Image Analogies
SIGGRAPH 2001

Aaron Hertzmann\(^1\,\!^2\)
Chuck Jacobs\(^2\)
Nuria Oliver\(^2\)
Brian Curless\(^3\)
David Salesin\(^2\,\!^3\)

\(^1\)New York University
\(^2\)Microsoft Research
\(^3\)University of Washington

The Approach

Unfiltered source
Filtered source

Unfiltered target
Filtered target

Function: CREATEIMAGEANALOGY(A, A', B, B')
Compute Gaussian pyramids for A, A', and B
Compute features for A, A', and B
Initialize search structures (e.g., for ANN) for each level \(\ell\), from coarsest to finest. do
for each pixel \(q \in B\), in scan-line order, do:
\(p \leftarrow \text{BESTMATCH}(A, A', B, B', s, \ell, q)\)
\(s(q) \leftarrow p\)
return \(B\)

Function: BESTMATCH(A, A', B, B', s, \ell, q)
\(p_{\text{in}} \leftarrow \text{BESTPHOTOMETRICMATCH}(A, A', B, B', \ell, q)\)
\(d_{\text{in}} \leftarrow \|F_{\text{in}}(p_{\text{in}}) - F(p_{\text{in}})\|^2\)
if \(d_{\text{in}} < (d_{\text{in}} + d_{\text{out}}) + \epsilon\) then
return \(p_{\text{in}}\)
else
return \(p_{\text{out}}\)
Implementation Details

• Use approximate nearest neighbor search and Ashikhmin’s *coherence search* heuristic
• Use *feature vectors* instead of pixel values
  – Feature vector can consist of RGB values plus additional “channels” such as luminance, outputs of derivative filters
• *Luminance remapping* to align color histograms of source and target images

Blur Filter

Edge Filter

Colorization

Texture Synthesis

• Source images (A, B) are blank/constant
Texture Transfer

- A and A’ is the same (or A is a blurred version of A’)
- Optional: Tunable weight to control the tradeoff between matching (A, B) and (A’, B’)

Unfiltered source

Filtered source

Artistic Filters

Artistic Filters
Artistic Filters

• Project idea: inverse “texture by numbers”

Dealing with progressively variant textures
Super-resolution

Super-resolution (result!)