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

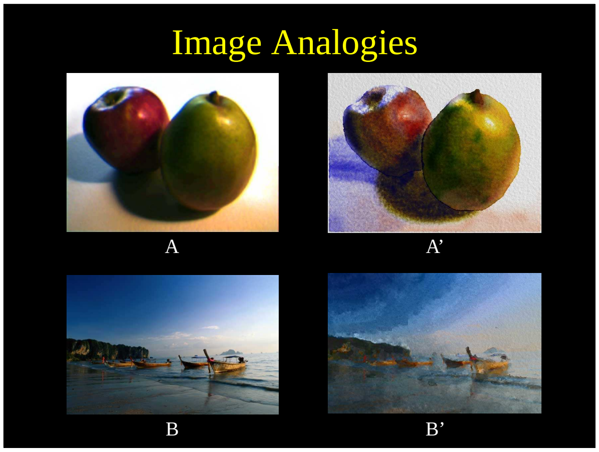


Image Analogies

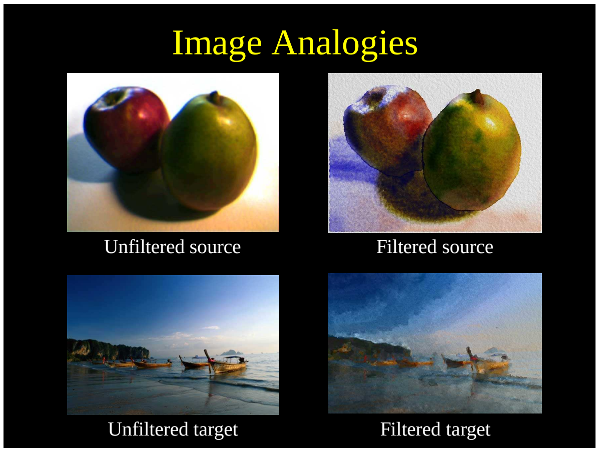
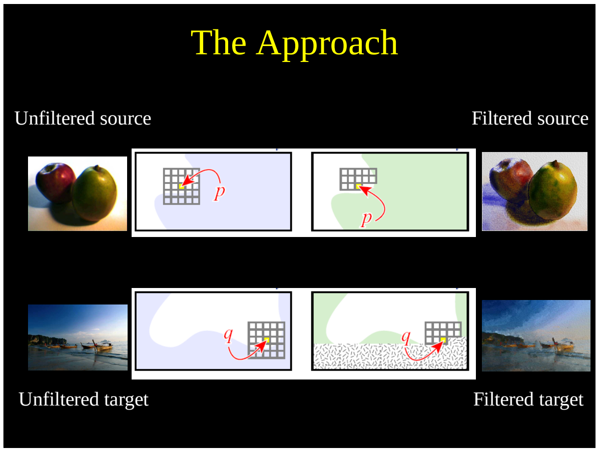
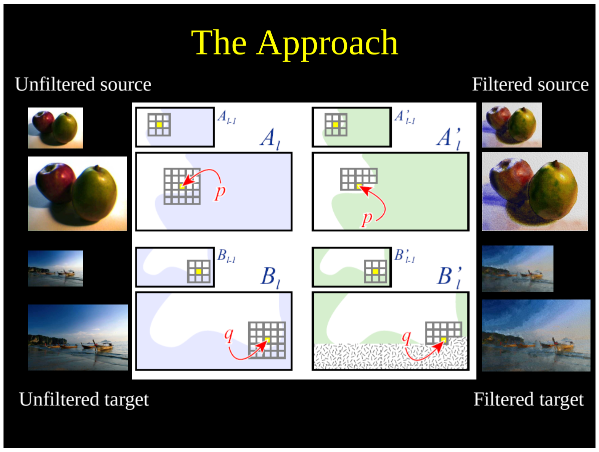


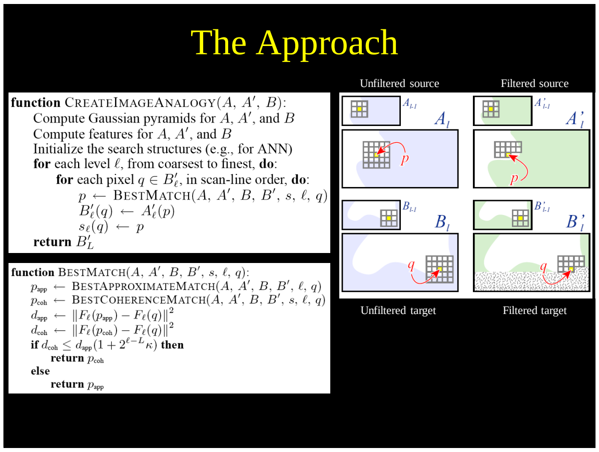
Image Analogies



The Approach



The Approach



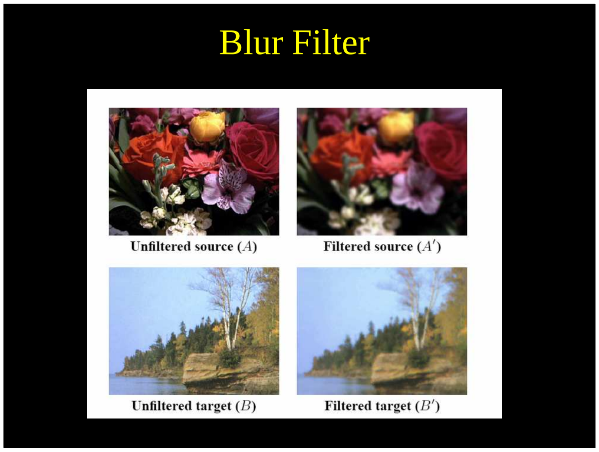
The Approach

```
function CREATEIMAGEANALOGY($A$, $A'$, $B$):
    Compute Gaussian pyramids for $A$, $A'$, and $B$
    Compute features for $A$, $A'$, and $B$
    Initialize the search structures (e.g., for ANN)
    for each level $\ell$, from coarsest to finest, do:
        for each pixel $q \in B'_\ell$, in scan-line order, do:
            $p \leftarrow$ BESTMATCH($A$, $A'$, $B$, $B'$, $s$, $\ell$, $q$)
            $B'_\ell(q) \leftarrow A'_\ell(p)$
            $s_\ell(q) \leftarrow p$
    return $B'_L$
```

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

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



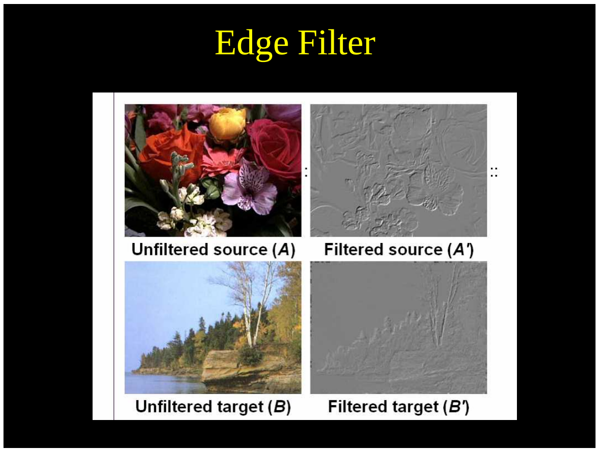
Blur Filter

Unfiltered source (A)

Filtered source (A')

Unfiltered target (B)

Filtered target (B')



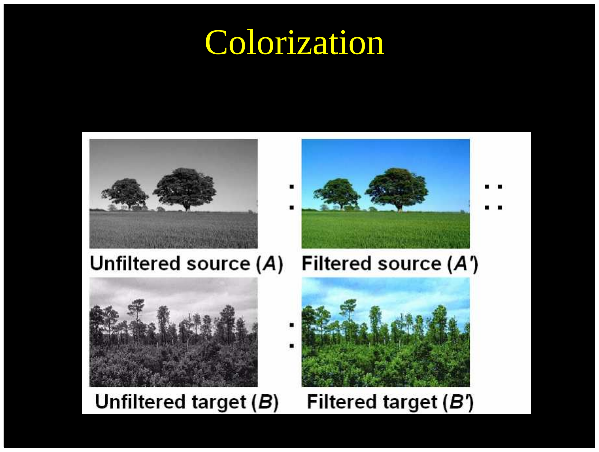
Edge Filter

Unfiltered source (*A*)

Filtered source (*A'*)

Unfiltered target (*B*)

Filtered target (*B'*)



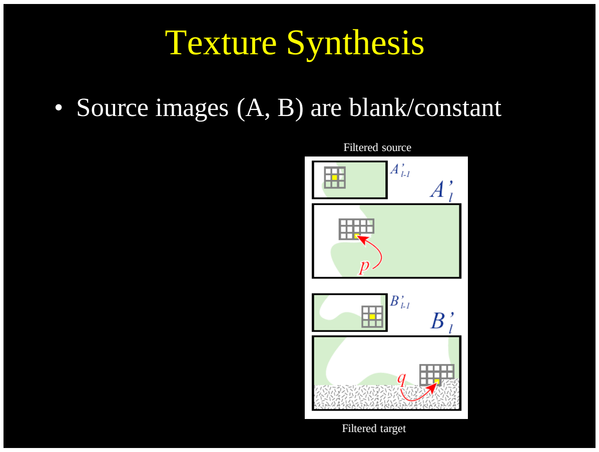
Colorization

Unfiltered source (*A*)

Filtered source (*A'*)

Unfiltered target (*B*)

Filtered target (*B'*)

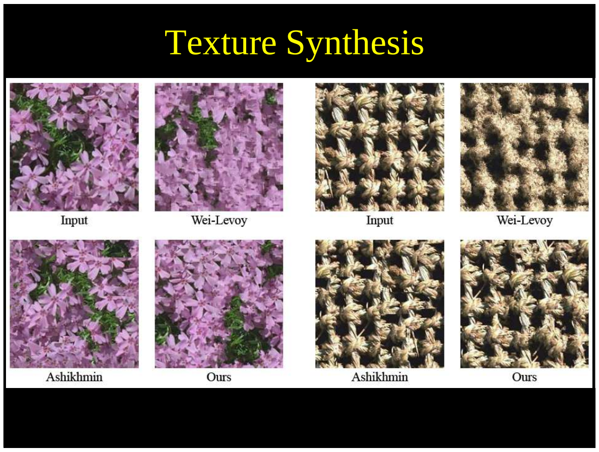


Texture Synthesis

- Source images (A, B) are blank/constant

Filtered source

Filtered target



Texture Synthesis

Input

Wei-Levoy

Input

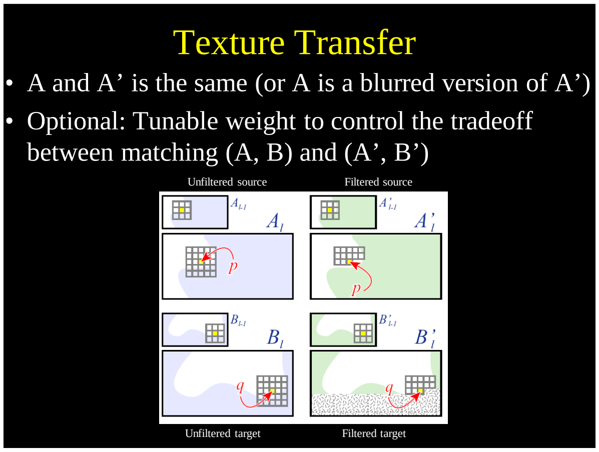
Wei-Levoy

Ashikhmin

Ours

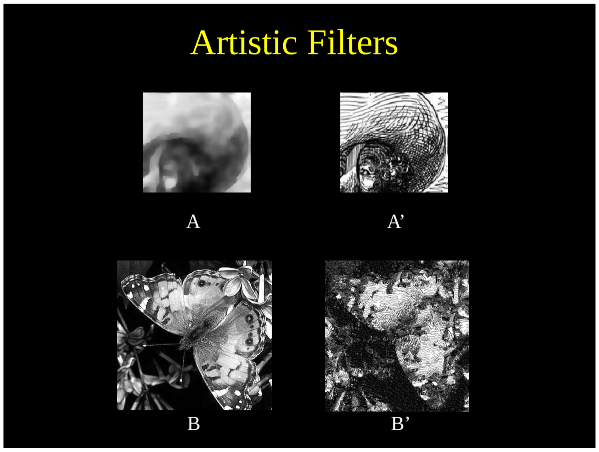
Ashikhmin

Ours

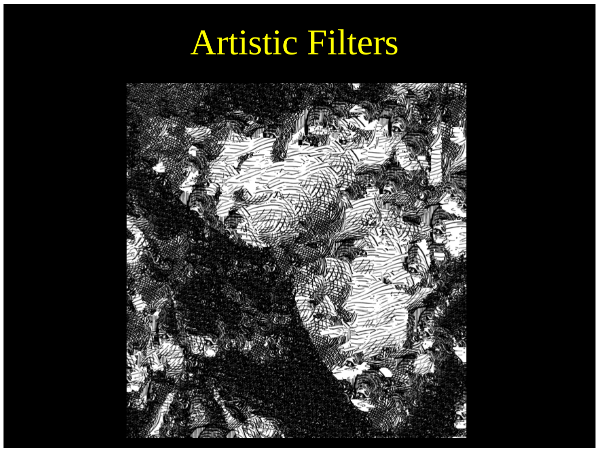


Texture Transfer

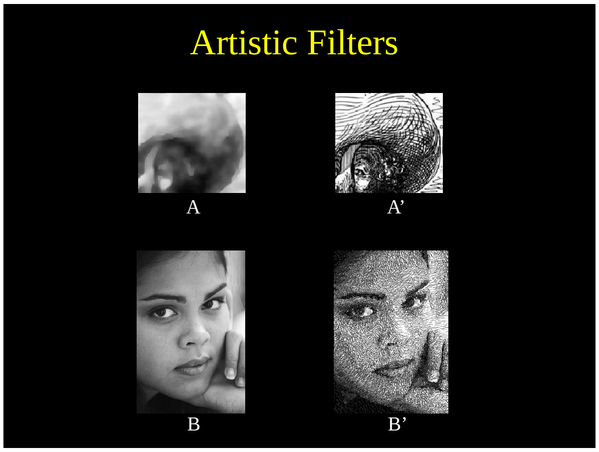
- A and A’ is the same (or A is a blurred version of A’)
- Optional: Tunable weight to control the tradeoff between matching (A, B) and (A’, B’)



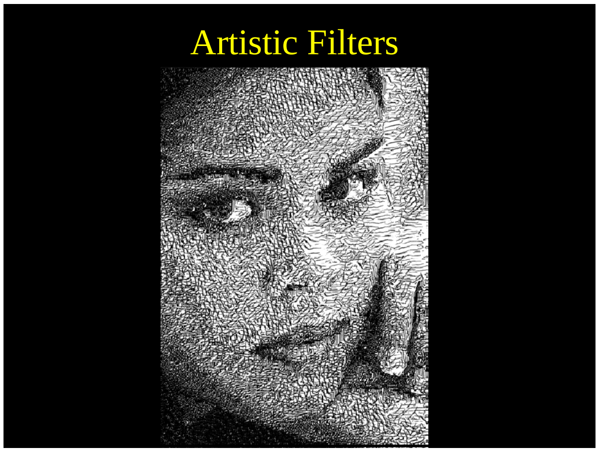
Artistic Filters



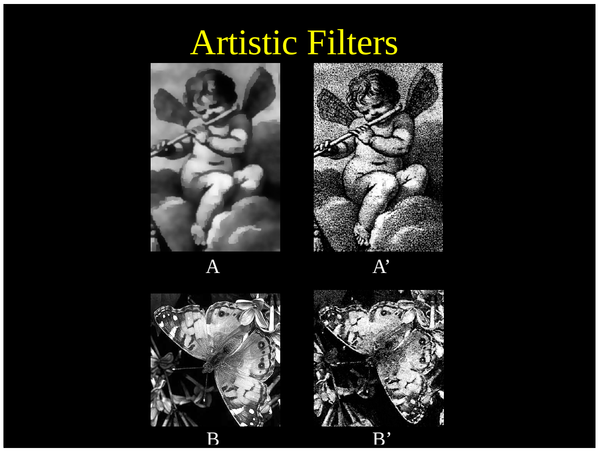
Artistic Filters



Artistic Filters

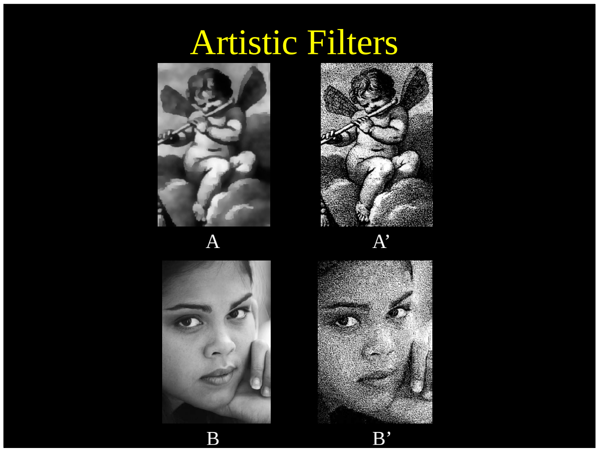


Artistic Filters

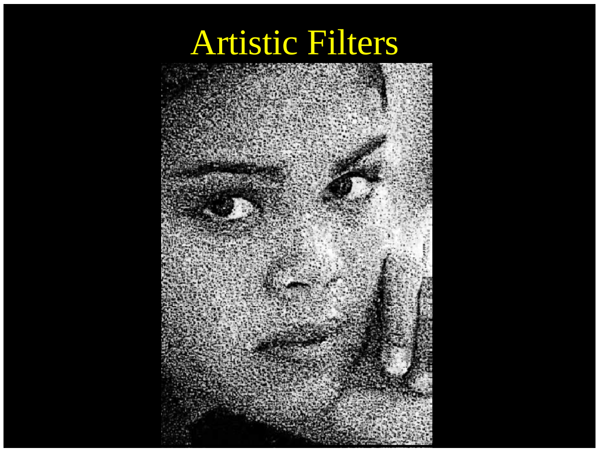


Artistic Filters

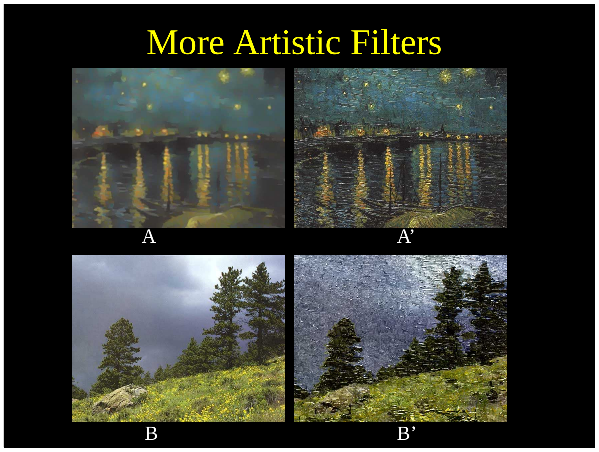
Artistic Filters



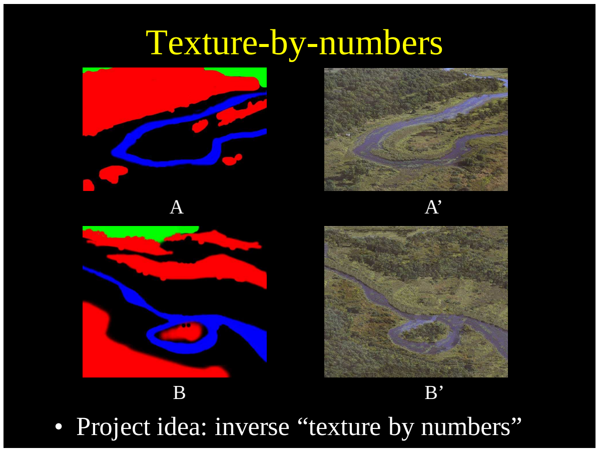
Artistic Filters



Artistic Filters

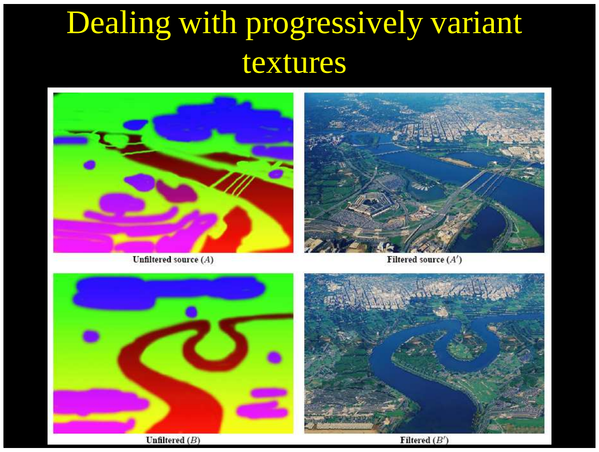


More Artistic Filters

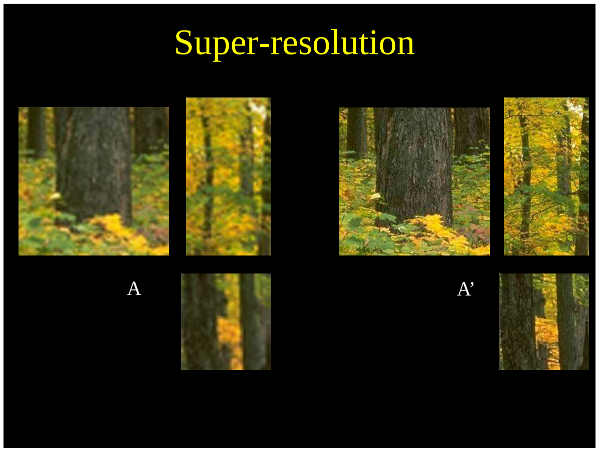


Texture-by-numbers

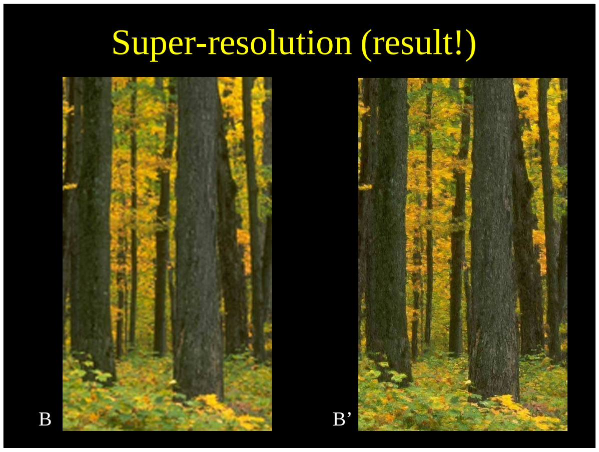
- Project idea: inverse “texture by numbers”



Dealing with progressively variant textures



Super-resolution



Super-resolution (result!)