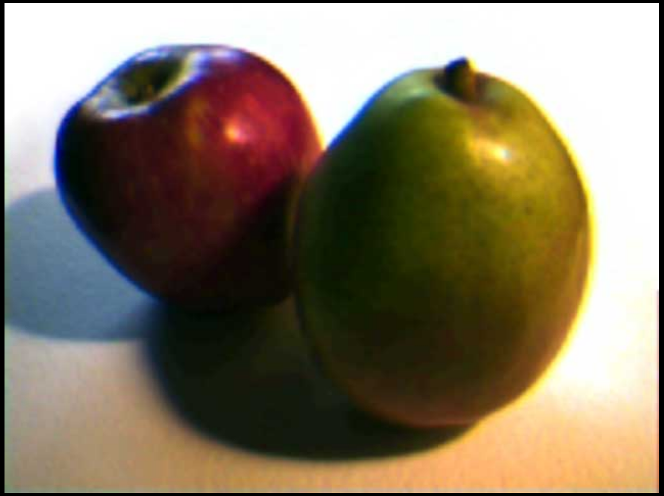
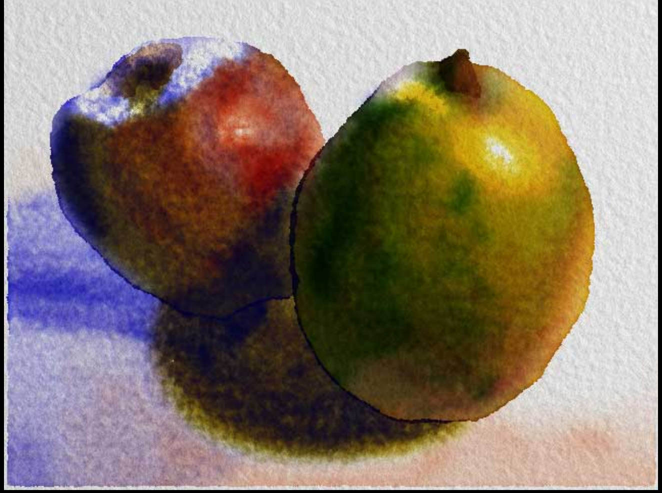
Image Analogies

报告人：沈心逸

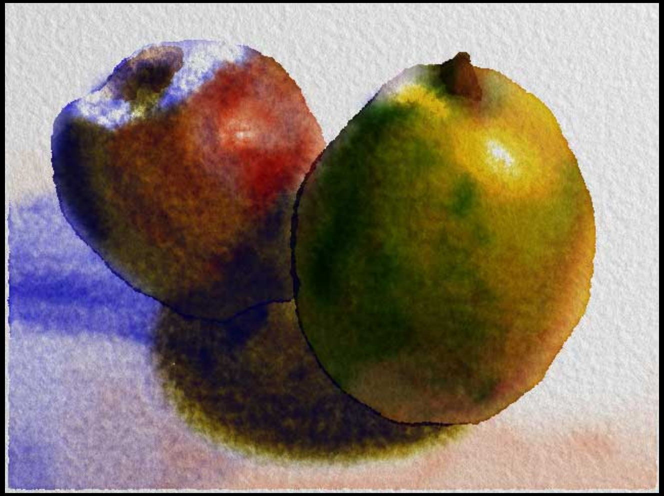
Introduction

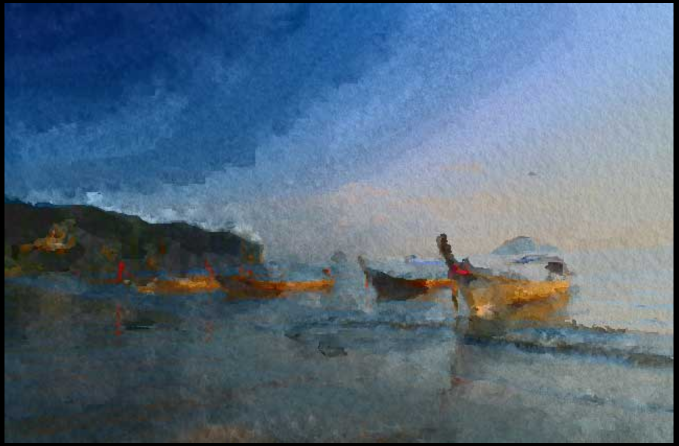




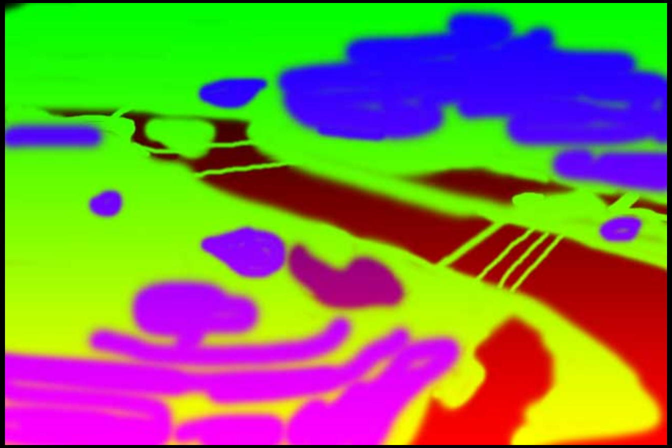
?

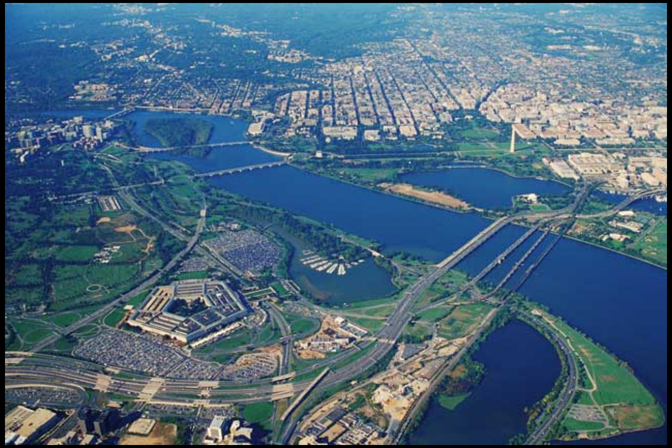
Introduction

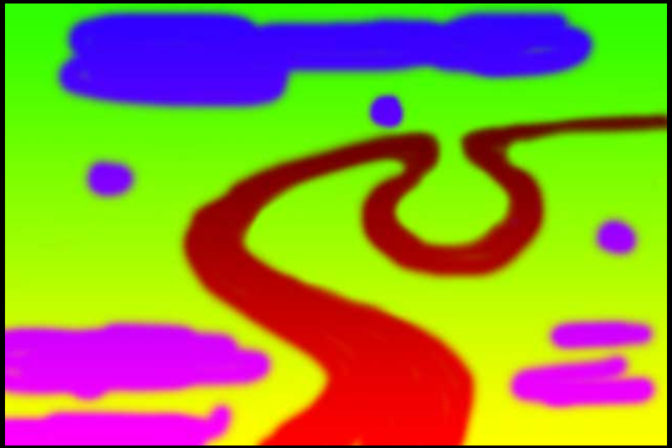




Introduction

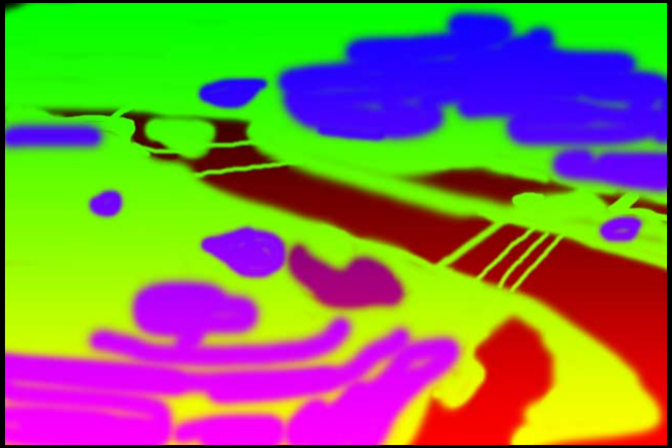


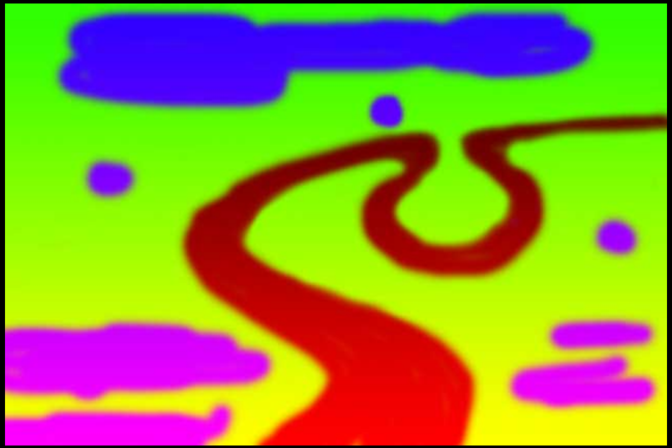




?

Introduction





Introduction

- Our problem is to compute a new “analogous” image B' that relates to B in “the same way” as A' relates to A.

- Here, A, A', and B are inputs to our algorithm, and B' is the output.

$$A : A' \quad :: \quad B : B'$$

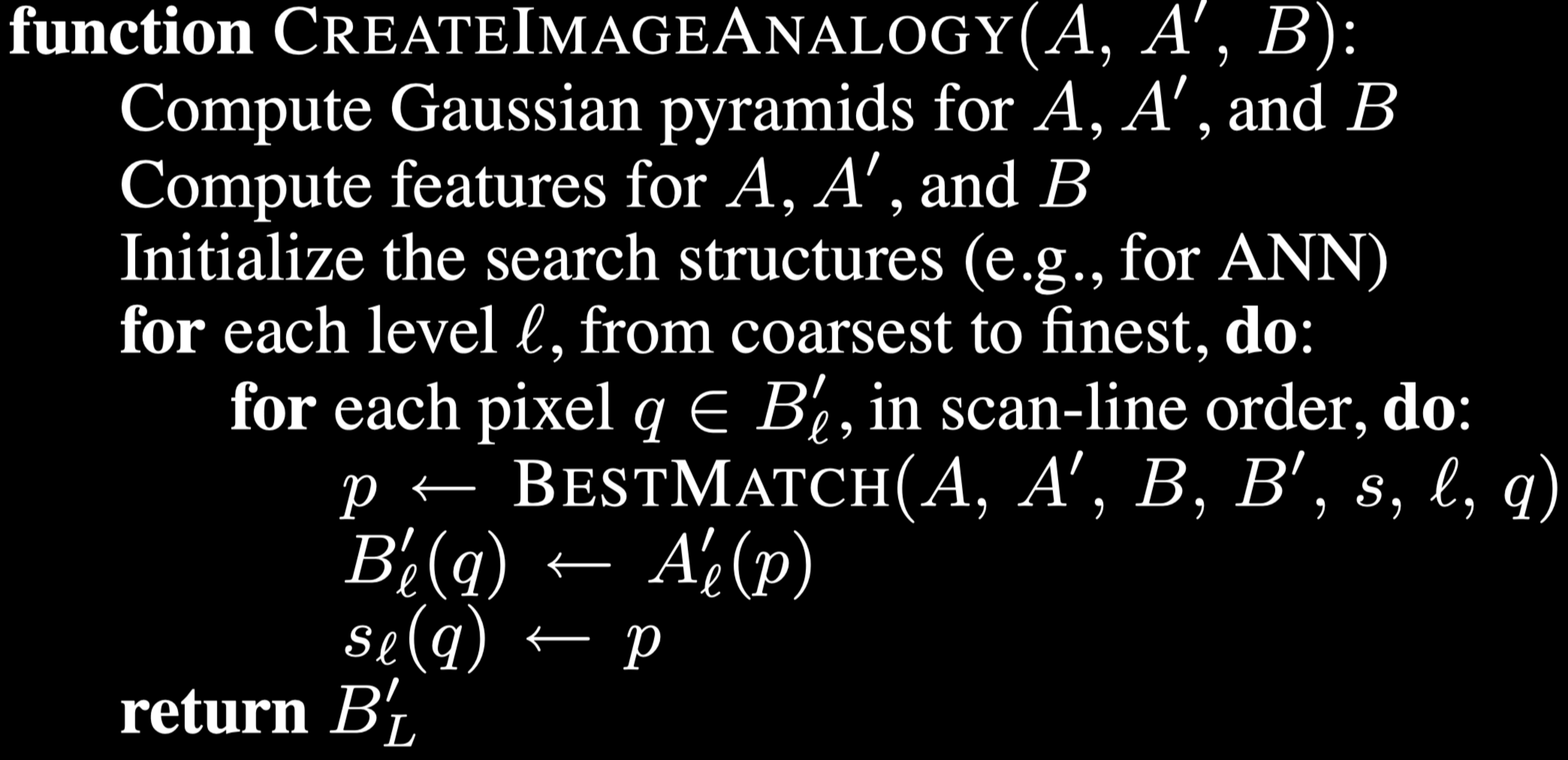
Application

- **Toy filters**, such as blurring or embossing.
- **Texture synthesis** from an example texture.
- **Super-resolution**, inferring a high-resolution image from a low-resolution source.
- **Texture transfer**, in which images are "texturized" with some arbitrary source texture.
- **Artistic filters**, in which various drawing and painting styles, including oil, pastel, and pen-and-ink rendering, are synthesized based on scanned real-world examples.
- **Texture-by-numbers**, in which realistic scenes, composed of a variety of textures, are created using a simple "painting" interface.
- **Image colorization**, where color is automatically added to grayscale images.

Algorithm

1. Generate Gaussian pyramids for A, A' and B
2. Compute features for A, A' and B
3. Initialize the search structures for ANN
4. Find the best match for each pixel in each level

Algorithm

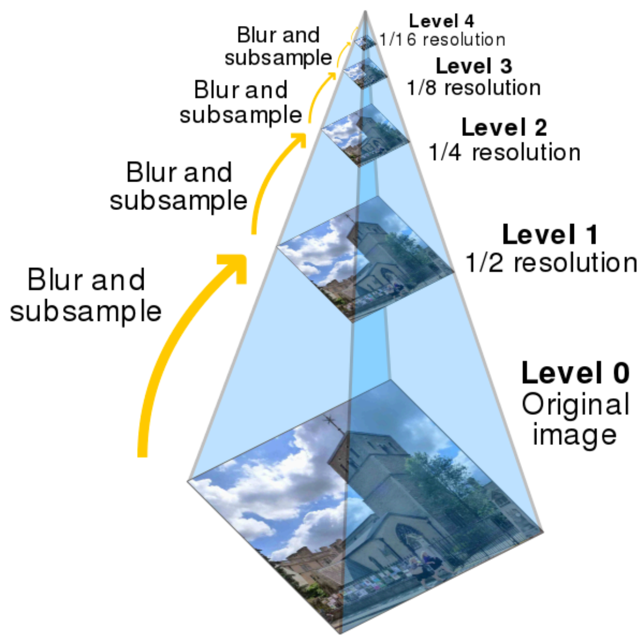


```
function CreateImageAnalogy($A$, $A'$, $B$):
    Compute Gaussian pyramids for $A$, $A'$, and $B$
    Compute features for $A$, $A'$, and $B$
    Initialize the search structures (e.g., for ANN)
    for each level $\ell$, from coarsest to finest, do:
        for each pixel $q \in B'_\ell$, in scan-line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

Gaussian Pyramids

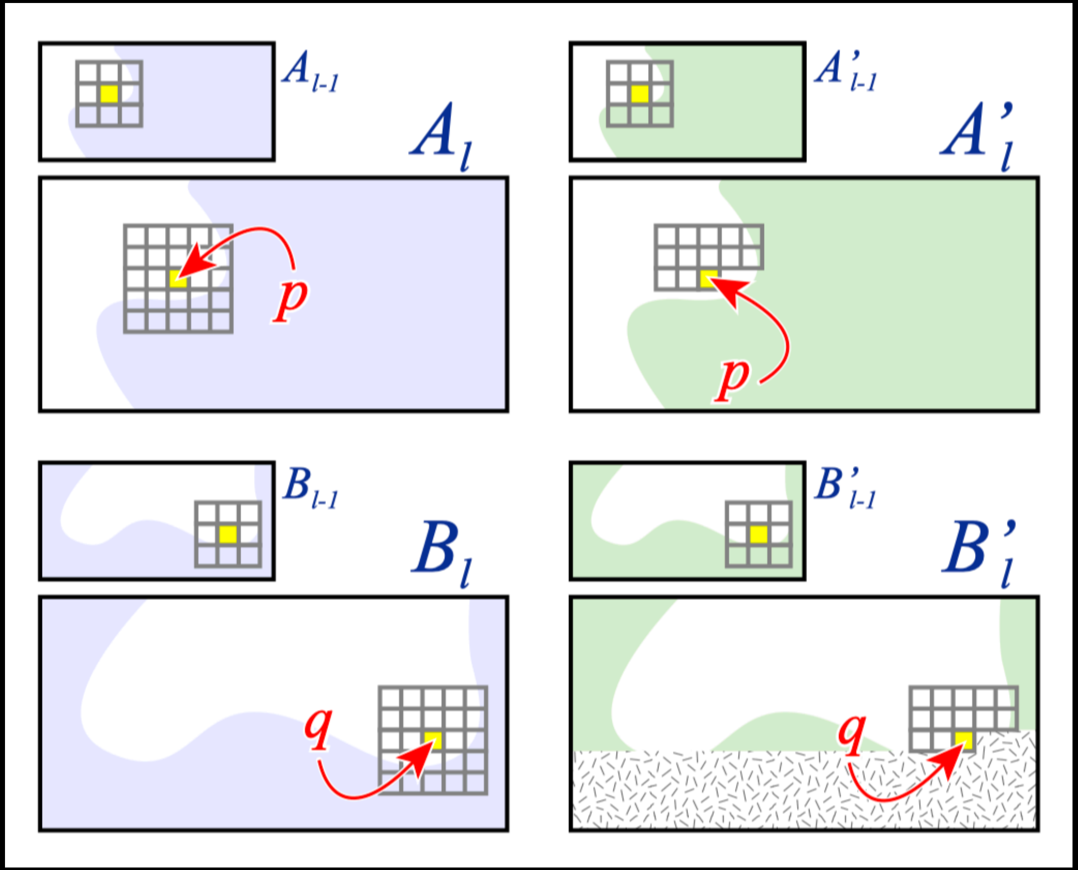
- Smooth the image with Gaussian filter kernel

- Subsample the smoothed image by a factor of 2 along each coordinate direction.



Features

- RGB
- Luminance



ANN

- Approximate Nearest Neighbors
- KD Tree
- Library: Flann
- Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. Journal of the ACM, 45(6):891–923, 1998.

Best Match

```
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 $p_{\text{coh}}$
    else
        return $p_{\text{app}}$
```

Best Match

- Best Approximate Match: ANN
- Best Coherence Match

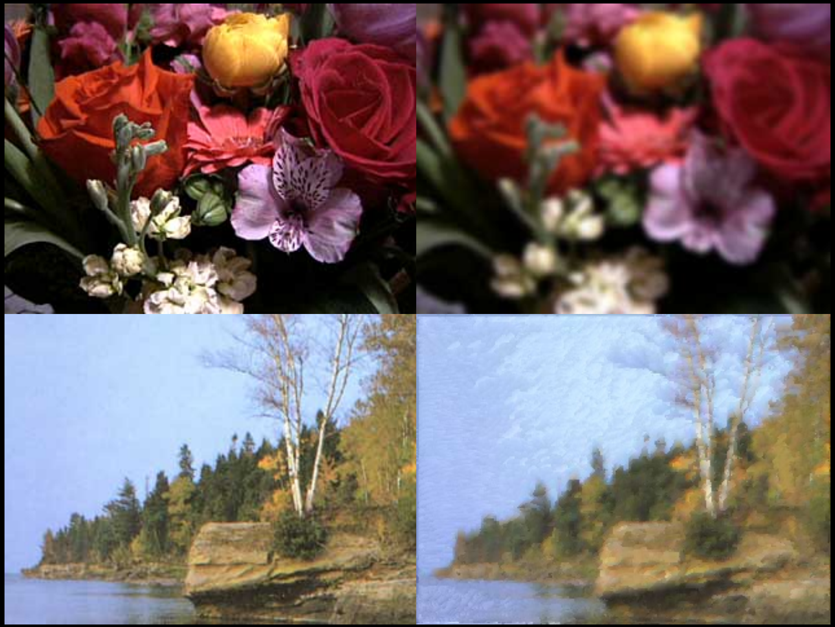
$$s(r*) + (q - r*)$$

where

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

Results

- Blurring



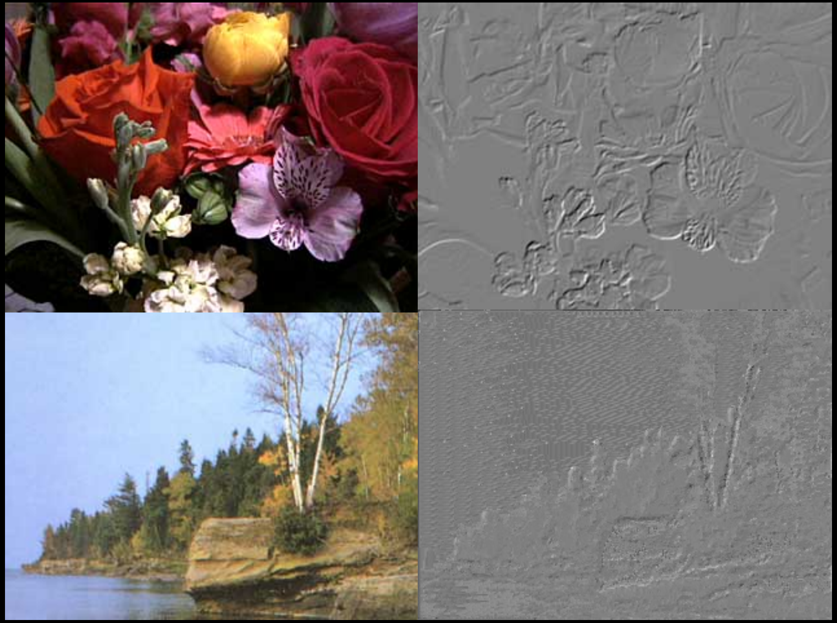
320 * 240 pixels

Run time: 27.4294s

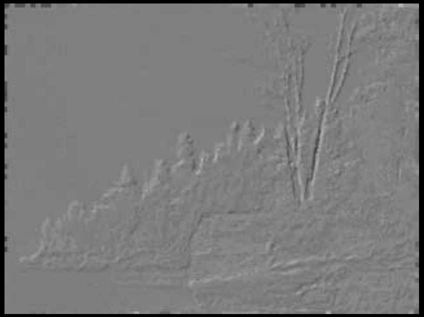
Paper result

Results

- Embossing



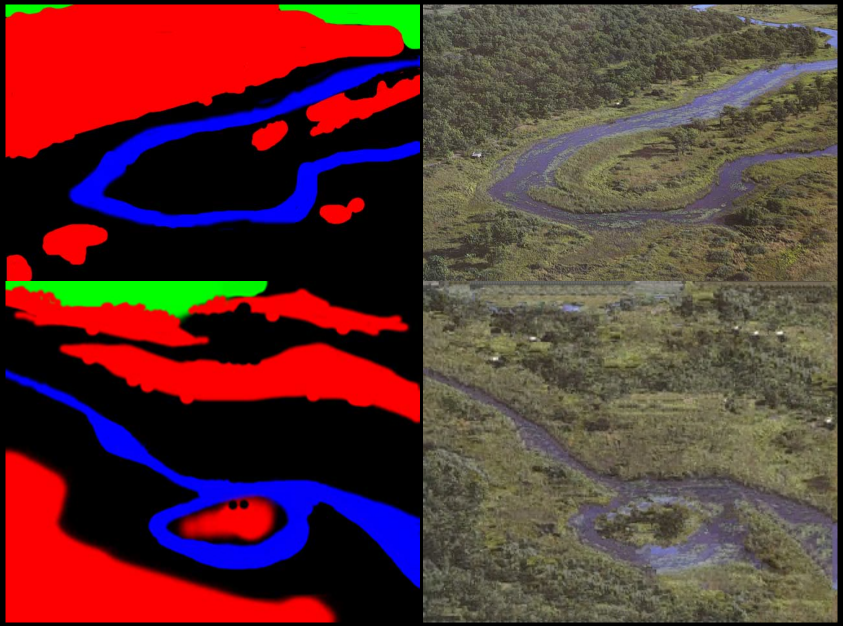
320 * 240 pixels
Run time: 32.1190s



Paper result

Results

- Texture-by-numbers: Oxbow

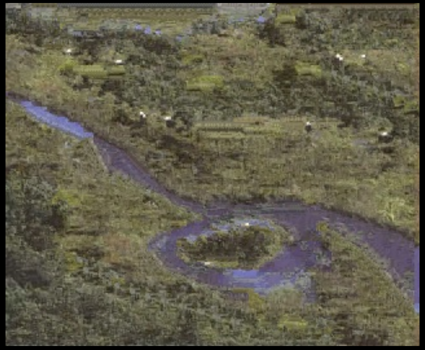


320 * 265 pixels

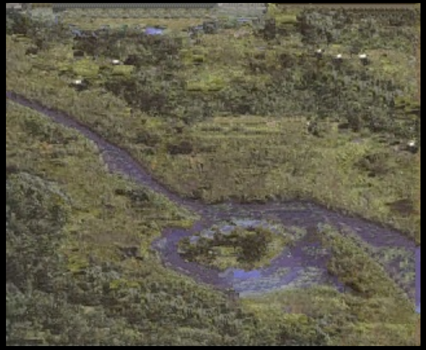
Run time: 59.8885s

Results

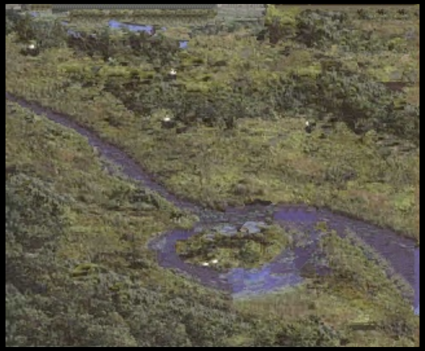
- Texture-by-numbers: Oxbow



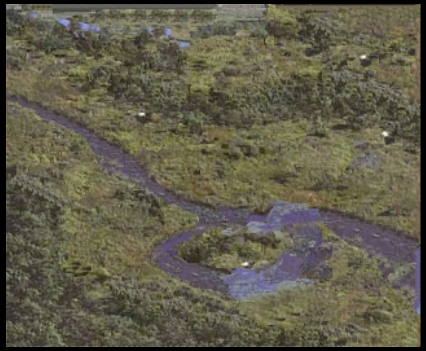
$\kappa = 0.5$



$\kappa = 2$



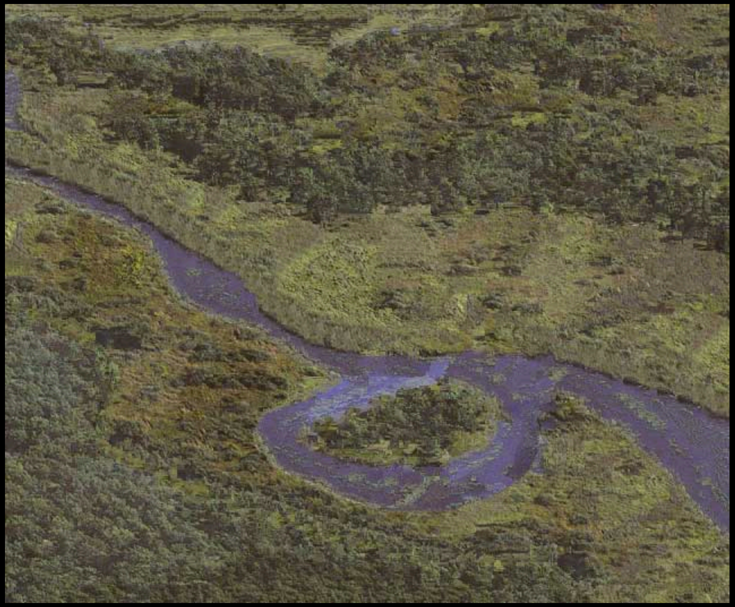
$\kappa = 5$



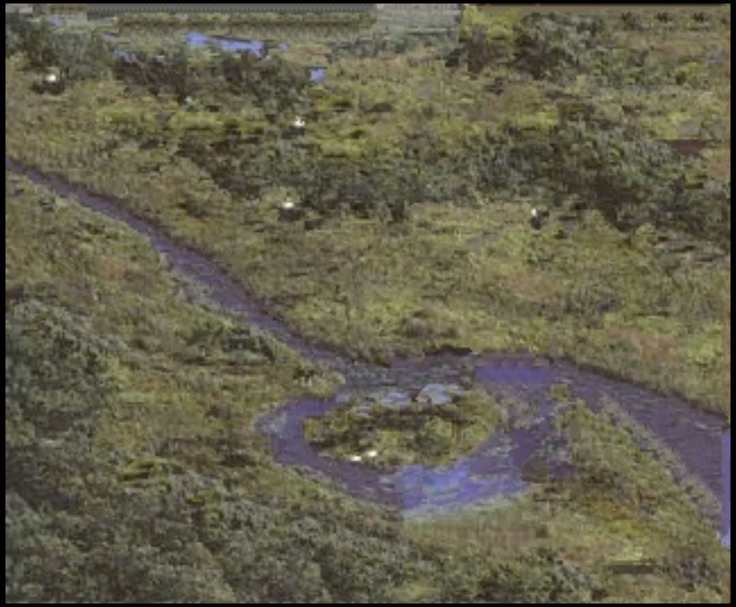
$\kappa = 10$

Results

- Texture-by-numbers: Oxbow



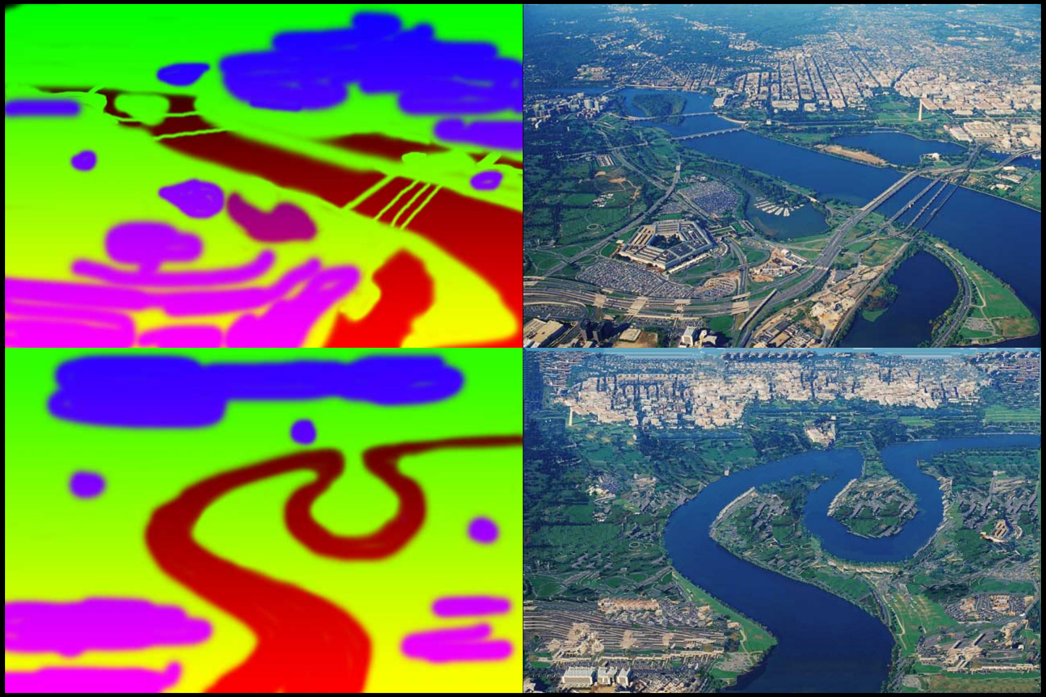
Paper result



My result

Results

- Texture-by-numbers: Potomac

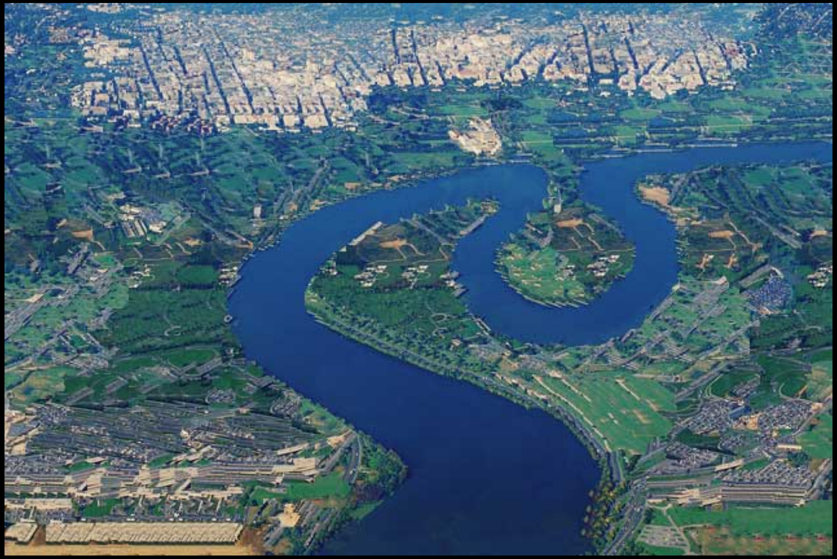


640 * 427 pixels

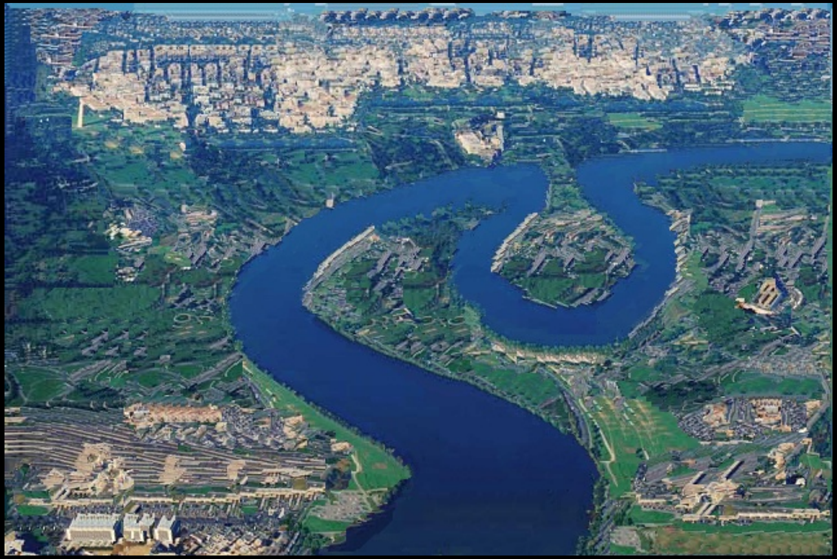
Run time: 139.3550s

Results

- Texture-by-numbers: Potomac



Paper result



My result

Future Work

- Luminance remapping
- Special processing at boundaries
- Find a general method of selecting parameter κ

Reference

- Hertzmann, Aaron, et al. "Image analogies." Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001.

- Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. Journal of the ACM, 45(6):891–923, 1998.

Thanks