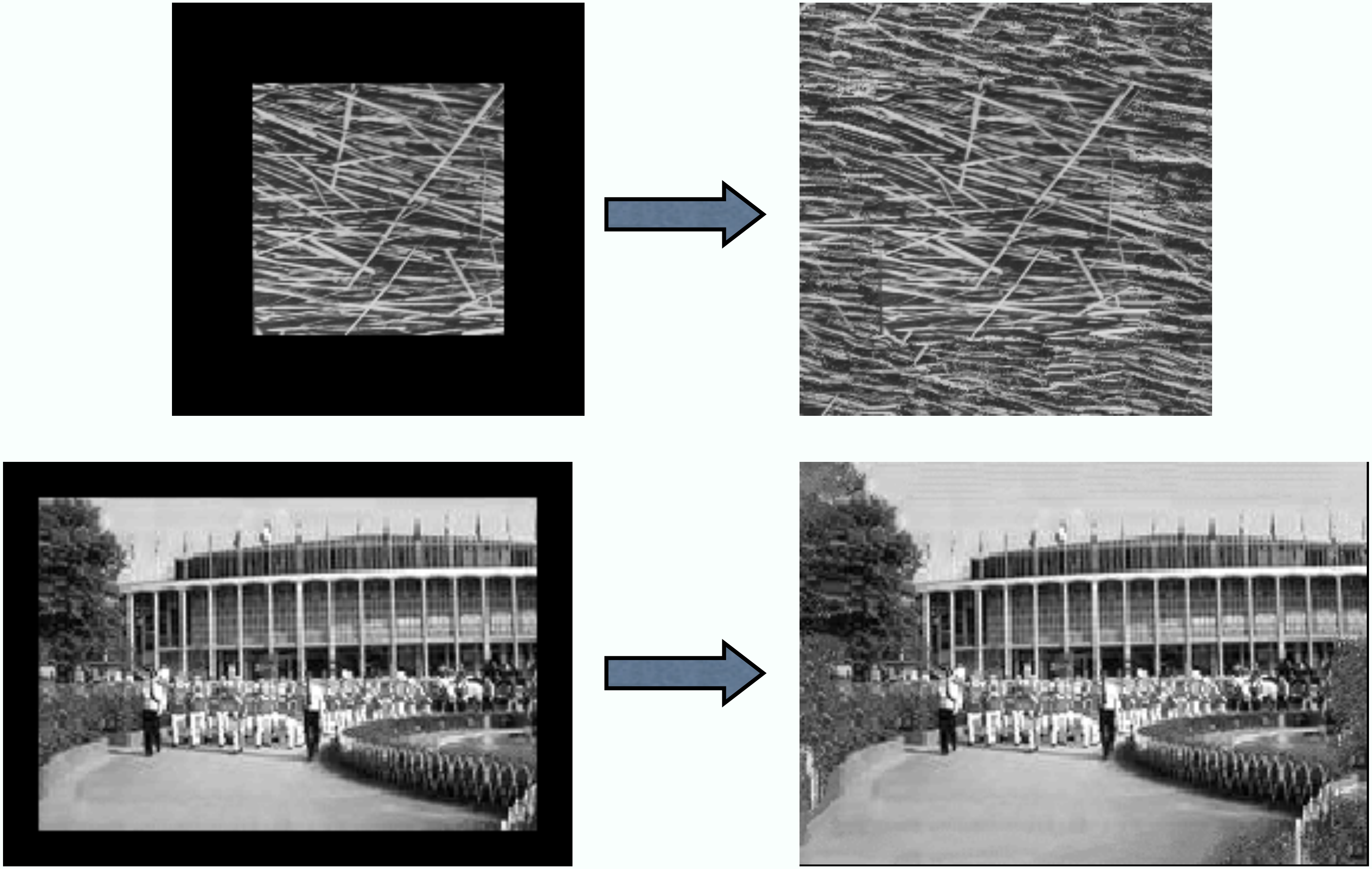
Efros & Leung ICCV99

Hole Filling



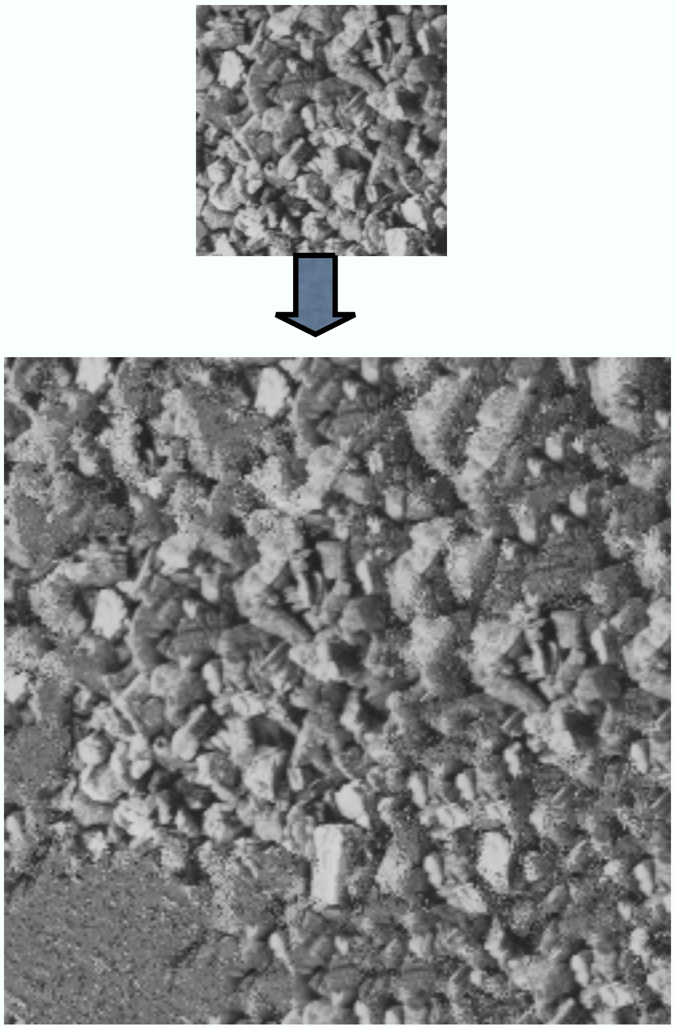
Efros & Leung ICCV99

Extrapolation

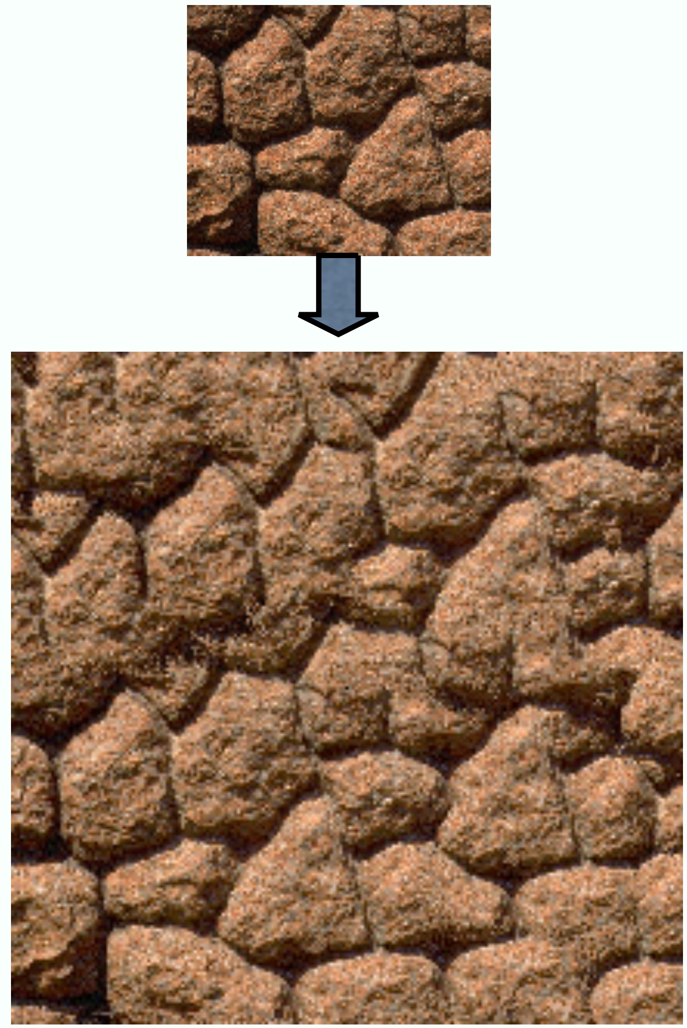


Efros & Leung ICCV99

Failure Cases



Growing garbage



Verbatim copying

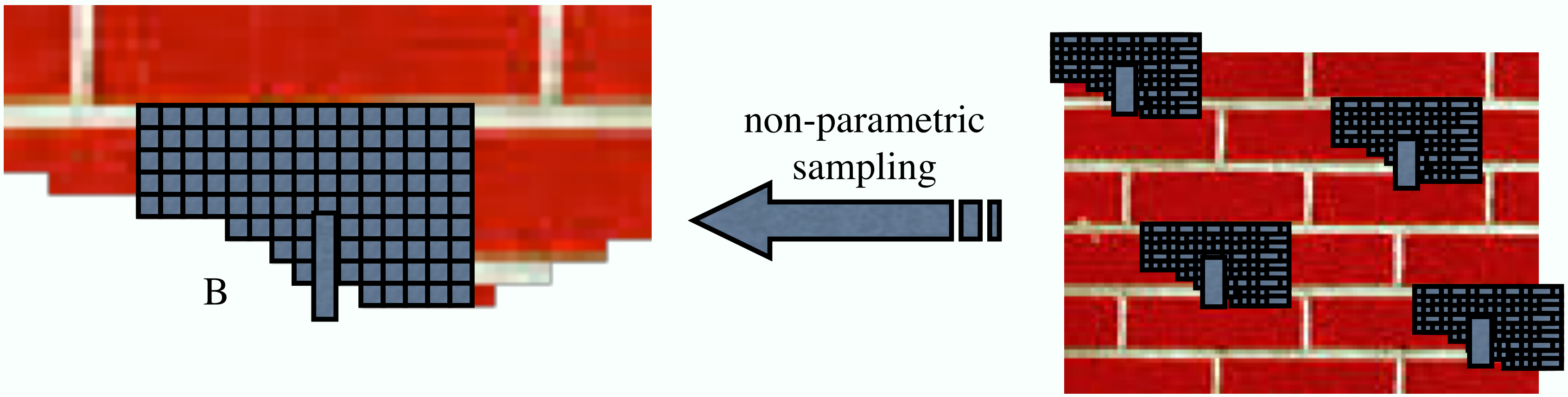
Efros & Leung ICCV99

Pros and Cons

- Very simple
- Easy to implement
- Promising results

- Very slooooooowwwwwww
- Idea:
  - Patches instead of pixels

Patch based

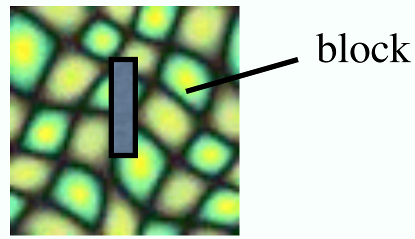


Synthesizing a block

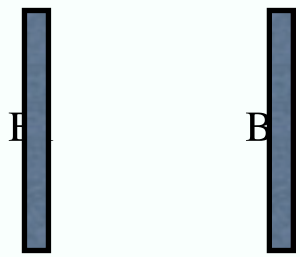
Input image

- Observation
  - neighbouring pixels are highly correlated
- Idea:
  - unit of synthesis = block

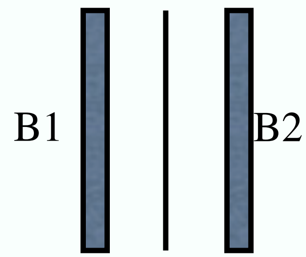
Efros & Freeman SIGGRAPH01



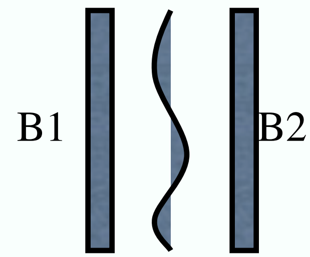
Input texture



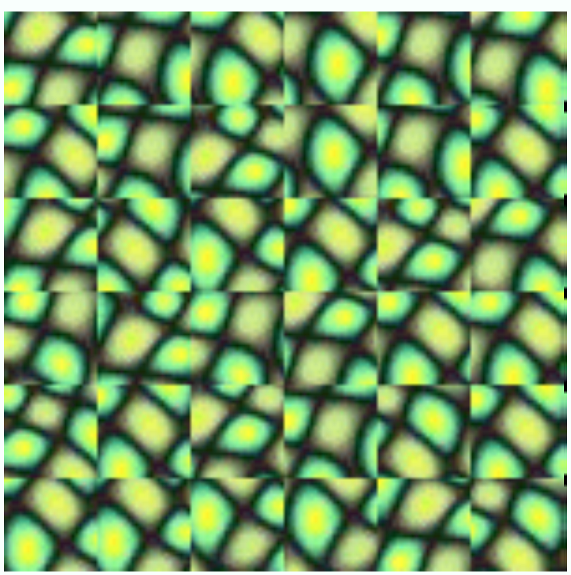
Random placement of blocks

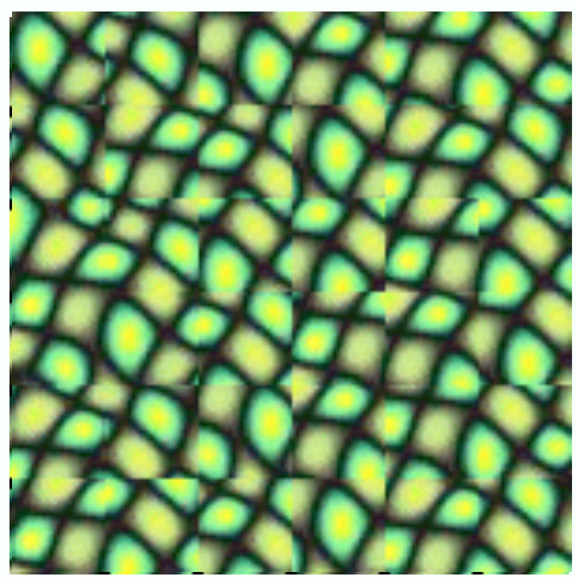


Neighboring blocks constrained by overlap



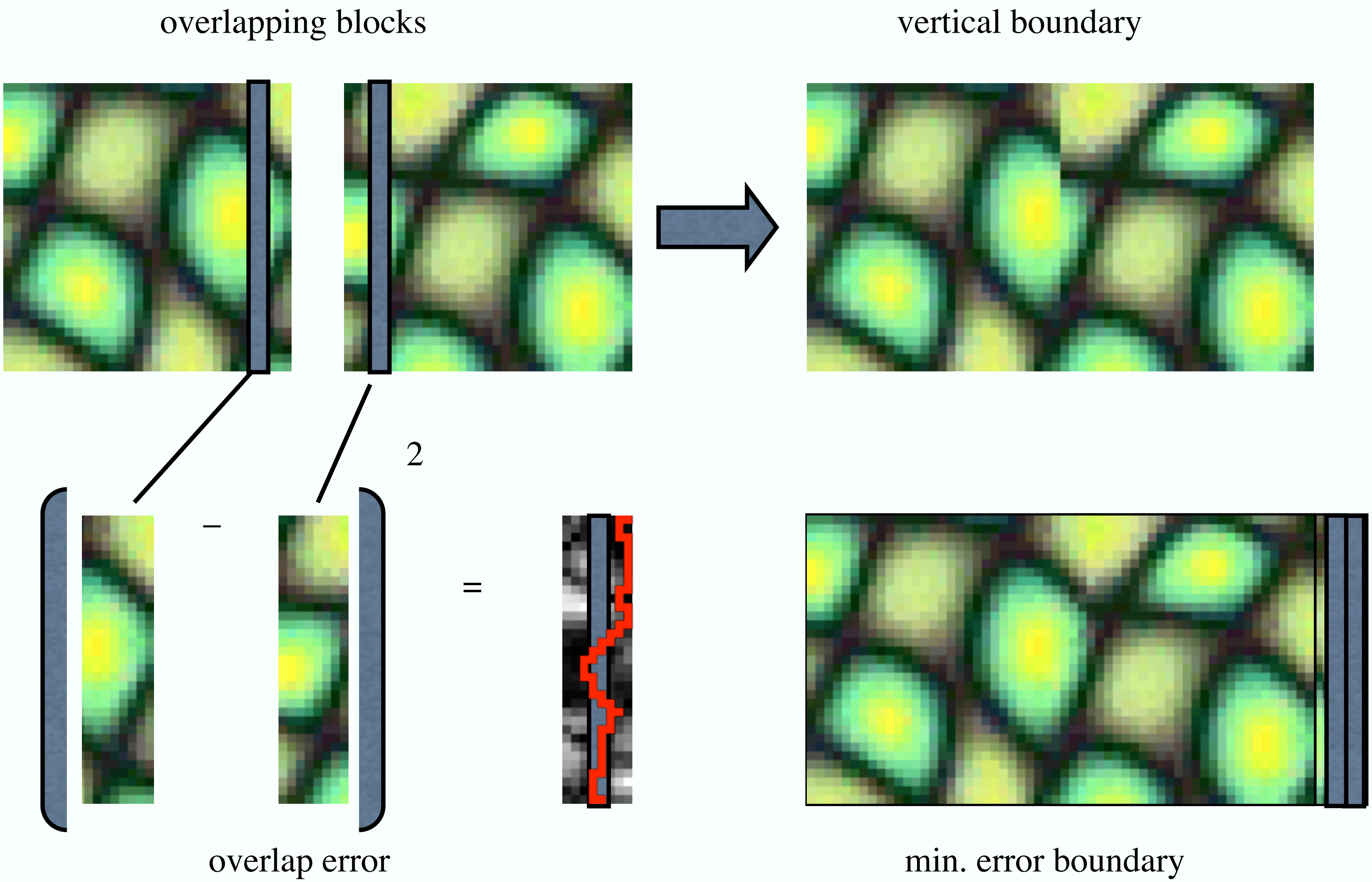
Minimal error boundary cut





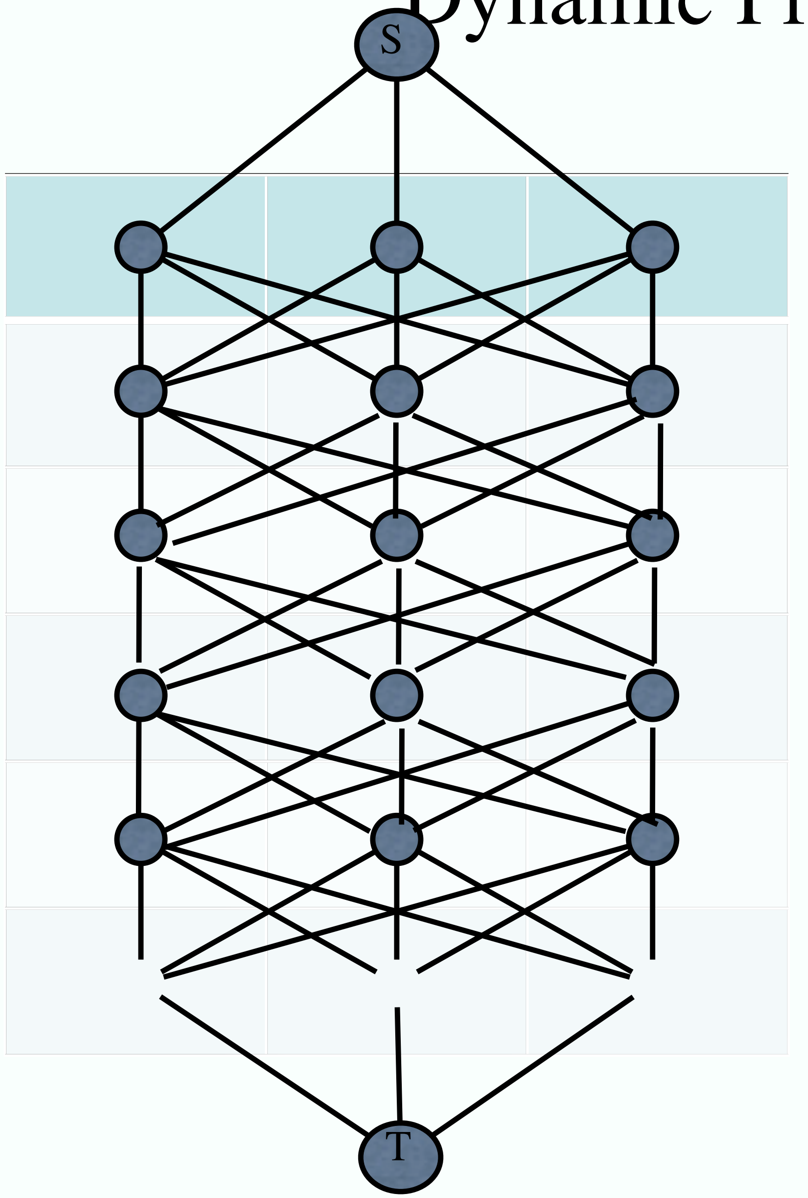
Efros & Freeman SIGGRAPH01

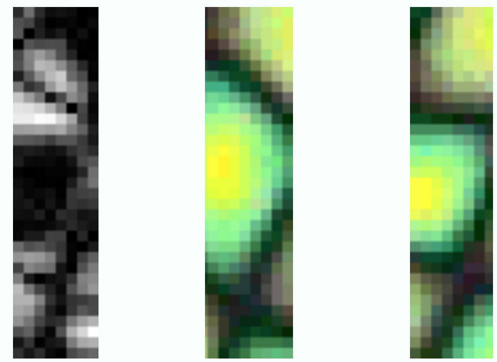
Minimal error boundary



Efros & Freeman SIGGRAPH01

Dynamic Programming

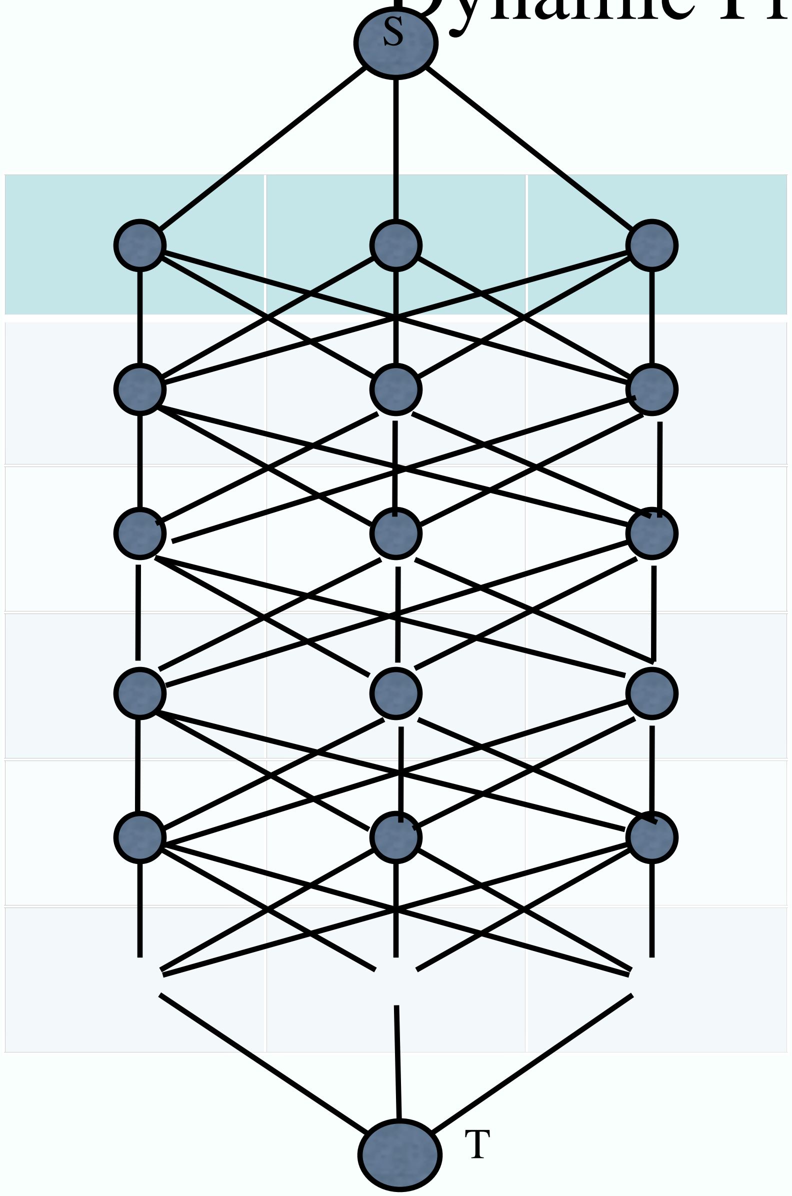


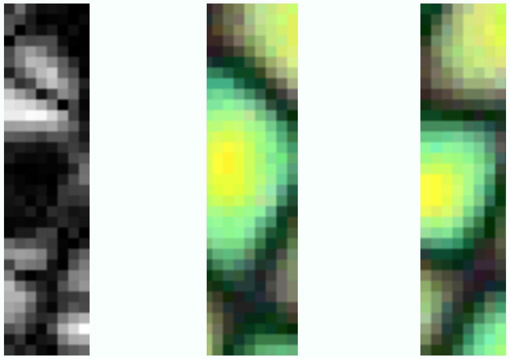


$$e_{ij} = (B1_{ij}^{ov} - B2_{ij}^{ov})^2$$

$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

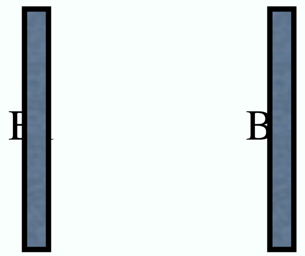
Dynamic Programming



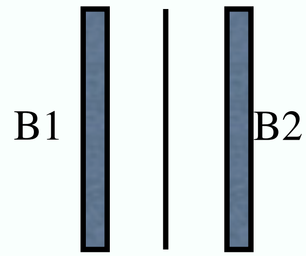


$$e_{ij} = (B1_{ij}^{ov} - B2_{ij}^{ov})^2$$

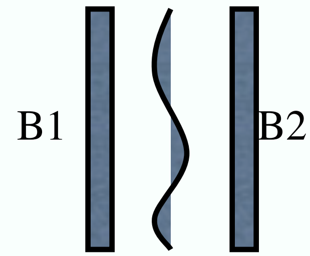
$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$



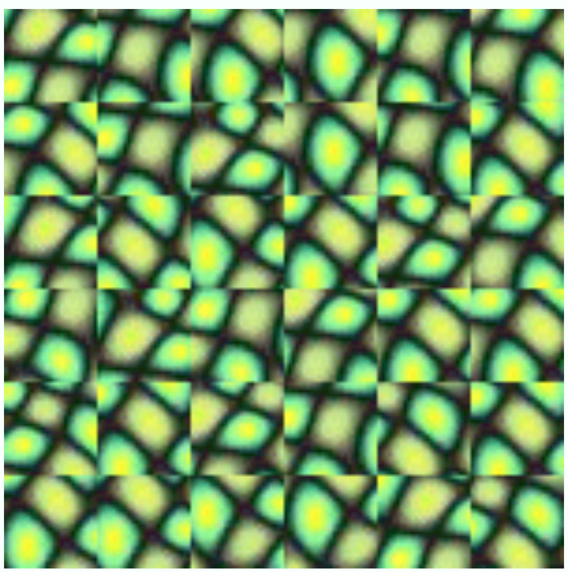
Random placement
of blocks

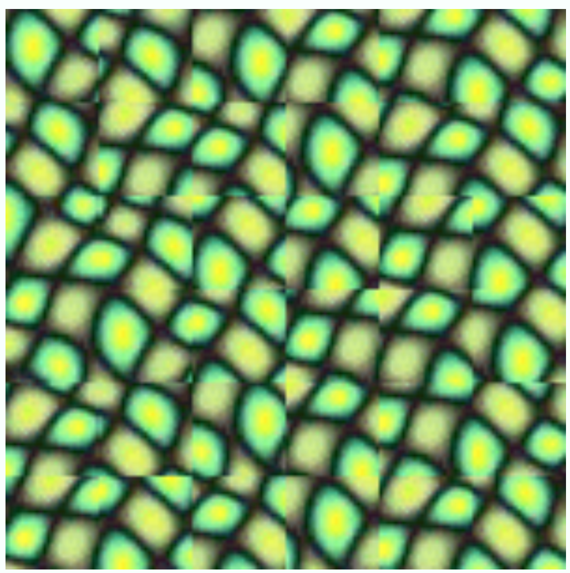


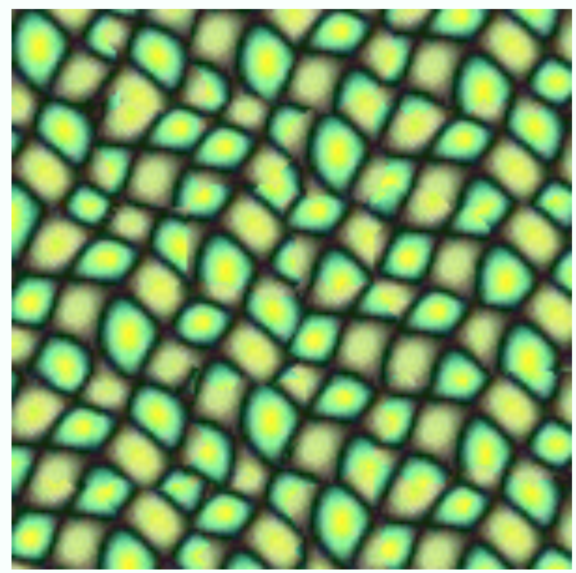
Neighboring blocks
constrained by overlap



Minimal error
boundary cut

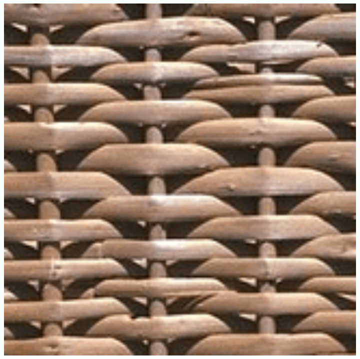


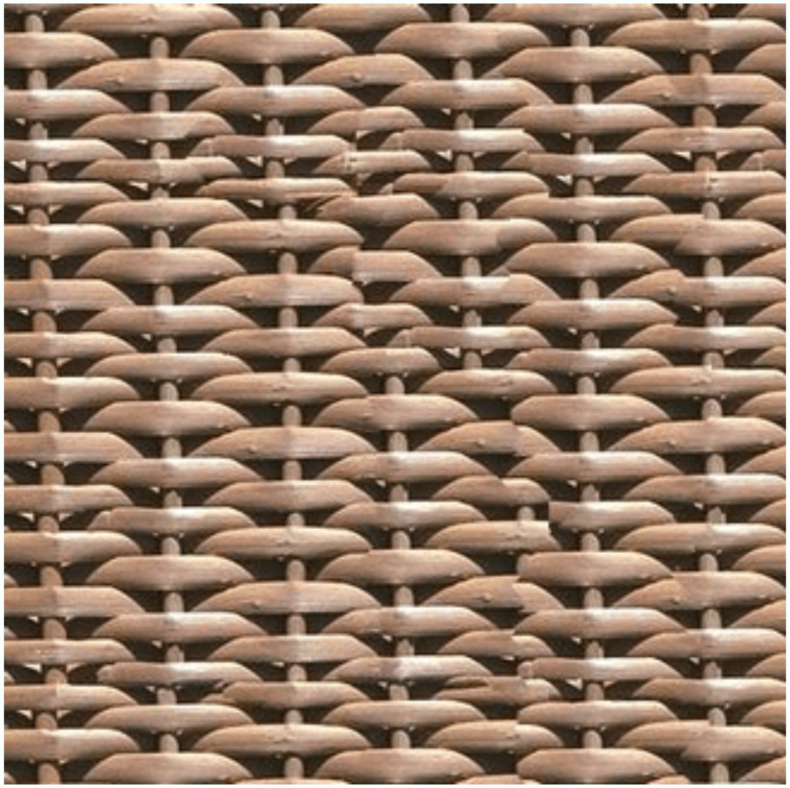




Efros & Freeman SIGGRAPH01

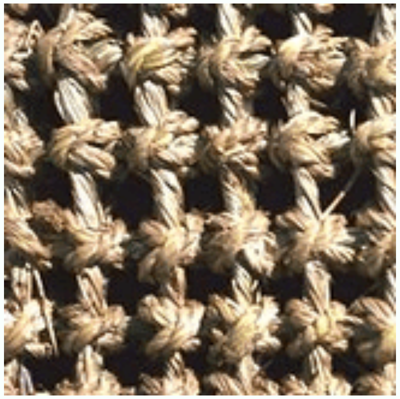
More Results

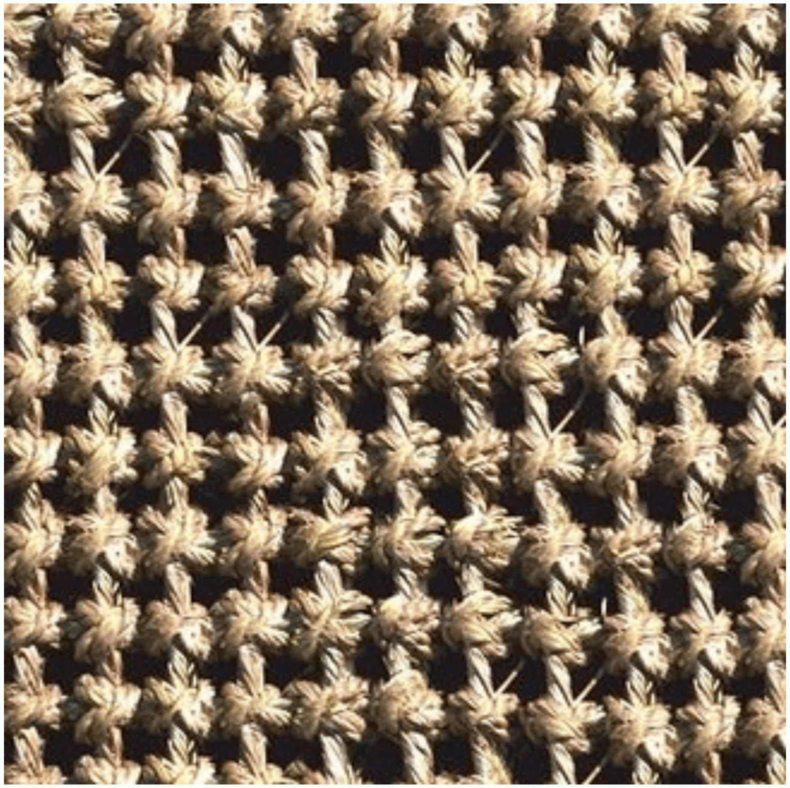


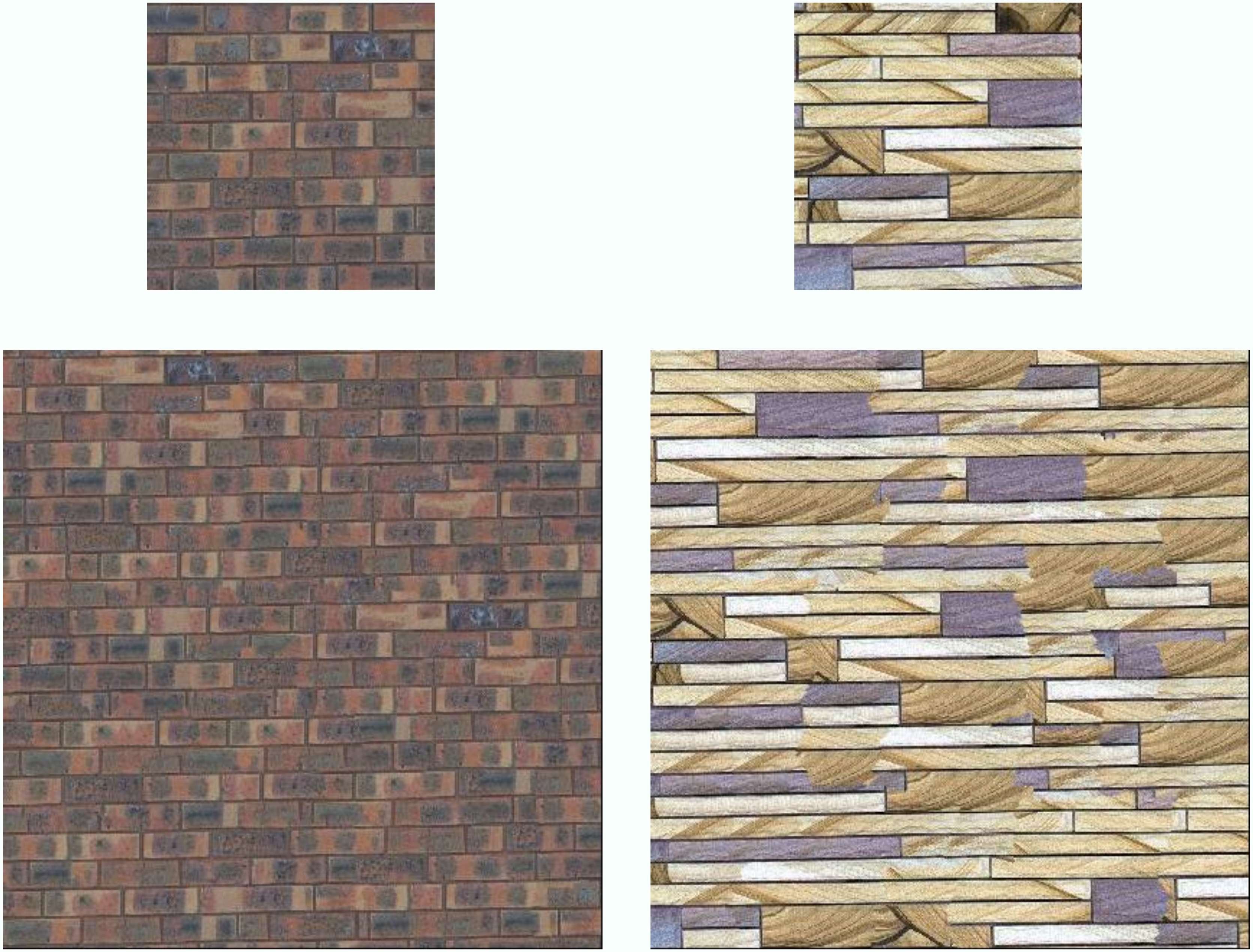


Efros & Freeman SIGGRAPH01

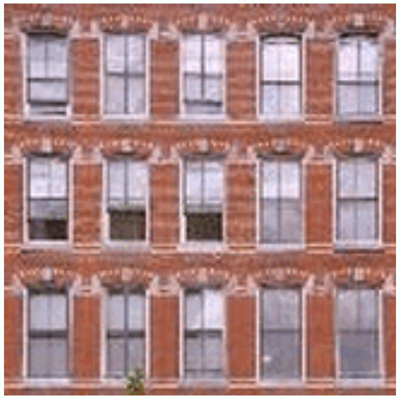
More Results







Efros & Freeman SIGGRAPH01




Efros & Freeman SIGGRAPH01




Efros & Freeman SIGGRAPH01

Efros & Freeman SIGGRAPH01

Failures

Efros & Freeman SIGGRAPH01

Texture Transfer

- Take the texture from on object and paint it on another object

+

=

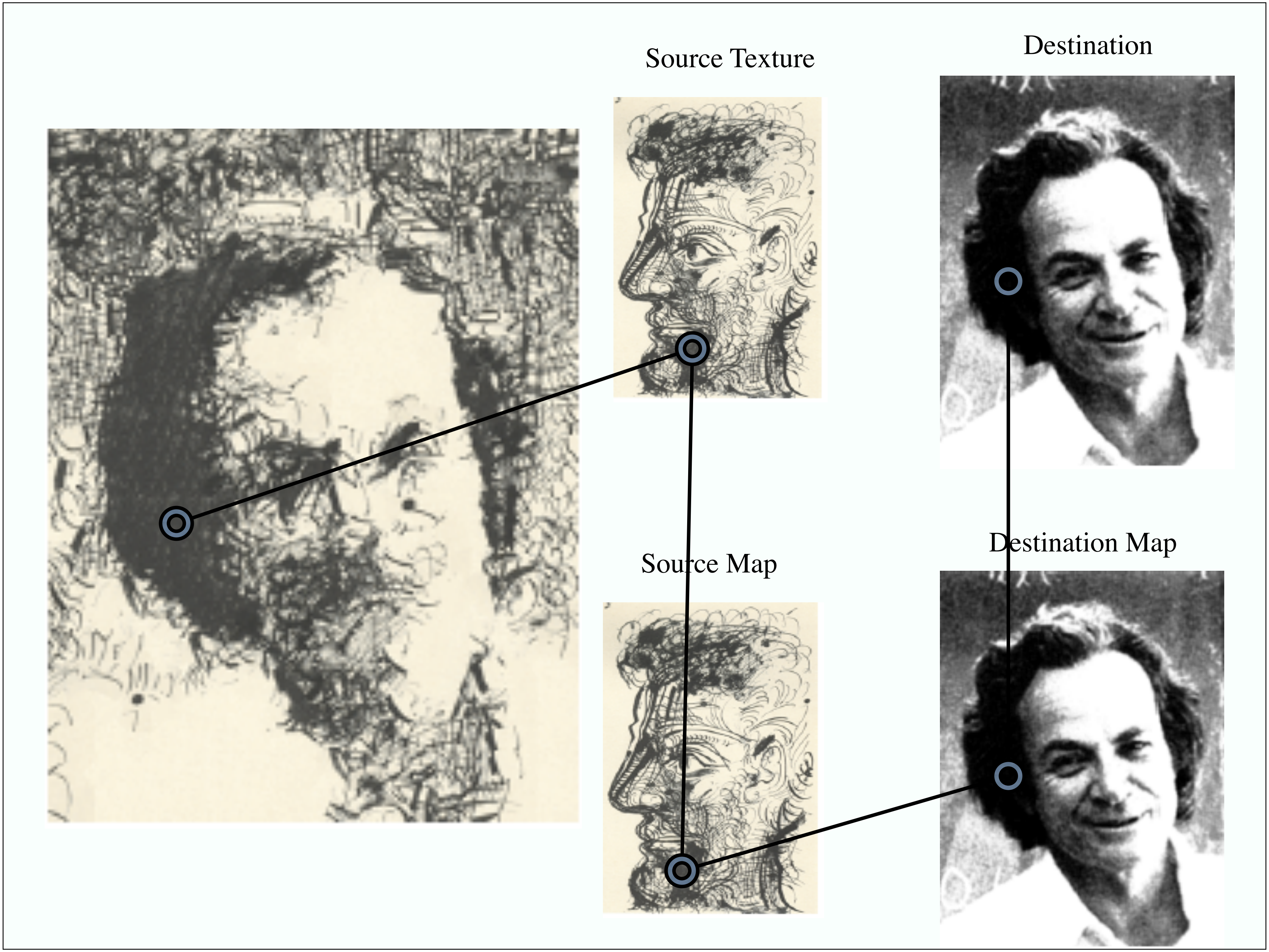
Decomposing shape and texture

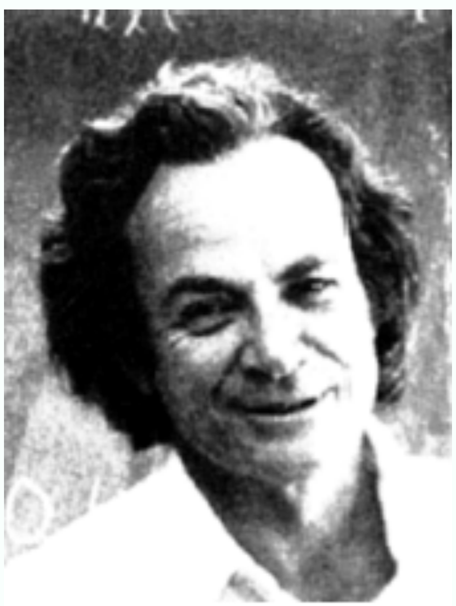
Very challenging

Walk around

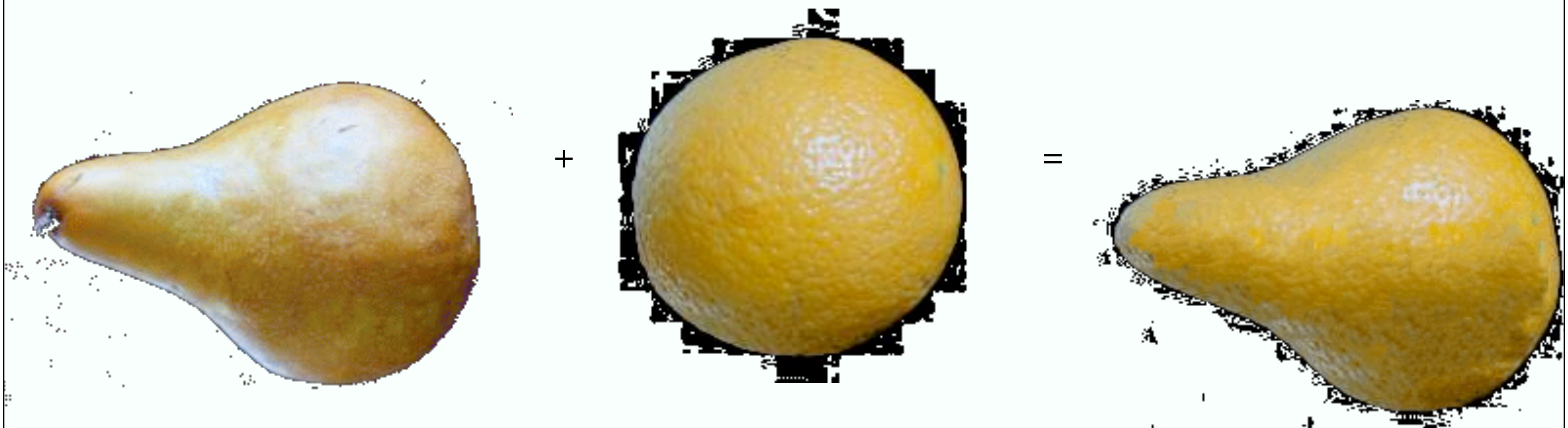
Add some constraint to the search

Efros & Freeman SIGGRAPH01


Source Texture
Destination
Source Map
Destination Map

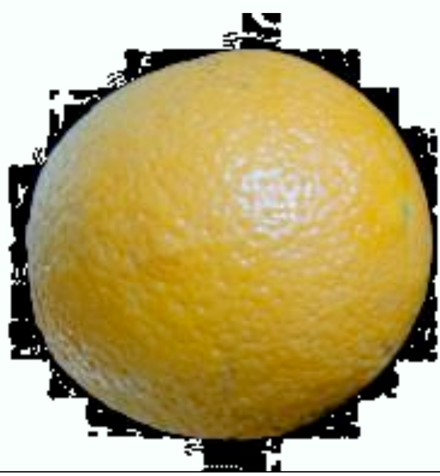


Texture Transfer



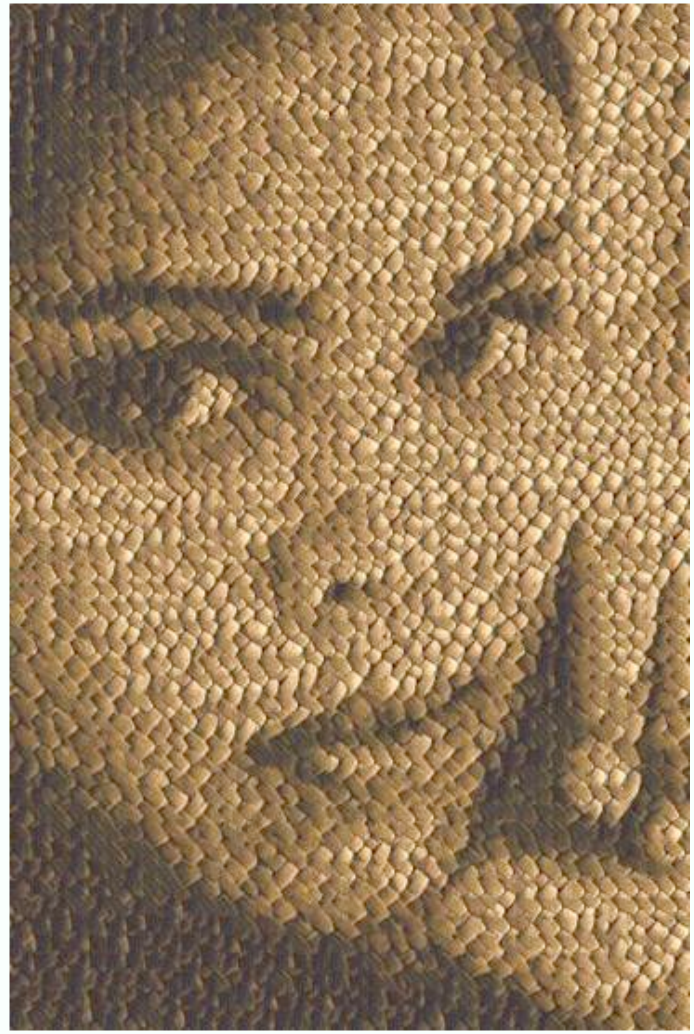
Efros & Freeman SIGGRAPH01

+



=

Efros & Freeman SIGGRAPH01



Efros & Freeman SIGGRAPH01

parmesan

+

=

rice

+

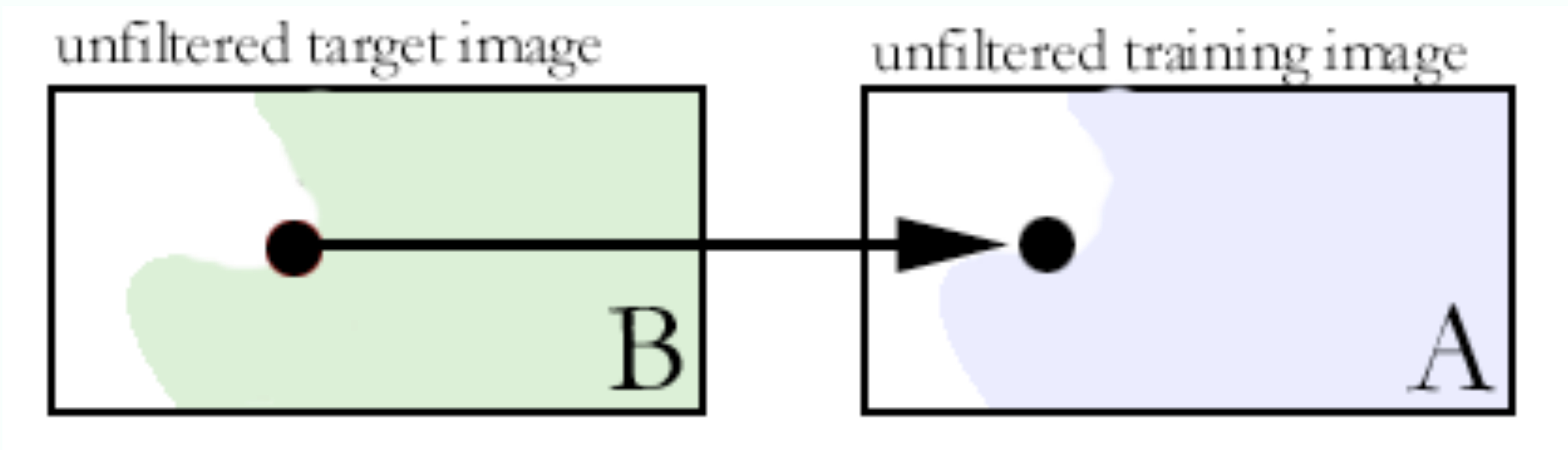
=

Efros & Freeman SIGGRAPH01

Image Analogies

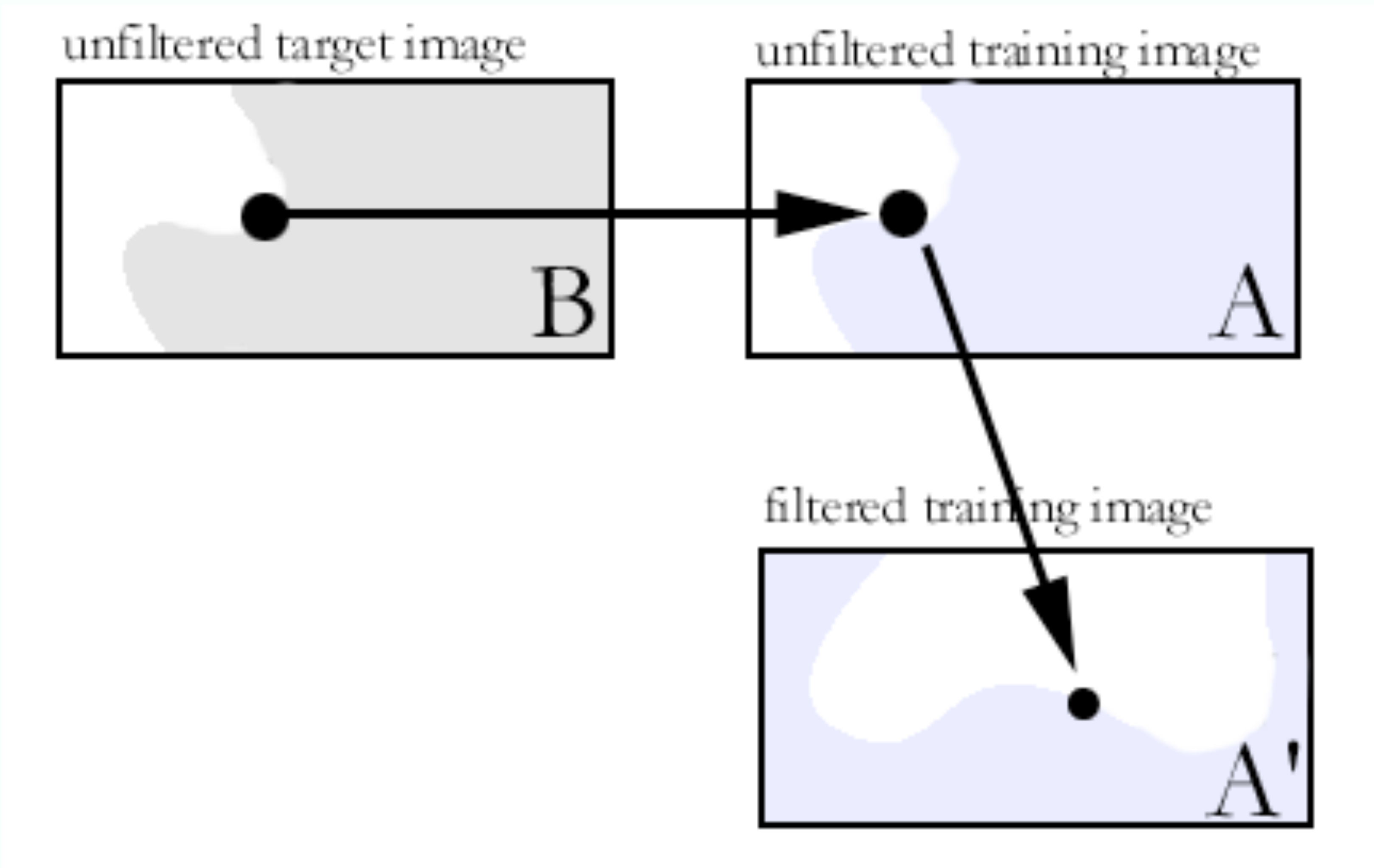
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Image Analogies



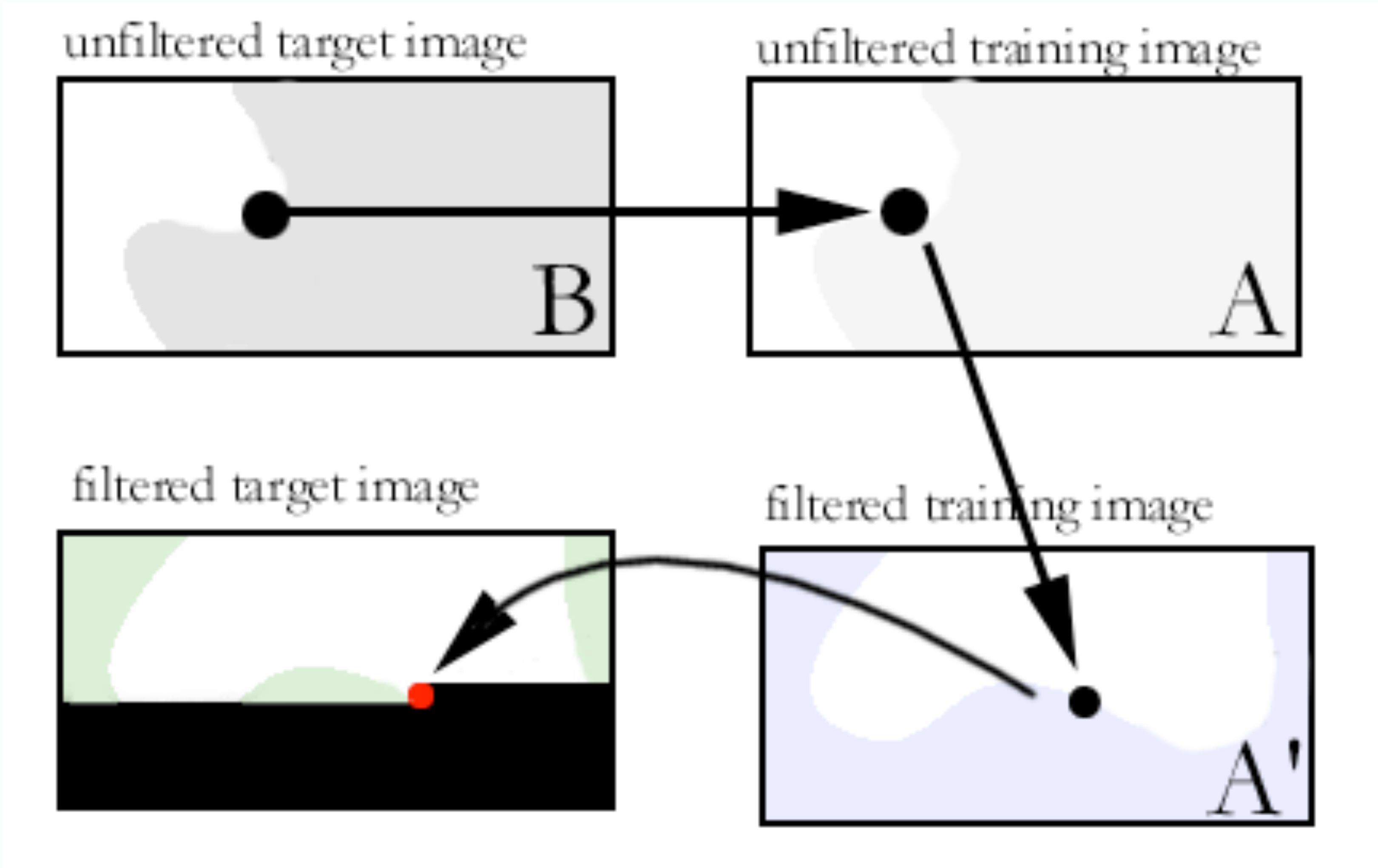
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Image Analogies



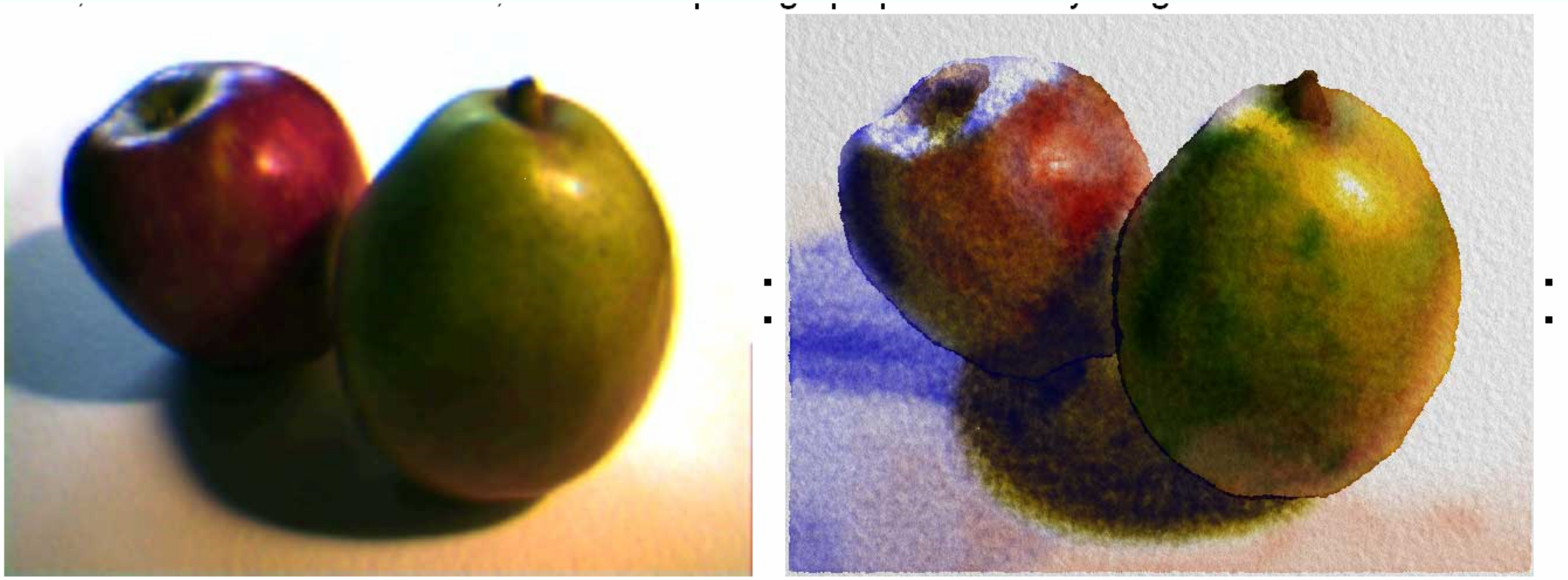
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Image Analogies



Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Training



Unfiltered source (A) Filtered source (A')

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

::

B

:

B’

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

::

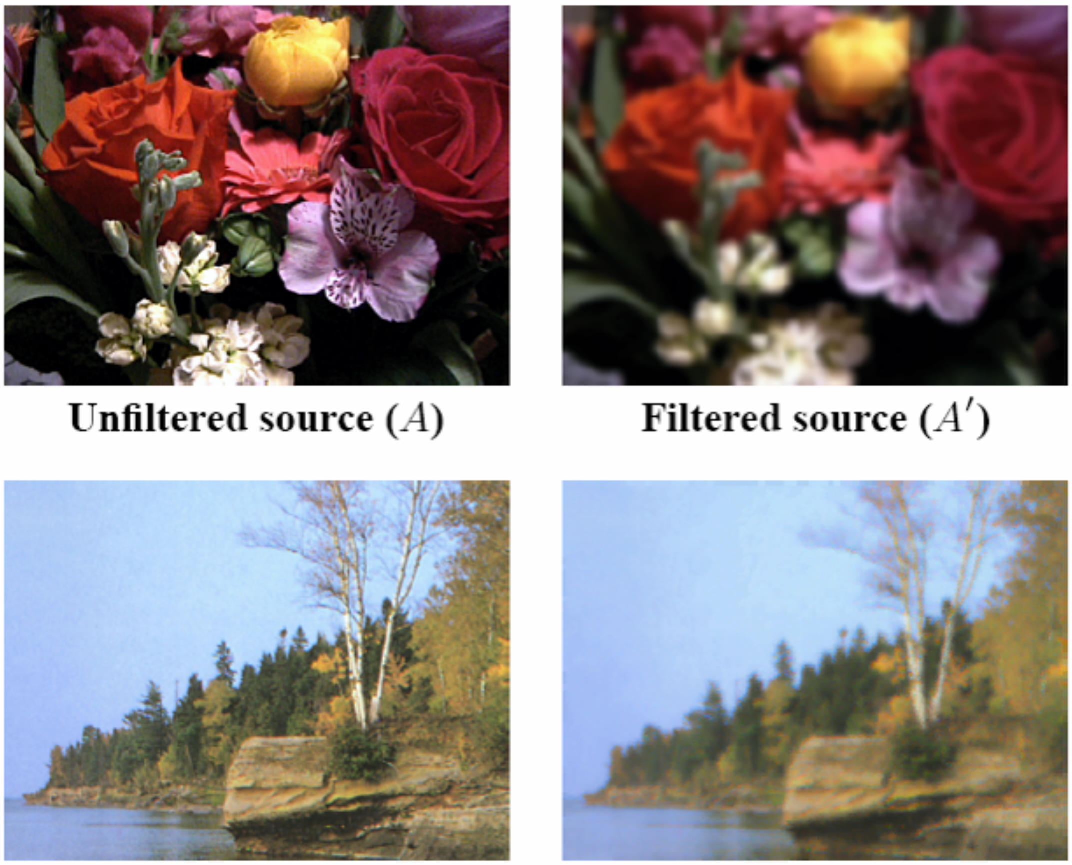
:

B

B’

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Learn to Blur

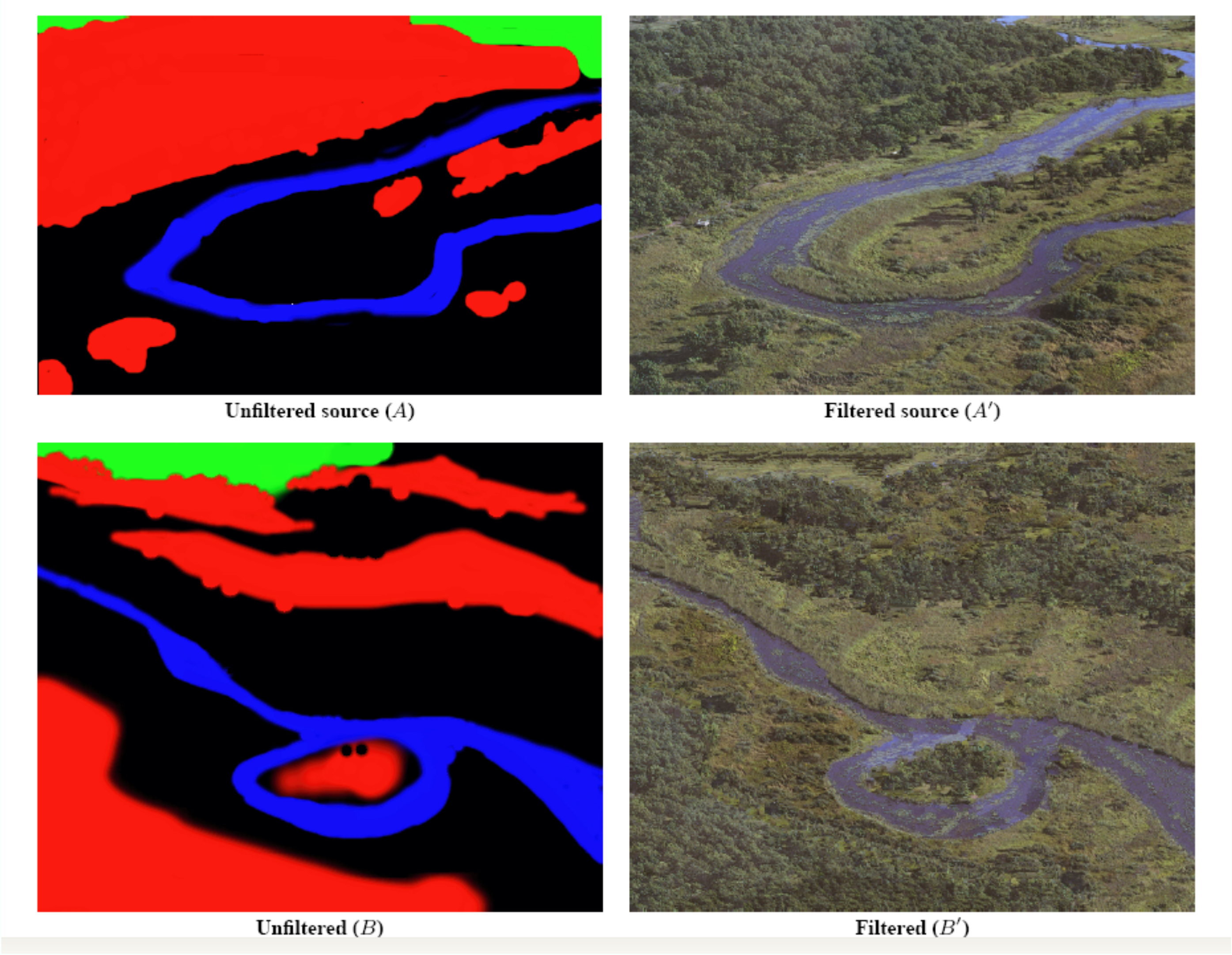


Unfiltered source (A) Filtered source (A')

Unfiltered target (B) Filtered target (B')

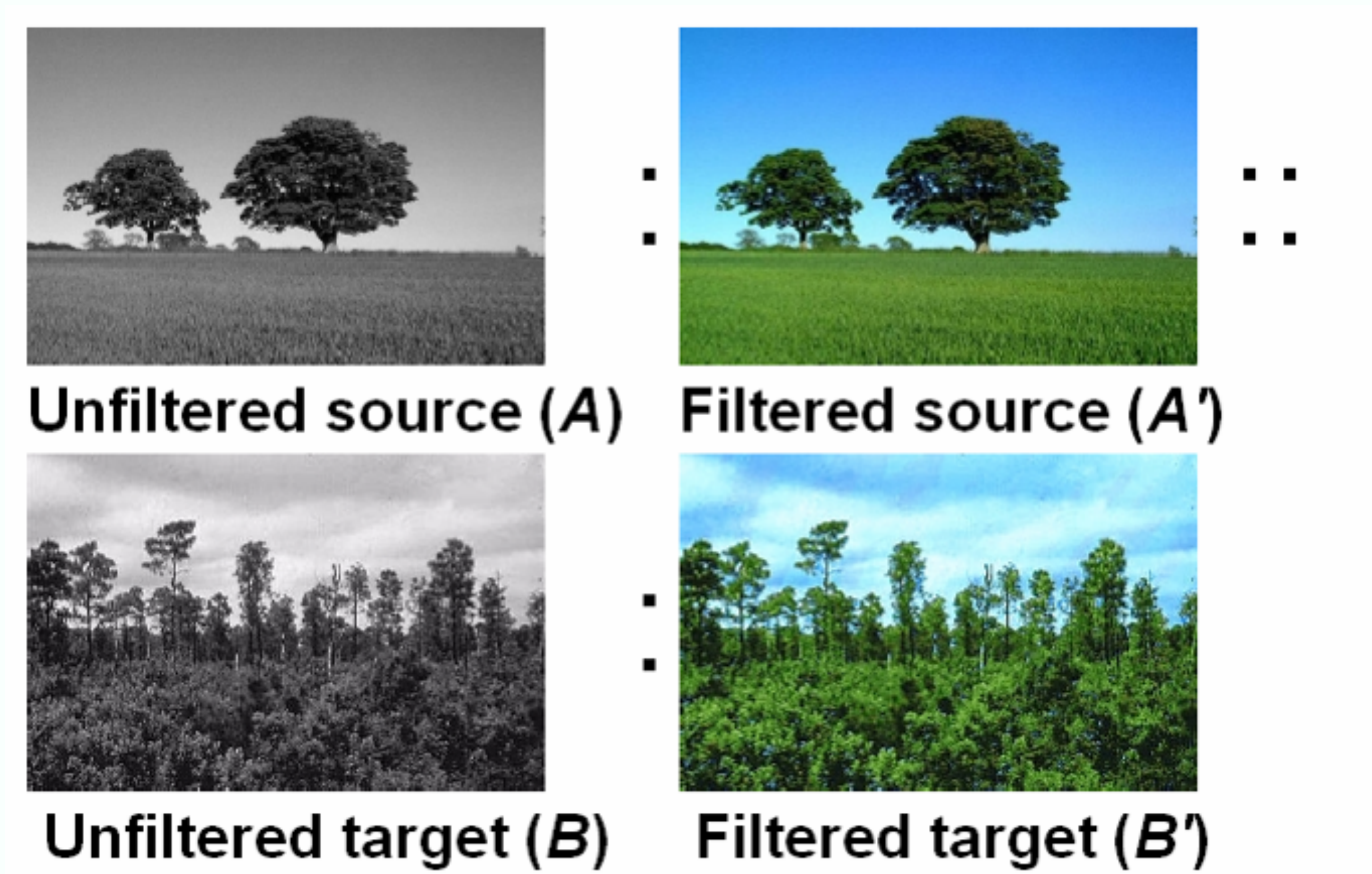
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Texture by Numbers



Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Colorization



Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Super-resolution

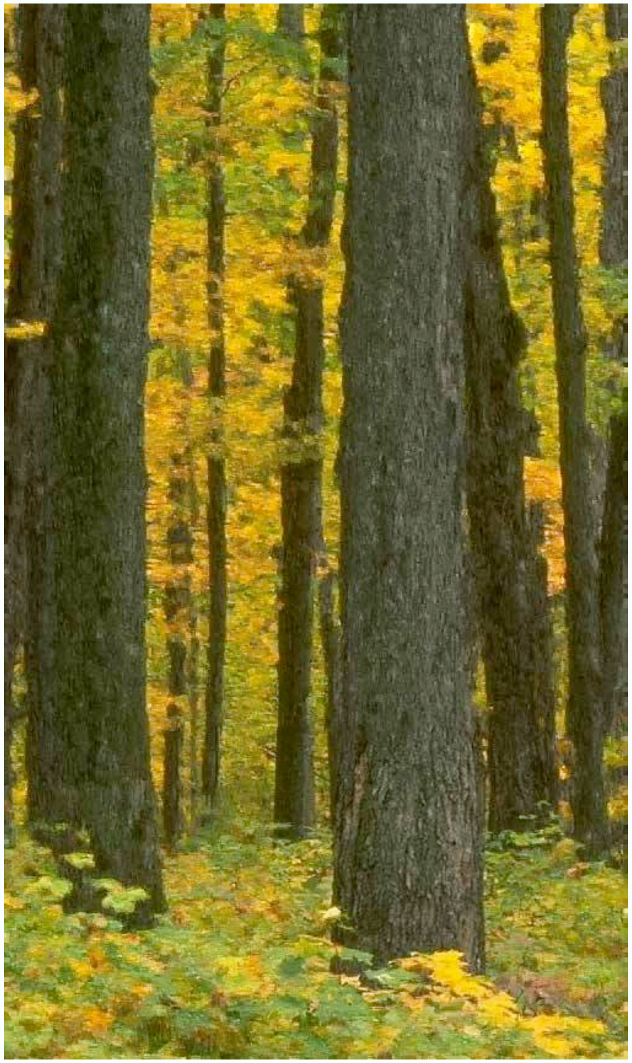
A

A’

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Super-resolution (result!)

B



B’

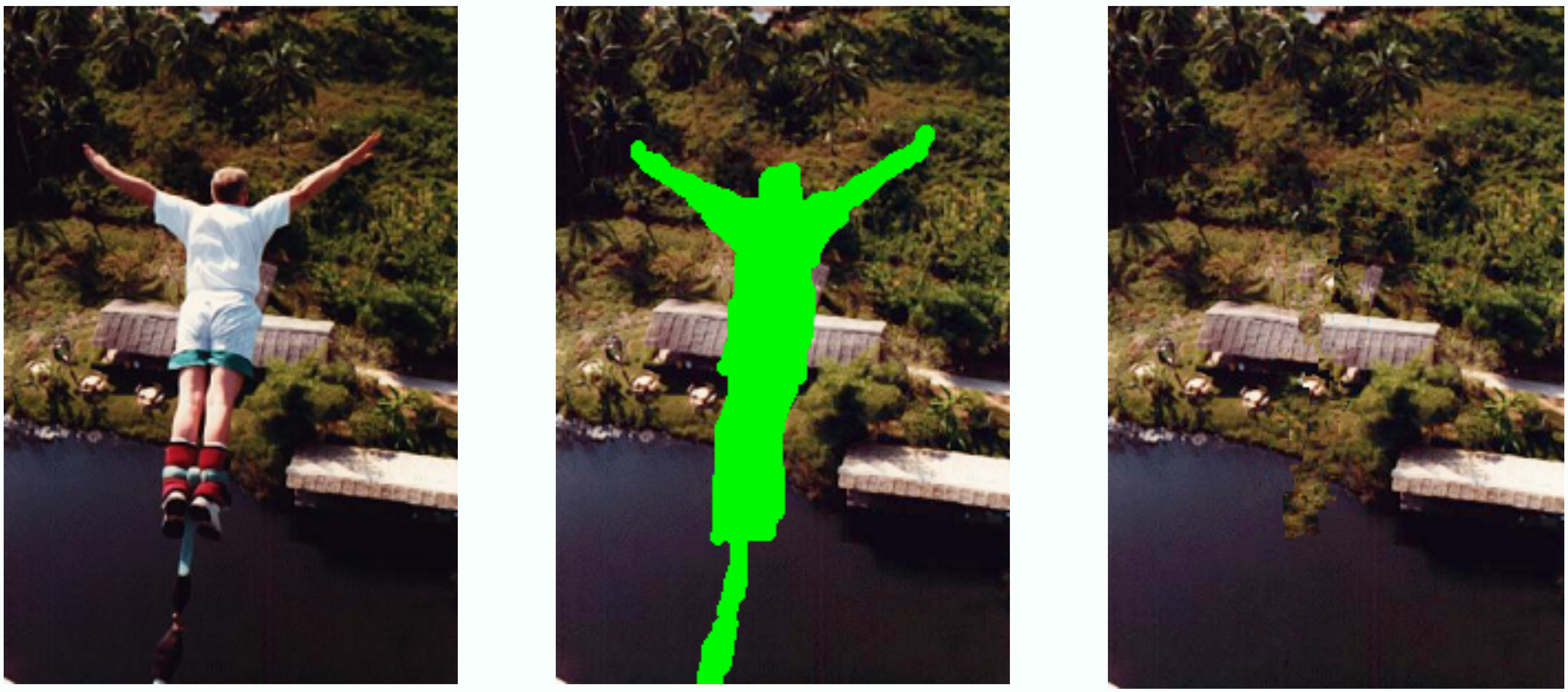
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Training images

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

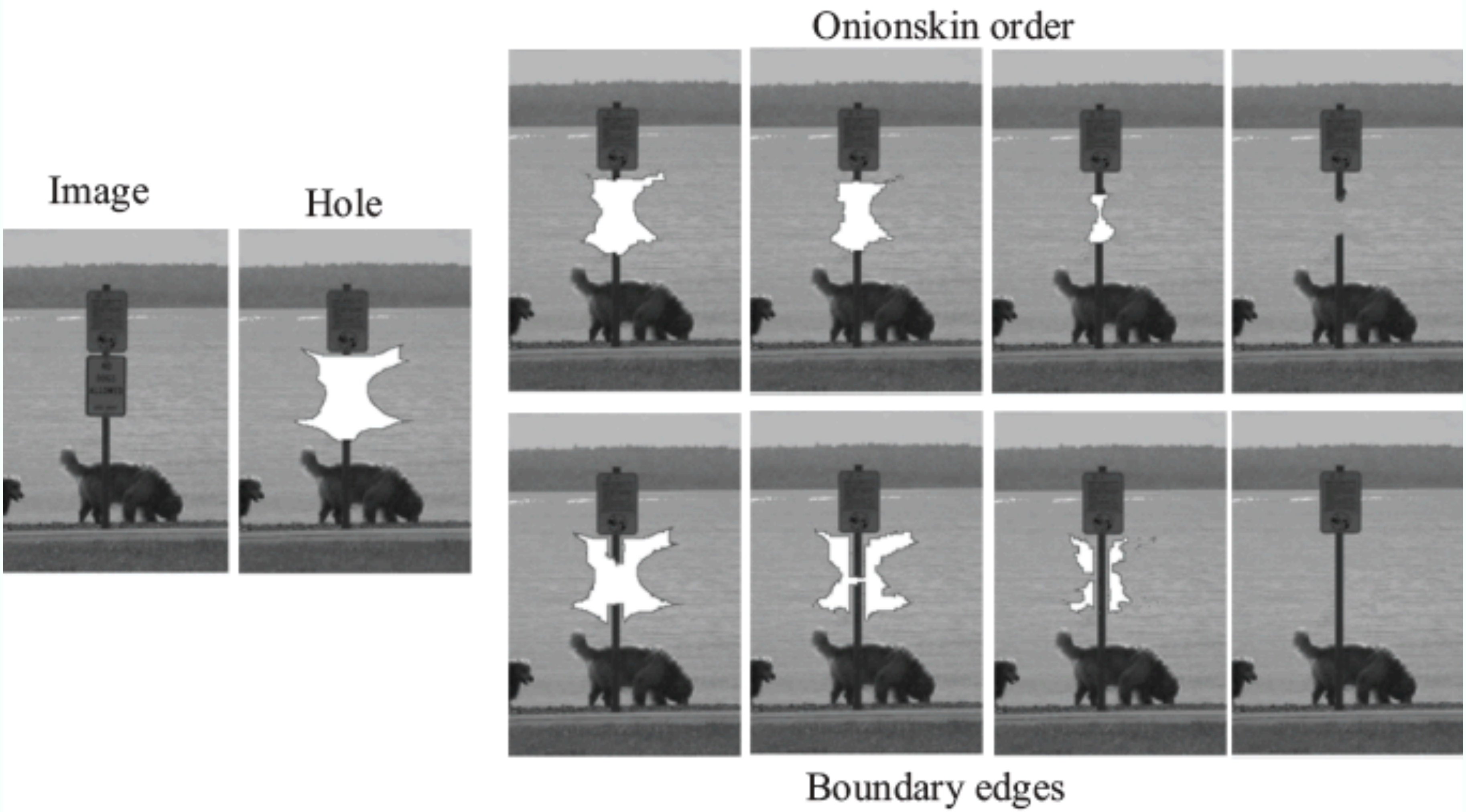
Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01

Inpainting



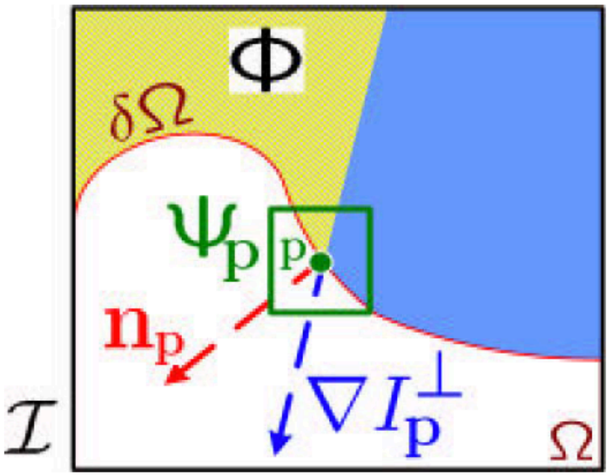
Criminisi et.al. CVPR03

Order of inpainting matters



Criminisi et al, 04

Choosing the order



Criminisi et al 03

Given a patch $\Psi_{\mathbf{p}}$ centred at the point $\mathbf{p}$ for some $\mathbf{p} \in \delta\Omega$ (see fig. 3), its priority $P(\mathbf{p})$ is defined as the product of two terms:

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}). \tag{1}$$

We call $C(\mathbf{p})$ the *confidence* term and $D(\mathbf{p})$ the *data* term, and they are defined as follows:

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \bar{\Omega}} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|}, \qquad D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha}$$

where $|\Psi_{\mathbf{p}}|$ is the area of $\Psi_{\mathbf{p}}$, α is a normalization factor (*e.g.*, $\alpha = 255$ for a typical grey-level image), and $\mathbf{n}_{\mathbf{p}}$ is a unit vector orthogonal to the front $\delta\Omega$ in the point $\mathbf{p}$. The

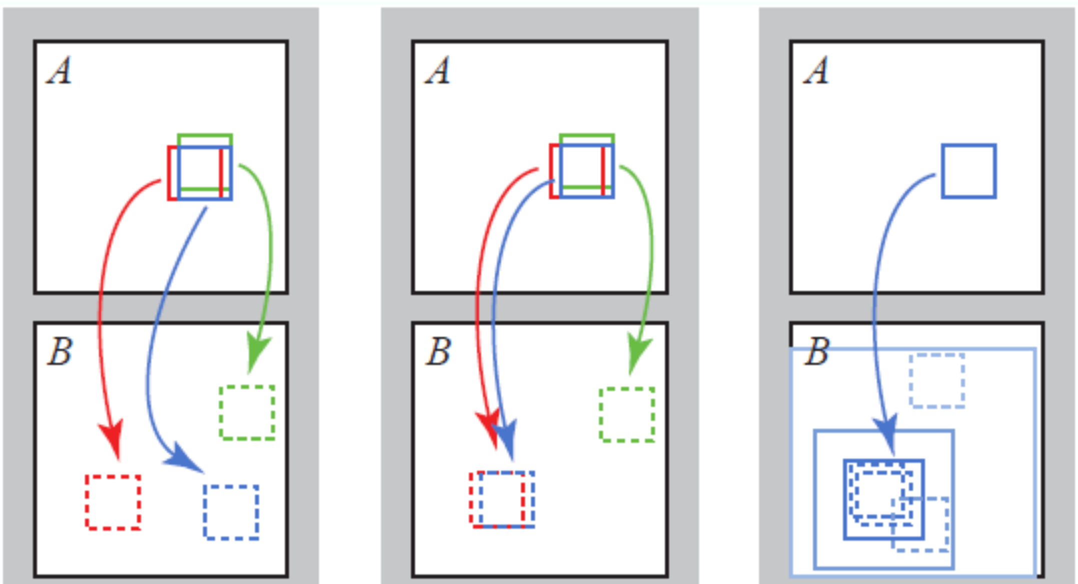
Constraining the match region

- We don’t have to look for matches in the whole image
  - idea: allow user to “paint” good sources of matches on top of the image

Nearest Neighbor search

The core of most of the patch based methods
Very slow

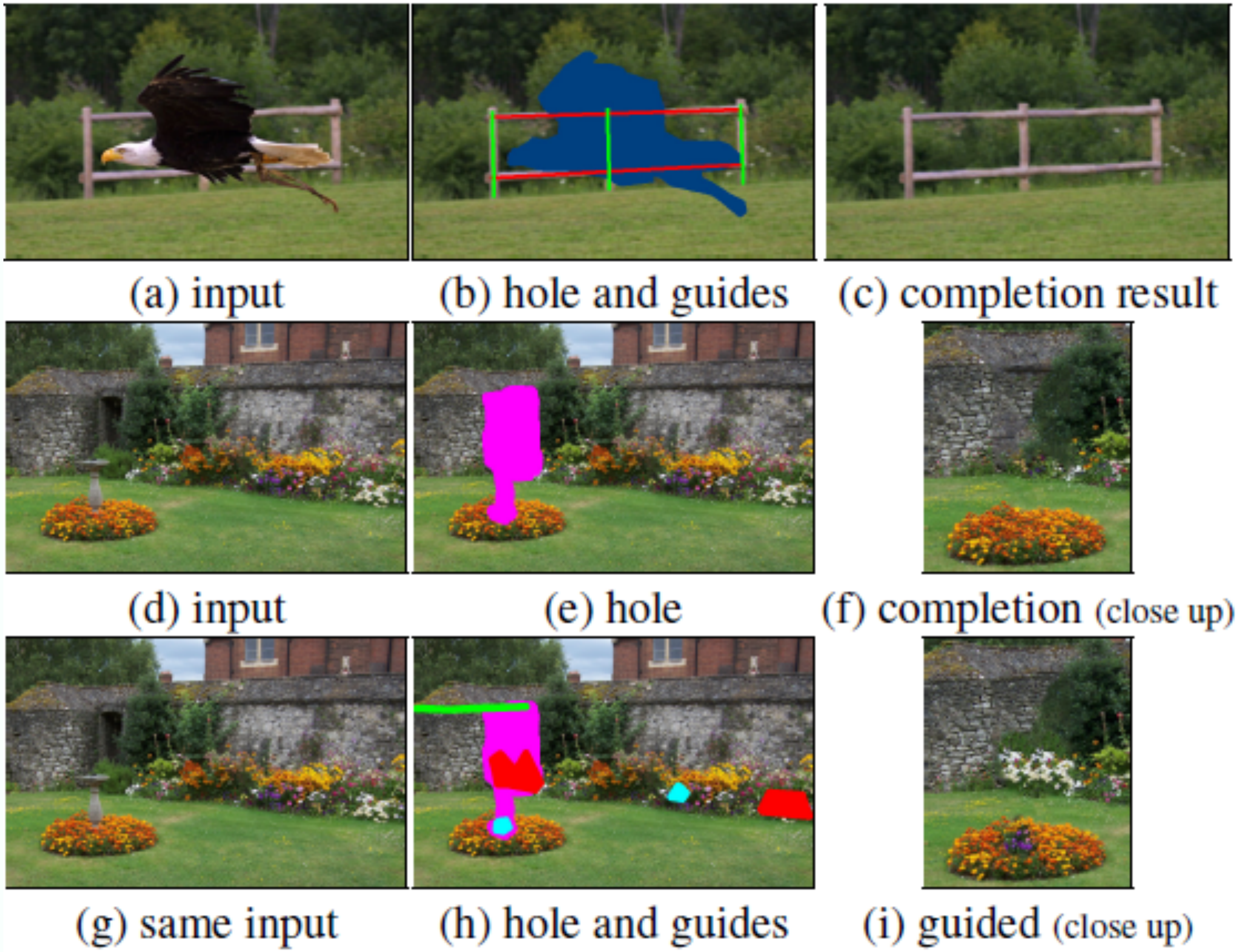
Smarter neighborhood search



(a) Initialization (b) Propagation (c) Search

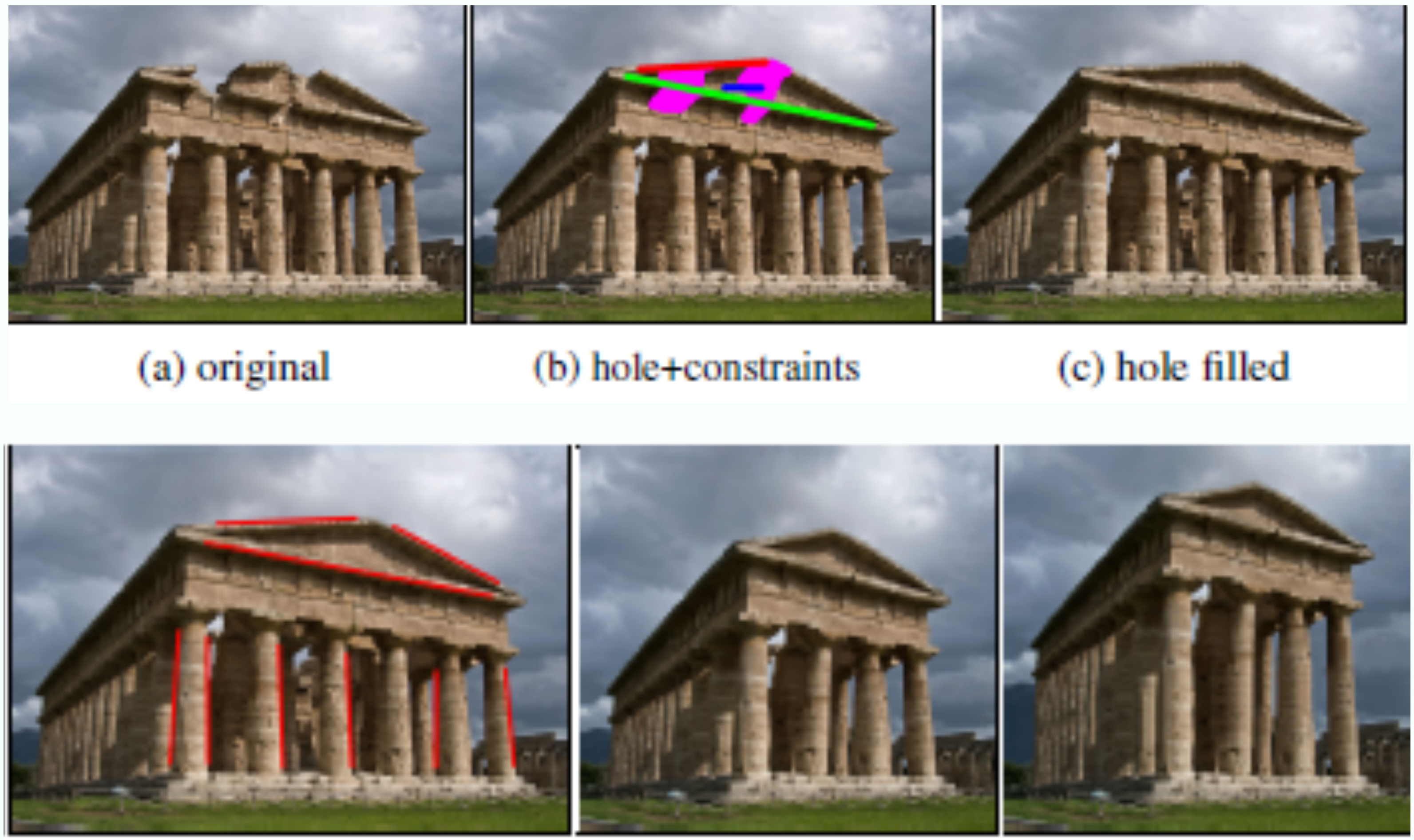
Barnes et.al. SIGGRAPH09

Inpainting


(a) input (b) hole and guides (c) completion result
(d) input (e) hole (f) completion (close up)
(g) same input (h) hole and guides (i) guided (close up)

Barnes et.al. SIGGRAPH09

Applications



(a) original (b) hole+constraints (c) hole filled

(d) constraints (e) constrained retarget (f) reshuffle

Barnes et.al. SIGGRAPH09

Retargeting

- Make an image bigger or smaller in one direction
  - eg change aspect ratio
- Traditional
  - cut off pixels
  - difficulty: lousy results
- Strategy
  - cut out a curve of pixels that “doesn’t matter much”
    - low energy at pixels
    - many energy functions, eg

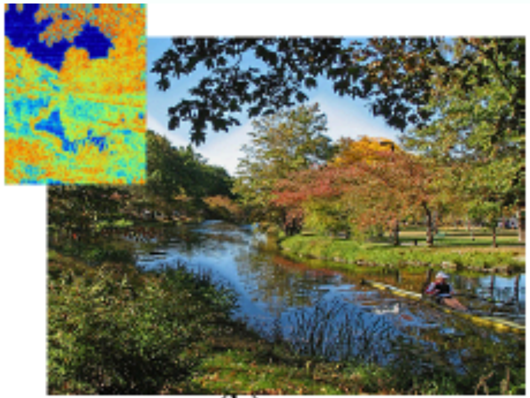
$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

Finding a seam=DP

Avidan, Shamir, SIGGRAPH07

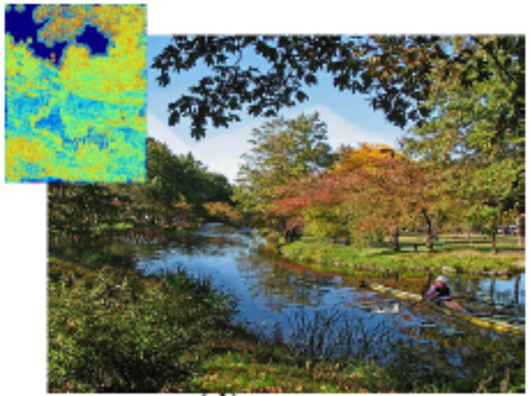
- Different energies give different results
  - e1 = abs gradient (as above)
  - ehog = (look for gradients in patch)
  - eentropy = (entropy of patch)
  - eseg = (segment image, e1 in segments, 0 on boundaries)

(a) Original

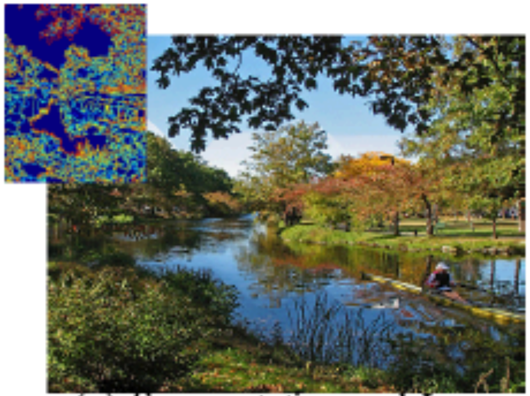


(b) e_1

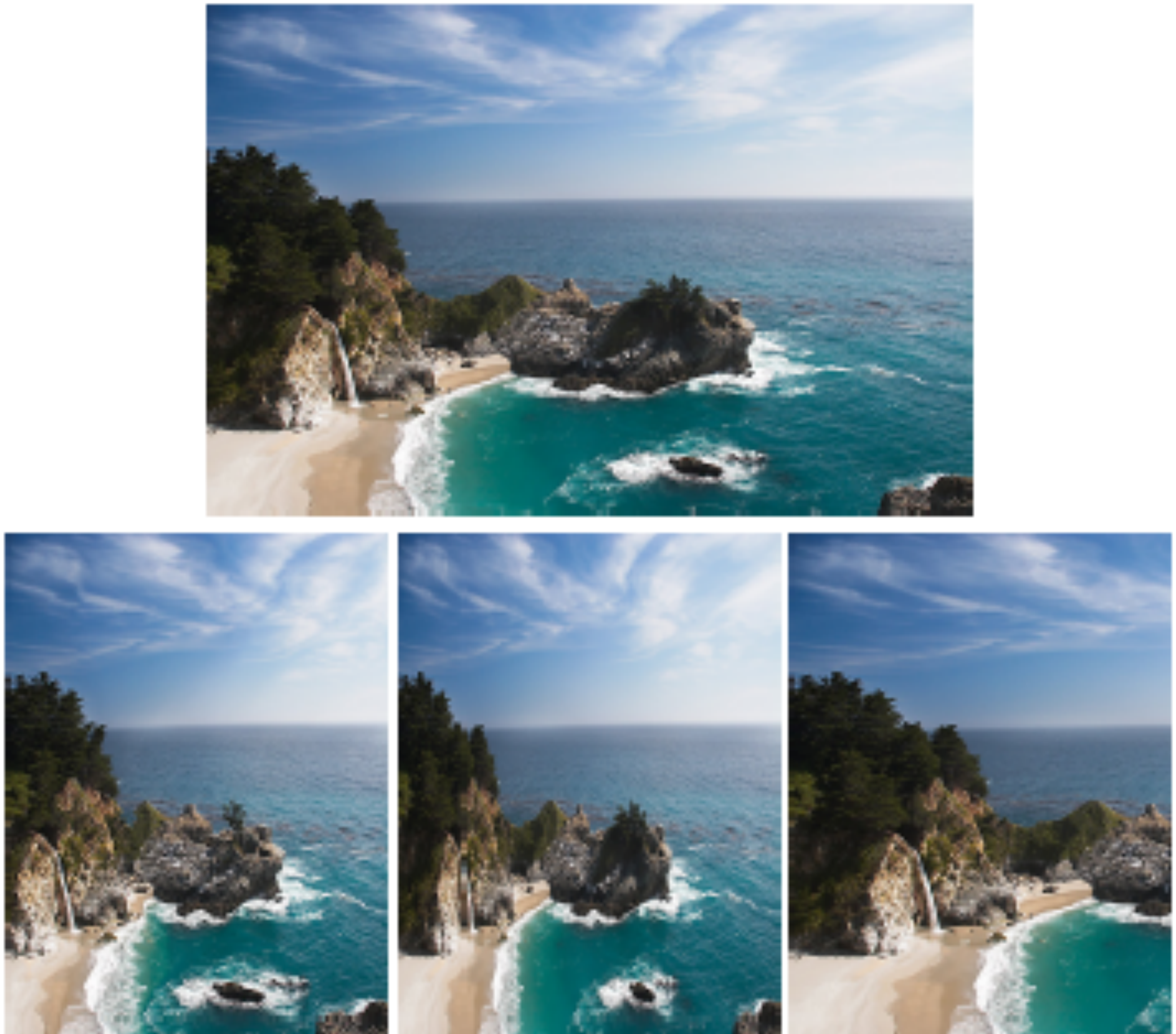
(c) $e_{Entropy}$



(d) e_{HoG}



(e) Segmentation and L_1



Seam removal | Scaling | Cropping

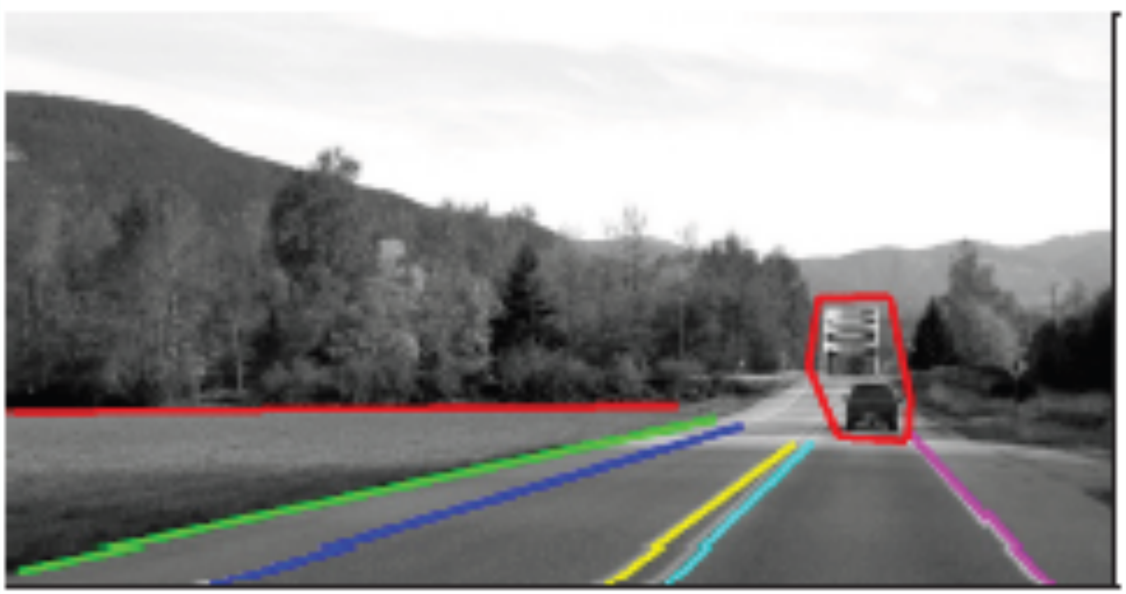
Avidan, Shamir, SIGGRAPH07

Retargeting

Avidan, Shamir, SIGGRAPH07

Avidan, Shamir, SIGGRAPH07

Can use constraints in retargeting

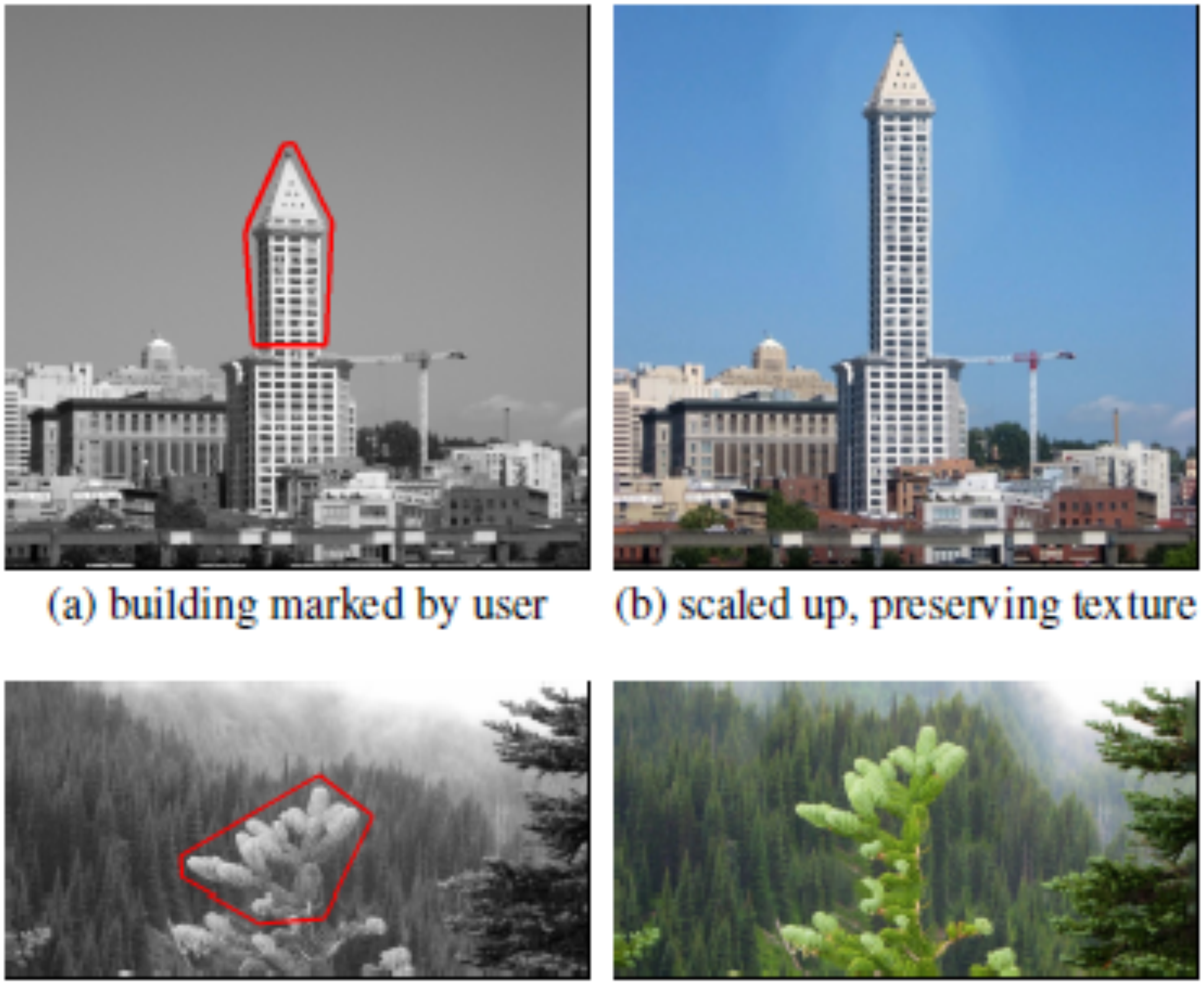


Barnes et.al. SIGGRAPH09

Constrained retargeting

Barnes et.al. SIGGRAPH09

Local scale editing



(a) building marked by user (b) scaled up, preserving texture

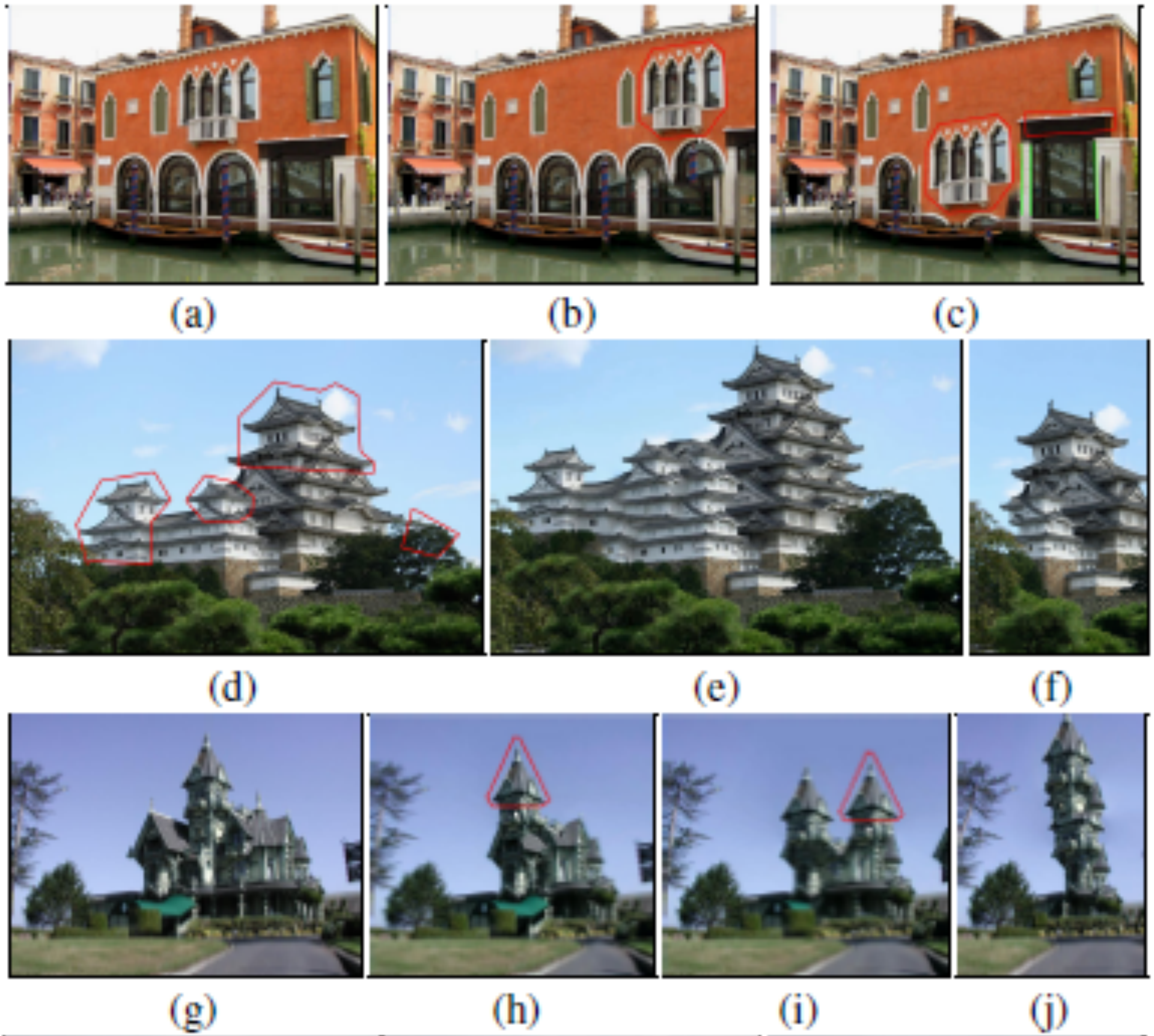
(c) bush marked by user (d) scaled up, preserving texture.

Barnes et.al. SIGGRAPH09

reshuffling

(a) input (b) our reshuffling

Barnes et.al. SIGGRAPH09



Barnes et.al. SIGGRAPH09