Computer Vision for Visual Effects

CVFX 2015

Related Work

- *Image Analogies*
  - Hertzmann *et al.*
  - SIGGRAPH 2001

  - *Fast Texture Synthesis using Tree-Structured Vector Quantization*
  - Wei and Levoy
  - SIGGRAPH 2000

  - *Synthesizing Natural Textures*
  - Ashikhmin
  - 2001 Symposium on Interactive 3D Graphics

Image Analogies

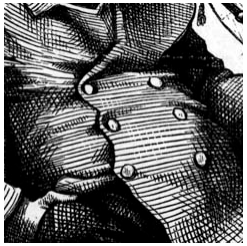
:

::

:

Image Analogies

:



::

:

The Idea of Image Analogies

- A framework for processing images by example
  - For creating complex image filters
- Design phase
  - Training data:
    - A pair of images: unfiltered and filtered
- Application phase
  - The learned filter is applied to some new source image
  - To create an analogous filtered result

Problem

- Given a pair of images *A* and *A'* (the *unfiltered* and *filtered example images*, respectively), along with some additional *unfiltered source image B*, synthesize a new *filtered target image B'* such that *A* : *A'* :: *B* : *B'*

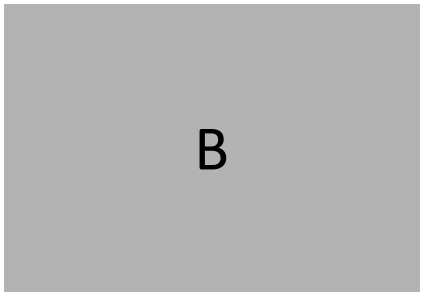
- B' is the analogous image

- B' relates to B in the same way A' relates to A
  - "Make it look like this."

Parametric?

assume a filter of some parametric form

learn the parameters from the pair = learn the filter

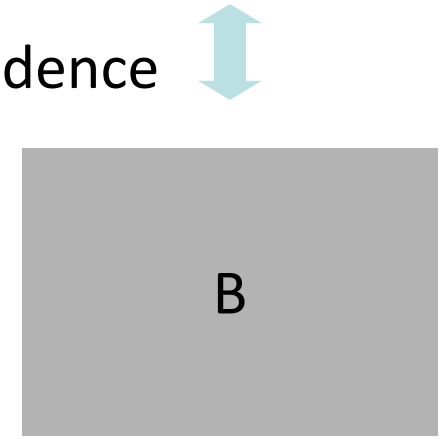


apply the learned filter to B

close correspondence

what features constitute the style?

no direct correspondence

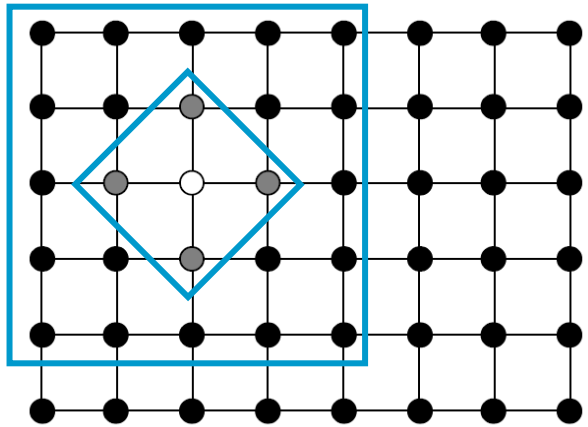


Nonparametric?

- Similarity measure between A and A'
  - Registered
- Similarity measure between A and B
  - Features
- Choosing the appropriate parts of the transform A→A' in synthesizing B→B'

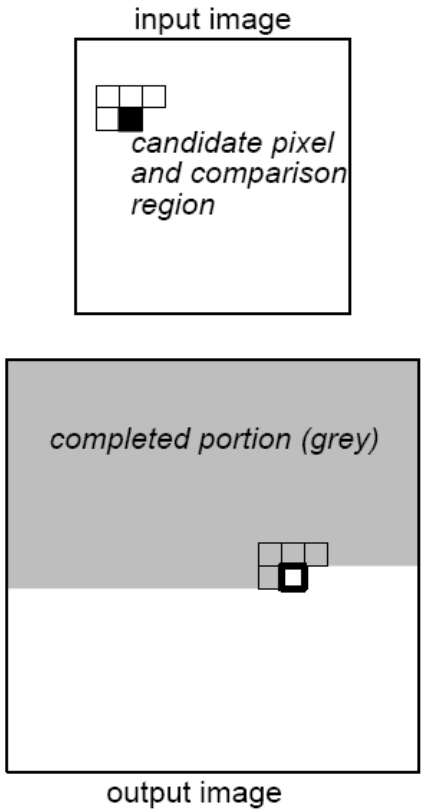
Approximation to an MRF Model

- To estimate the joint statistics of small neighborhoods within the images via sampling

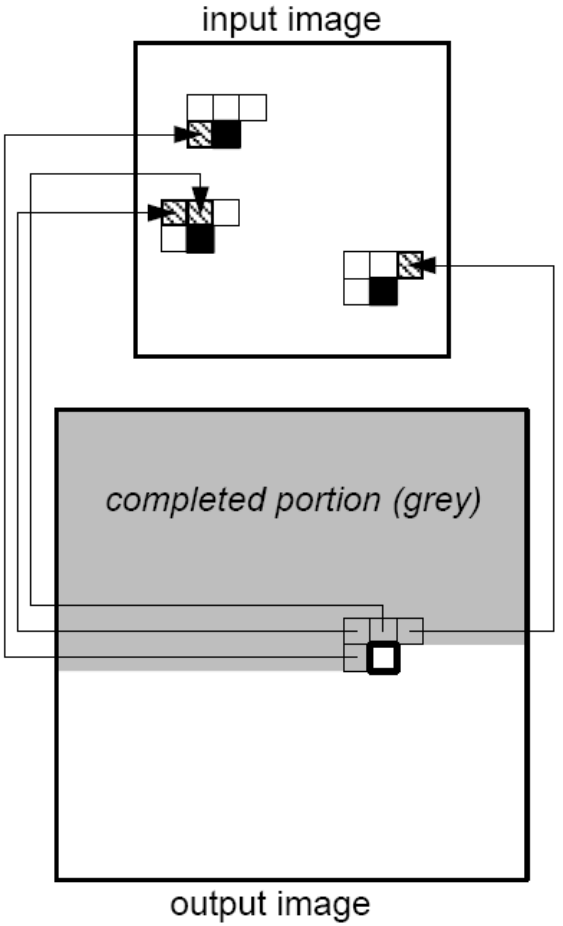


Combining These Two Methods

Wei & Levoy

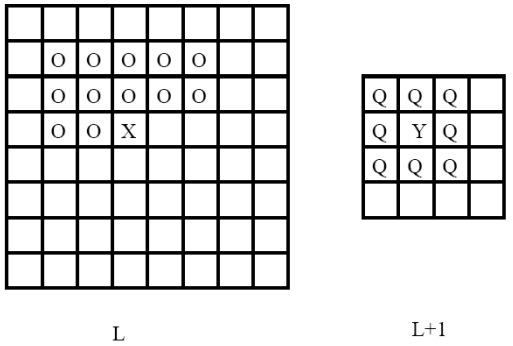


Ashikhmin



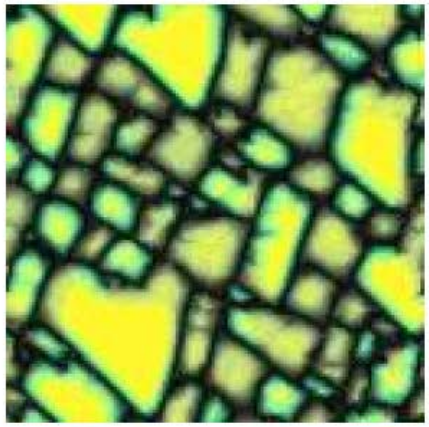
Wei & Levoy's Method

- Fixed neighborhood
  - L shape
- Fixed synthesis order
  - Raster scan
- Multi-resolution
- TSVQ
  - Acceleration
  - Approximate nearest neighbor search

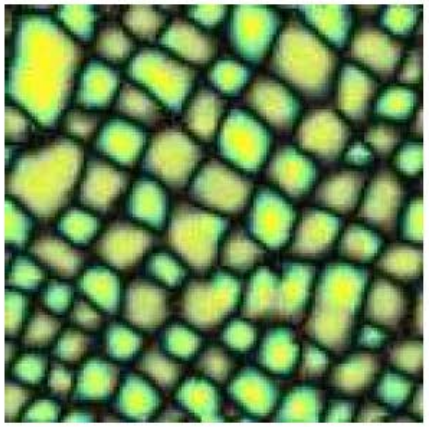


Multi-resolution Synthesis –from Low to High

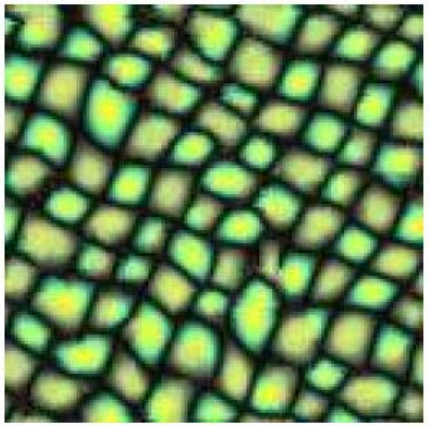
same neighborhood, different numbers of pyramid levels



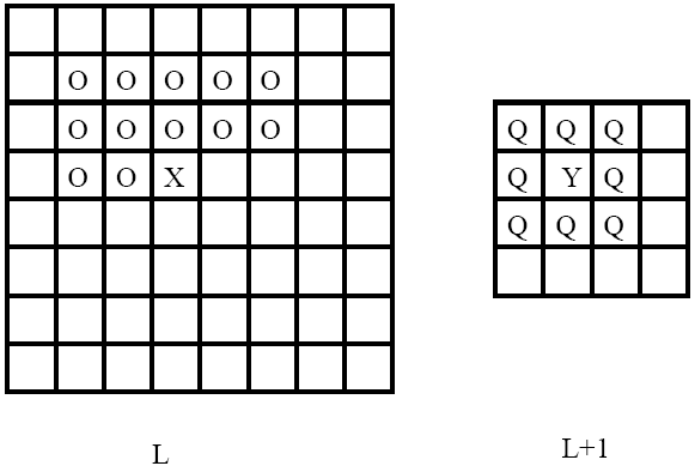
1 level



2 levels



3 levels



use full neighborhood at low-resolution level

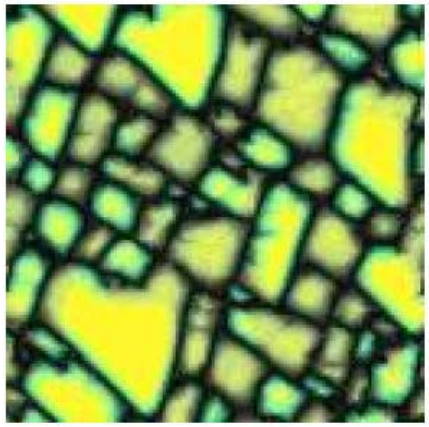
Gaussian Pyramid

- Search over scale
- Blur and downsample

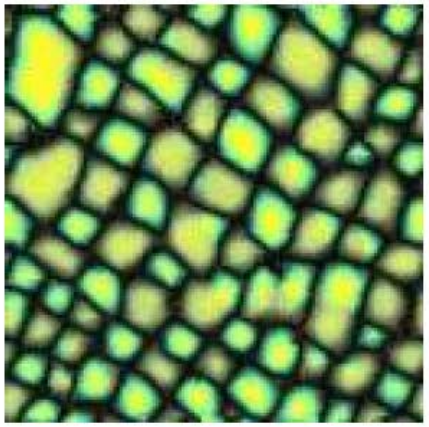
blur and subsample

Multi-resolution Synthesis –from Low to High

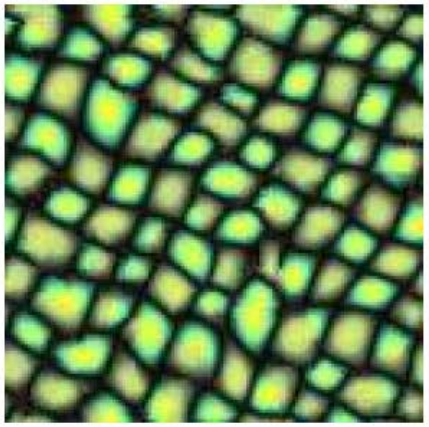
same neighborhood, different numbers of pyramid levels



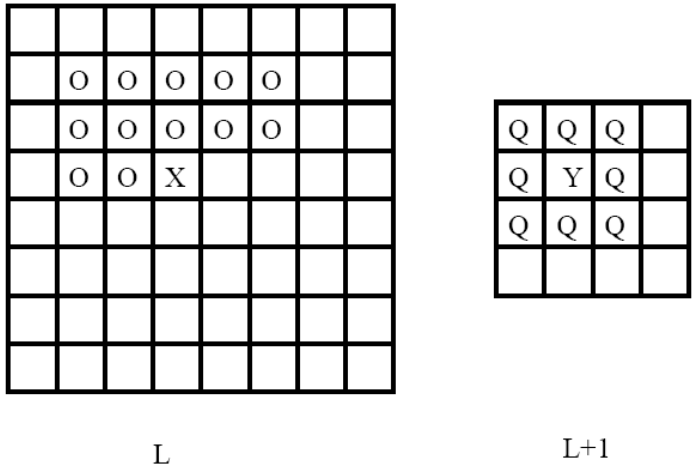
1 level



2 levels



3 levels



use full neighborhood at low-resolution level

Ashikhmin's Method

- Verbatim copy
  - Search the candidate list
- Has difficulty restarting effectively when a patch being copied ends

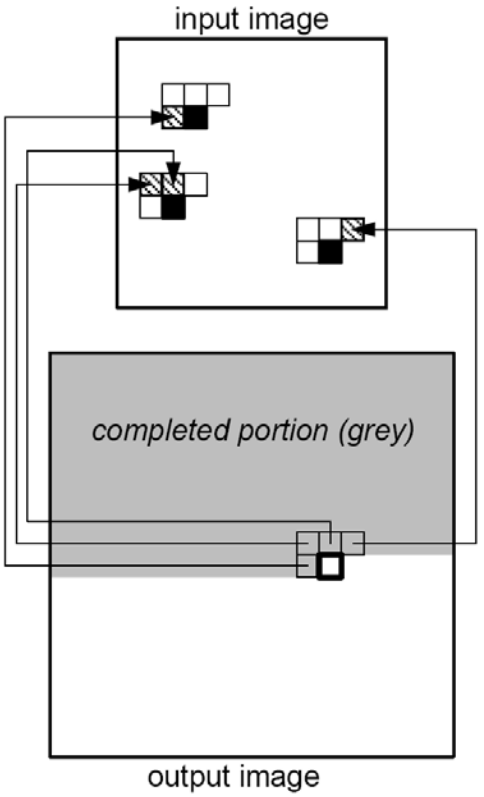


Image Analogies

- Input
  - Unfiltered source image A
  - Filtered source image A'

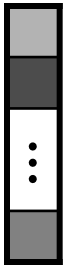
  } registered

  the transform is local

  - Unfiltered source image B
- Output
  - Filtered target image B'

Feature Vector for Each Pixel

- Colors
- Illumination
- Additional filter responses …
- Depends on the problem



- Feature A(p) and feature B(q) use the same representation of feature vector, but A(p) and A'(p) need not to be of the same representation

Notation

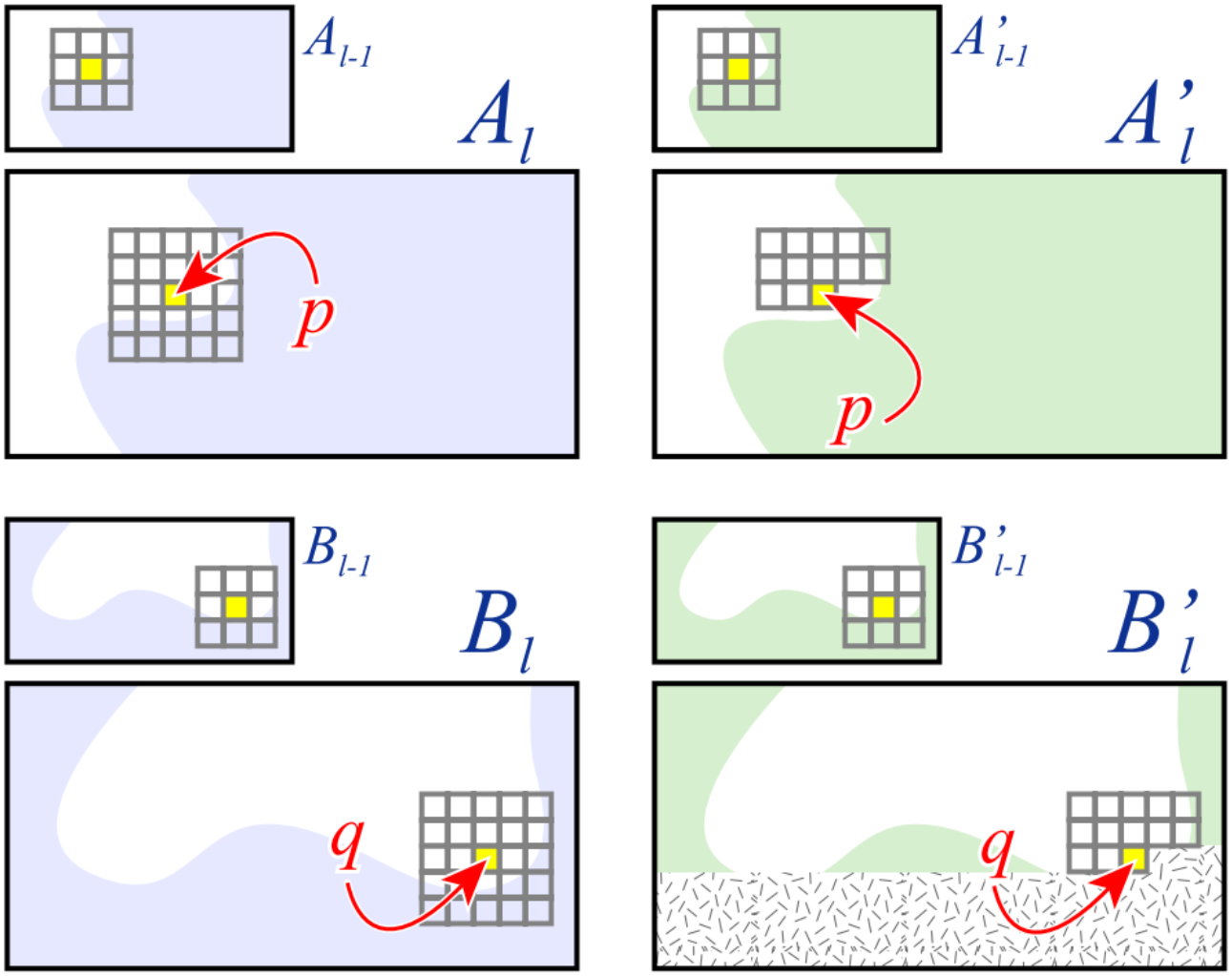
$A(p)$: **array** $p \in$ *SourcePoint* **of** *Feature*
$A'(p)$: **array** $p \in$ *SourcePoint* **of** *Feature*$'$
$B(q)$: **array** $q \in$ *TargetPoint* **of** *Feature*
$B'(q)$: **array** $q \in$ *TargetPoint* **of** *Feature*$'$
$s(q)$: **array** $q \in$ *TargetPoint* **of** *SourcePoint*

Image Analogies Algorithm

- Build Gaussian pyramids of A, A', and B
- Construct data structures for matching
  - Approximate nearest neighbor search (ANN)

- Coarse to fine (for each level)
  - For each pixel q in B' (in raster scan order)
    - Find the best match of B(q), say p in A
    - Copy the feature of A'(p) to B'(q)
    - Record the correspondence s(q) ←→ p

Combine Information from Successive Levels



Approximate Match or Coherence Match

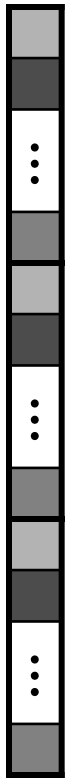
function BESTMATCH(A, A', B, B', s, ℓ, q):
  $p_{\text{app}} \leftarrow$ BESTAPPROXIMATEMATCH(A, A', B, B', ℓ, q)
  $p_{\text{coh}} \leftarrow$ BESTCOHERENCEMATCH(A, A', B, B', s, ℓ, q)
  $d_{\text{app}} \leftarrow \|F_\ell(p_{\text{app}}) - F_\ell(q)\|^2$
  $d_{\text{coh}} \leftarrow \|F_\ell(p_{\text{coh}}) - F_\ell(q)\|^2$
  if $d_{\text{coh}} \leq d_{\text{app}}(1 + 2^{\ell - L}\kappa)$ **then**
    return p_{coh}
  else
    return p_{app}

Gaussian kernel weighted

coherence parameter κ

attenuation factor $2^{\ell - L}$

$F_\ell(p)$ concatenation of all features vectors within some neighborhood in A and A'



Approximate Matching

- Nearest neighbor search
  - ANN $dist(p,q) \leq (1+\epsilon)\, dist(p^*,q)$
    - Tree-based
  - TSVQ
  - PCA for dimensionality reduction

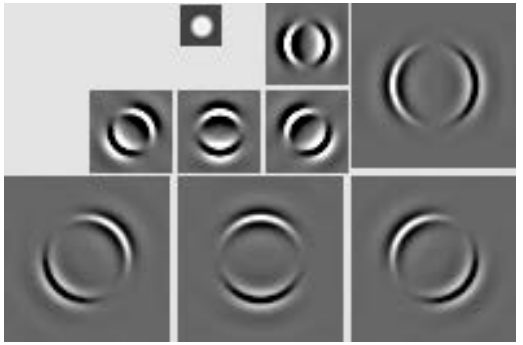
Coherence Matching

The BESTCOHERENCEMATCH procedure simply returns $s(r^\star) + (q - r^\star)$, where

$$r^\star = \arg \min_{r \in N(q)} \|F_\ell(s(r) + (q - r)) - F_\ell(q)\|^2$$

What Features to Use

- The dimensionality of neighborhood space for RGB space is quite large
- For some applications
  - RGB → YIQ
  - Use the Y channel only
- Oriented derivative filters



Luminance Remapping

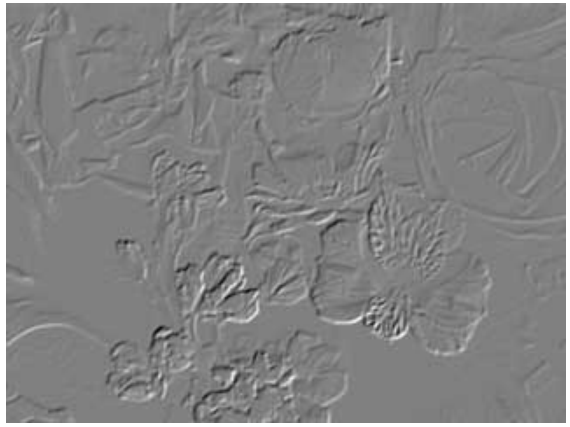
- Linear mapping

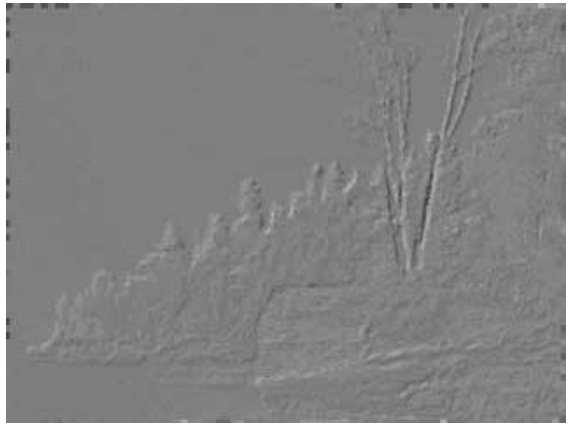
$$Y(p) \leftarrow \frac{\sigma_B}{\sigma_A}(Y(p) - \mu_A) + \mu_B$$

Applications

Traditional Image Filters





Improved Texture Synthesis

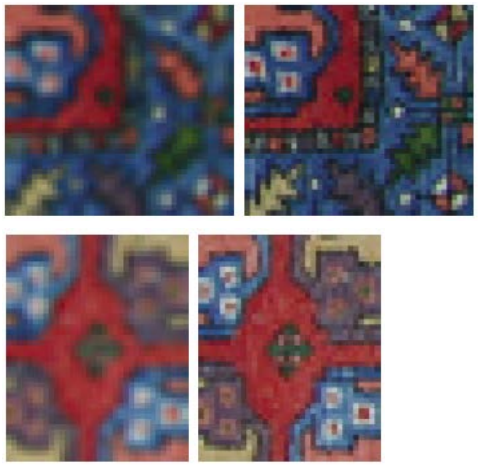
Input

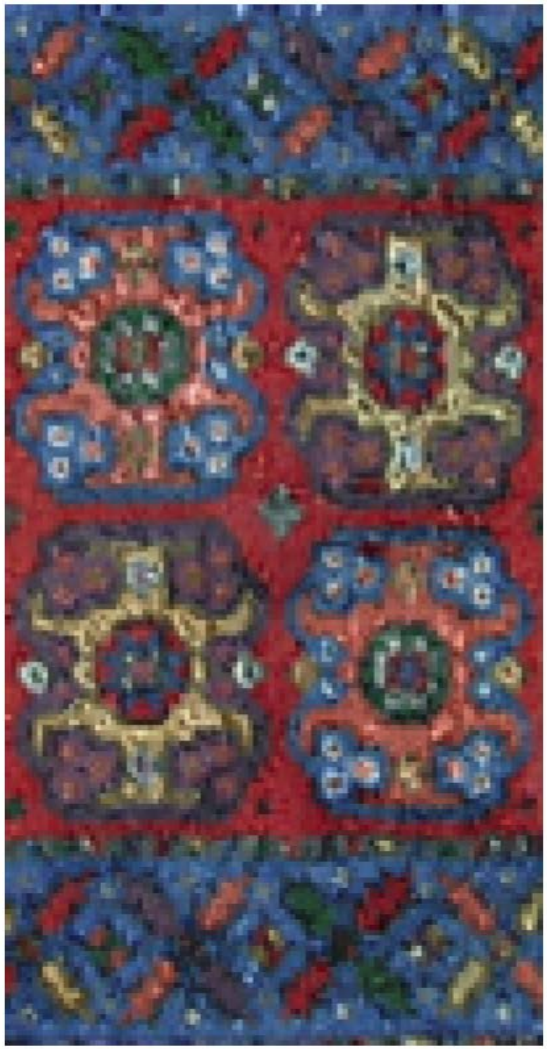
Wei-Levoy

Ashikhmin

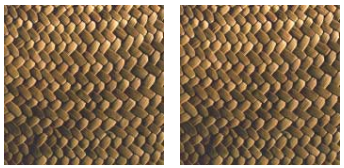
Hertzmann et al.

Super-Resolution

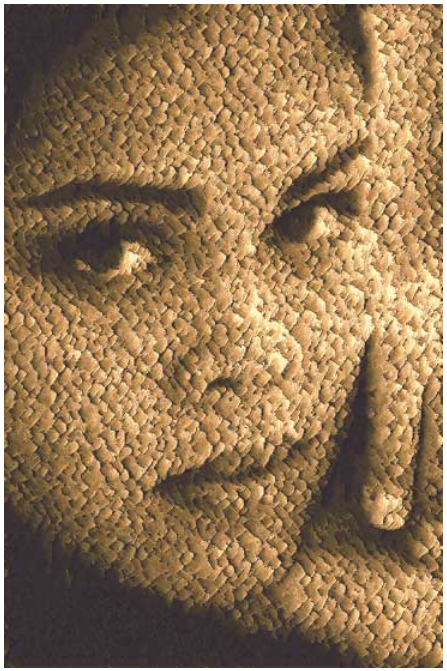




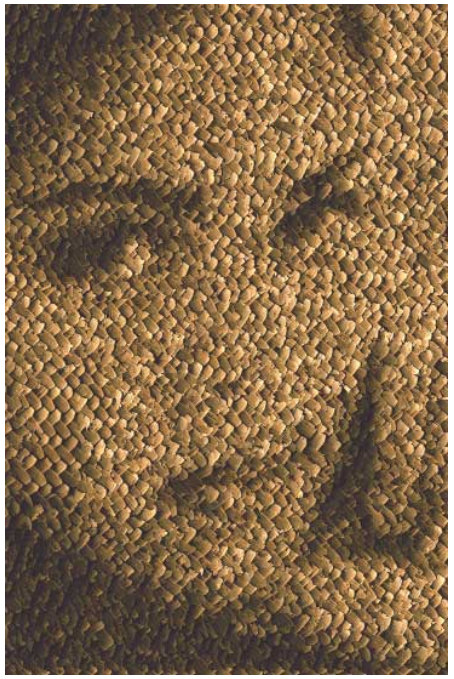
Texture Transfer



closer to the source image



closer to the texture

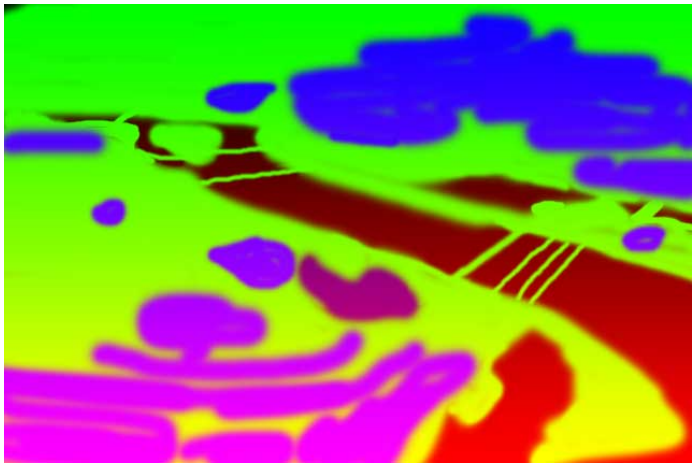


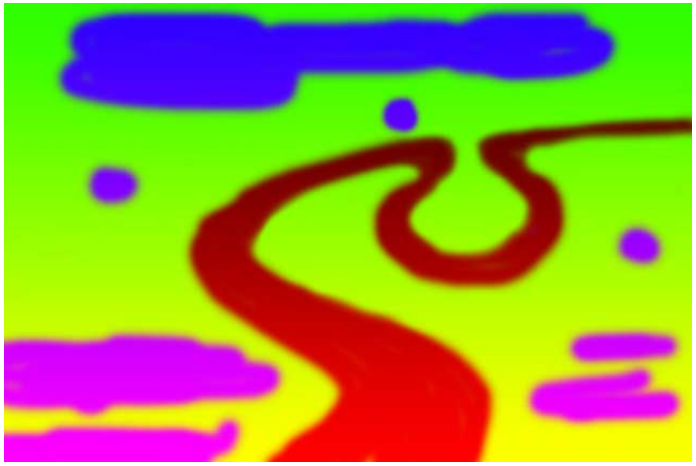
Artistic Filters

How to get this image?

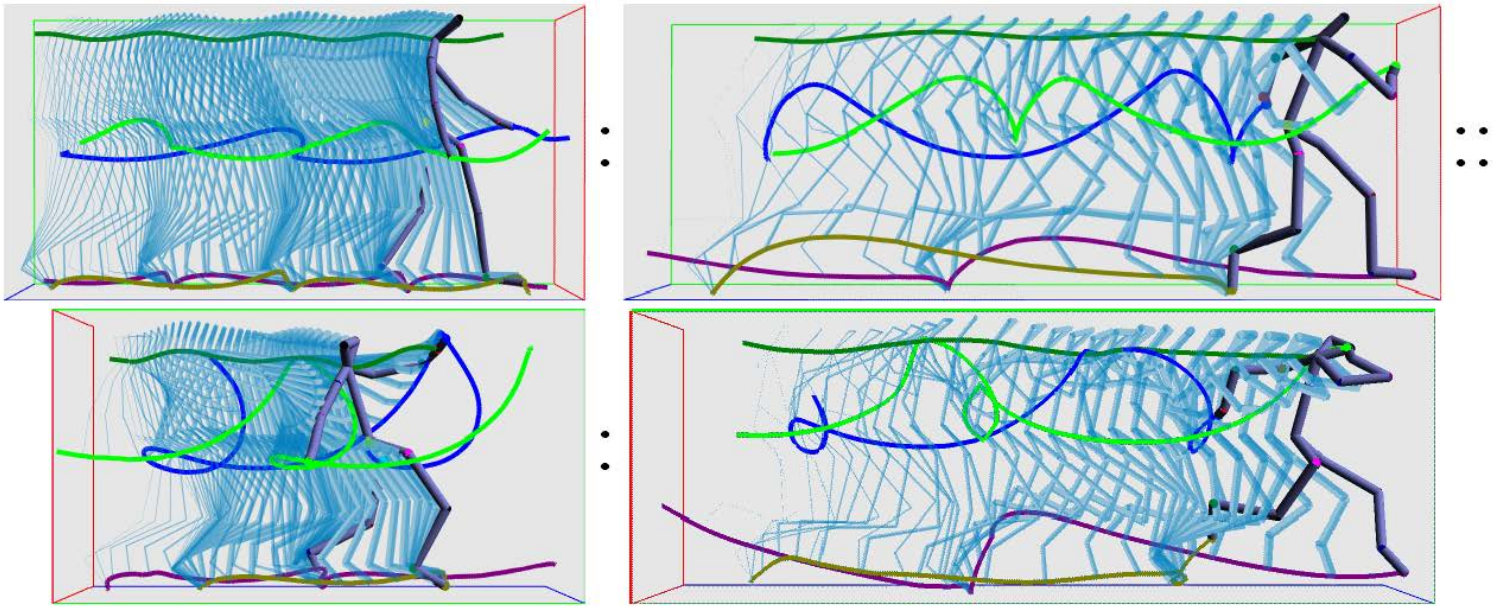
Texture-by-Numbers

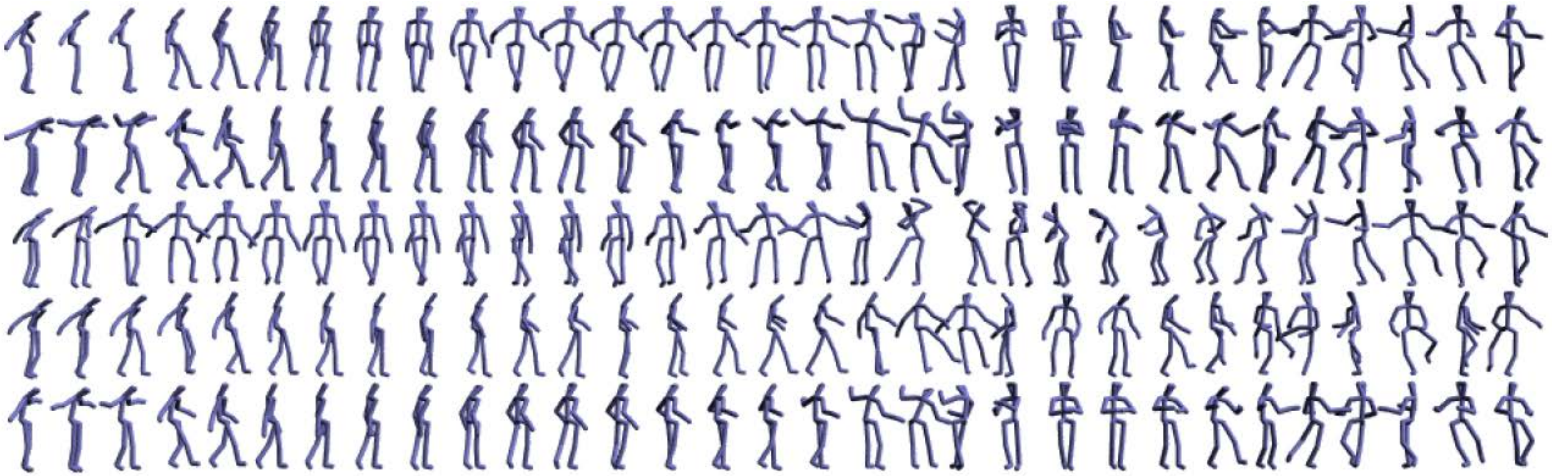
- User-guided texture synthesis






Graphics: Style Machines





Learning Style Translation for the Lines of a Drawing

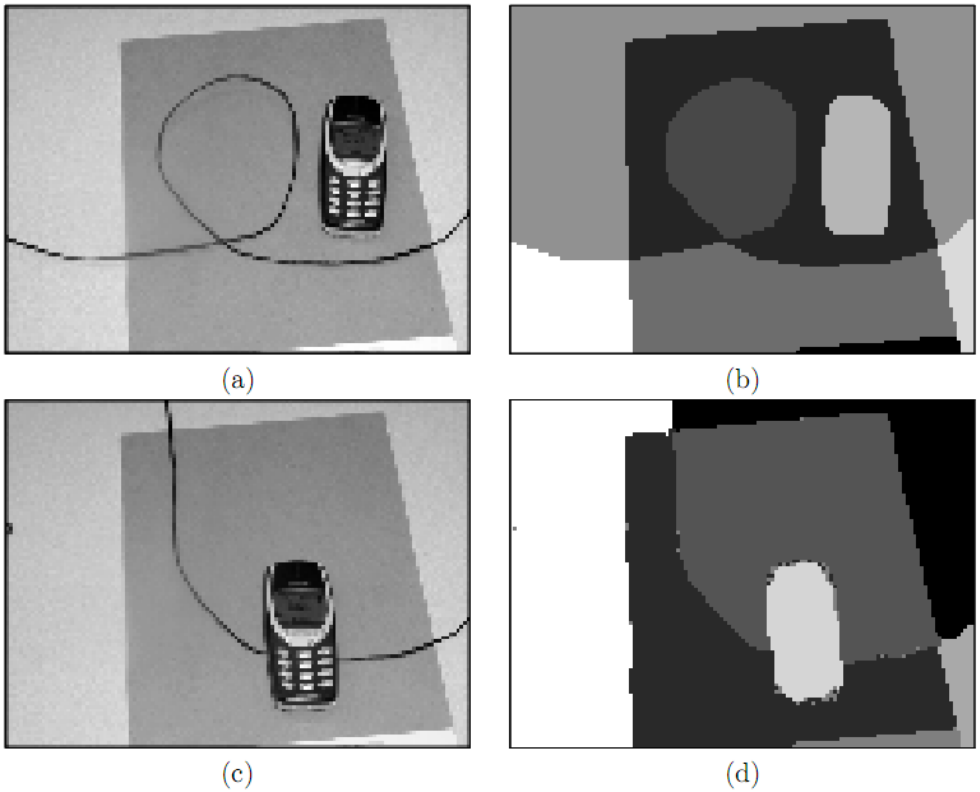
- Freeman, Tenenbaum, and Pasztor

Shape by Example

- Example-Based Photometric Stereo
  - Hertzmann and Seitz

Segmentation by Example

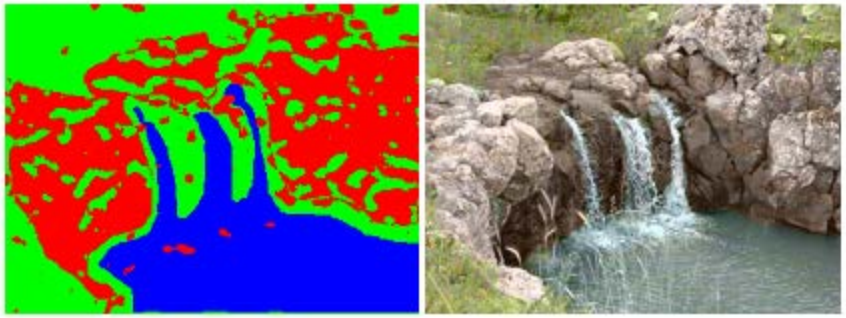
- Segmentation by Example
  - Sameer Agarwal and Serge Belongie



Colorization by Example

- Colorization by Example
- Irony, Cohen-Or, and Lischinski

(b) Reference image along with a partial segmentation.



(c) Our classification and resulting colorization.

Texture Synthesis

- *Texture Synthesis by Non-Parametric Sampling*
  - Efros and Leung
  - ICCV 1999

- *Fast Texture Synthesis using Tree-Structured Vector Quantization*
  - Wei and Levoy
  - SIGGRAPH 2000

- *Synthesizing Natural Textures*
  - Ashikhmin
  - 2001 Symposium on Interactive 3D Graphics

Texture Synthesis & Transfer

- *Image Quilting for Texture Synthesis and Transfer*
  - Efros and Freeman
  - SIGGRAPH 2001
- *Image Analogies*
  - Hertzmann *et al.*
  - SIGGRAPH 2001