Fragment-Based Image Completion

Iddo Drori   Daniel Cohen-Or   Hezy Yeshurum

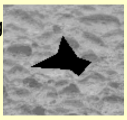
School of Computer Science
Tel Aviv University

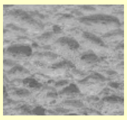
Goals

- Fill holes in images
- Remove unwanted objects from images

Previous Work

- Texture Synthesis by Non-parametric Sampling – Efros and Leung (ICCV'99)





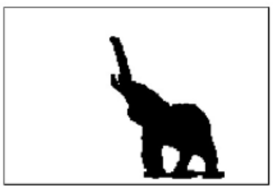
Previous Work

- Image Inpainting – Bertalmio and Caselles

Image Parts

Image

Inverse Matte



Fast Approximation

- Build an image pyramid (structure which contains images at different scales)
- Down-sample and up-sample image with a kernel at lowest scale until convergence to obtain approximation
- Use this approximation in next highest scale
- At coarser levels, the kernel affects low frequency data, whereas at levels of finer detail, the kernel will approximate higher frequencies

Fast Approximation

Multiply Image by inverse matte and add matte

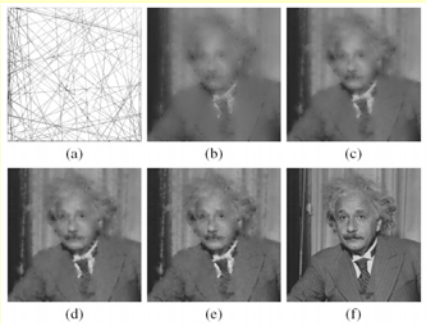
Downsample and upsample with kernel

Fast Approximation

Region after Approximation

Now use this region instead of white space for the approximation step at the next level.

Fast Approximation



Fast Approximation

Fast Approximation

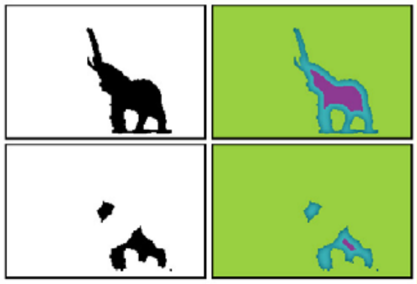
Confidence Map

- Store a confidence map for the image – each pixel gets a value between 0 and 1, with 1 being the most confident
- This tells us how confident we are in our approximation
- Regions outside the matte have confidence 1

Confidence Map

- For approximated regions, the confidence map is computed by checking the alpha value of the pixels in the neighborhood
- To select the next sub-region to fill we find the most confident pixel in our approximation and start from there

Confidence Map



Search

- For every target fragment we search for a best source match
- This is done across all translations (x,y), 8 orientations θ, and 5 scales *l*.
- Adds detail to approximated pixels created with kernel, while leaving the known pixels alone

Search

- How is this done? Consider fragments S and T
- Then we wish to minimize the function:

$$r^* = \arg\min_r \sum_{s=S_r(i),t=T(i),i\in N} (d(s,t)\beta_s\beta_t + (\beta_t - \beta_s)\beta_t).$$

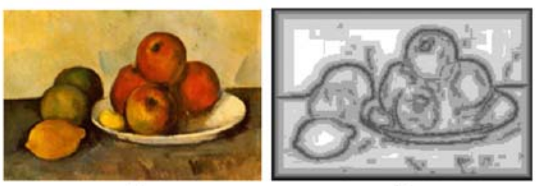
- First term penalizes different values in corresponding pixels with high confidence in both source and target fragments.
- Second term rewards pixels with a higher confidence in source than target, while penalizing pixels with lower confidence in source than target

Search

Q: But how to figure out how big of a neighborhood to use?

A: Use the contrast criterion of the absolute value of extreme values across channels.

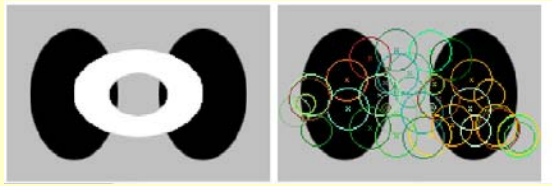
Search



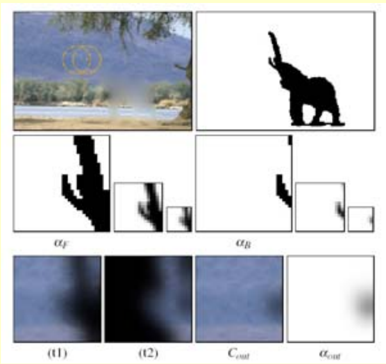
(a) (b)

Search

Search



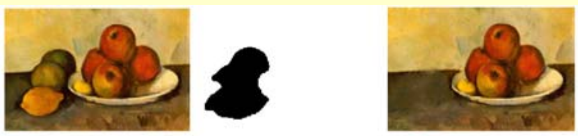
Compositing Fragments



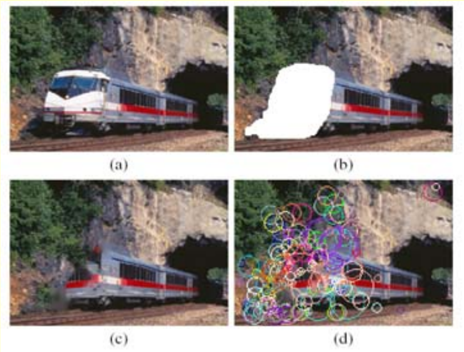
Results

Results

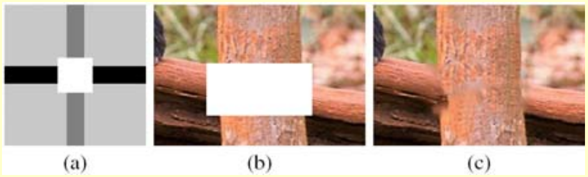
Results



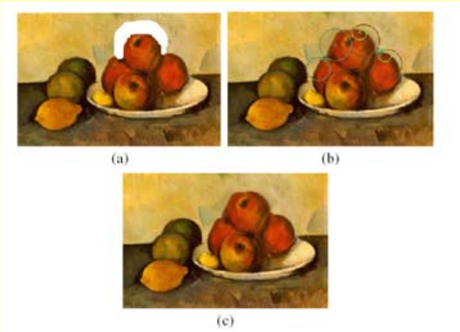
Results



Results



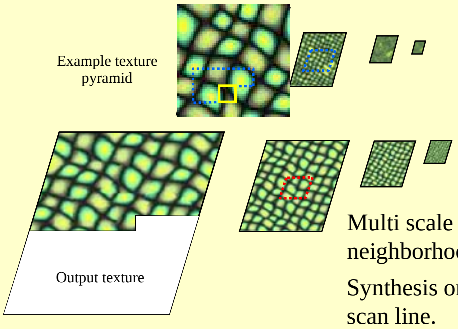
Results



Hole Filling Results*
(Why completion is important)

* With thanks to Lihi Zelnik and Yoni Wexler.

Multi-Resolution Pyramids*



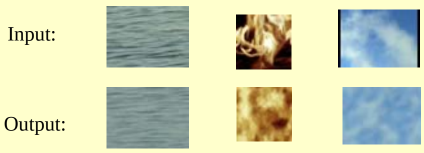
Multi scale causal neighborhood

Synthesis order - scan line.

* L.-Y.Wei, M.Levoy; "*Fast Texture Synthesis using Tree-structured Vector Quantization*"; SIGGRAPH00.

Extension to 3D Textures.

- Motion both in space and time
  - fire, smoke, ocean waves.
- How to synthesize?
  - extend 2D algorithm to 3D.



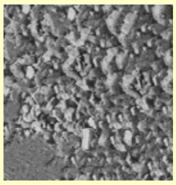
The Problems of Causal Scanning

- Scanning order:
  - Efros&Leung(1): Pixels with most neighbors.
  - Wei&Levoi(2): Raster scan.
- These are "causal" scans.

(1) A.A.Efros, T.K.Leung; "*Texture synthesis by non-parametric sampling*"; ICCV99.
(originally proposed by [Garber,'81])
(2) L.-Y.Wei, M.Levoy; "*Fast Texture Synthesis using Tree-structured Vector Quantization*"; SIGGRAPH00.

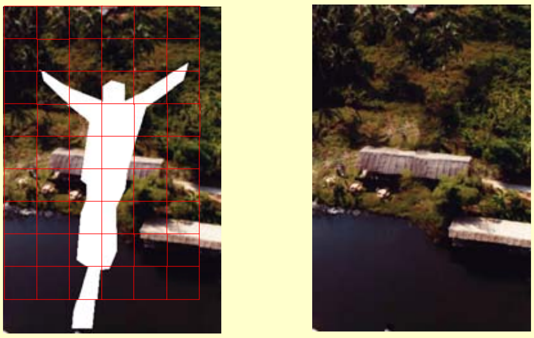
The Problems of Causal Scanning

- Can grow garbage.
- No natural means of refining synthesis.
- Cannot be parallelized.
- Problems are made worst for synthesis of 3D space-time volumes (a.k.a. video)...



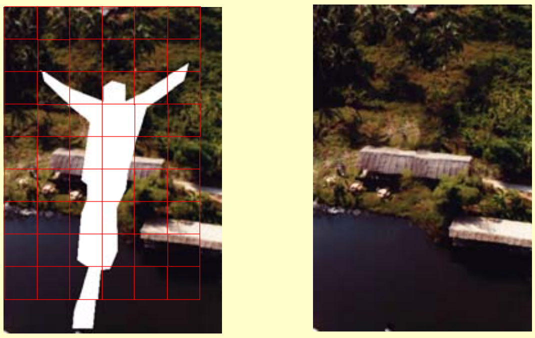
A.A.Efros, T.K.Leung; "*Texture synthesis by non-parametric sampling*"; ICCV99.

Iterative Synthesis



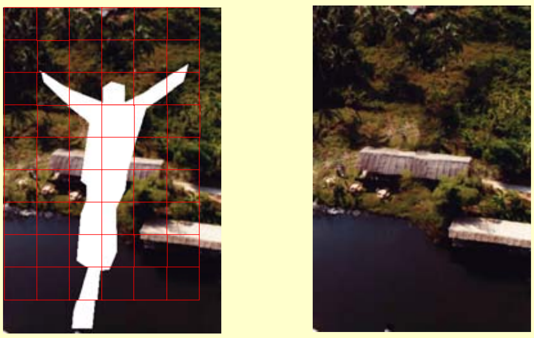
Y. Wexler, E. Shechtman, M. Irani; "Space-Time Video Completion"; CVPR'04.

Iterative Synthesis



Y. Wexler E. Shechtman M. Irani; "Space-Time Video Completion"; CVPR'04.

Iterative Synthesis



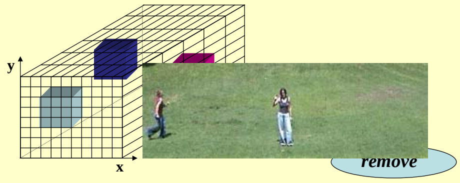
Y. Wexler E. Shechtman M. Irani; "Space-Time Video Completion"; CVPR'04.

Advantages

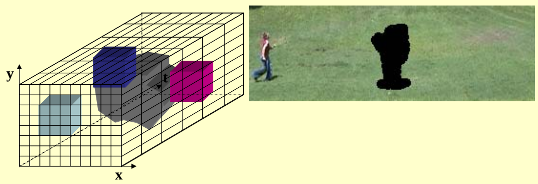
- No scan order heuristic (no garbage growing).
- Iterates until convergence to solution consistent both locally ***and*** globally.
- Parallelizable.

Multi Scale Synthesis

Sampling space-time patches

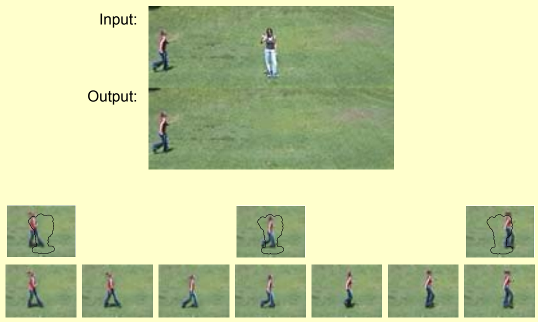


Sampling space-time patches



- Cover the “hole” with existing information
- Use small patches, and yet
- Reach globally consistent solution.

Video Results

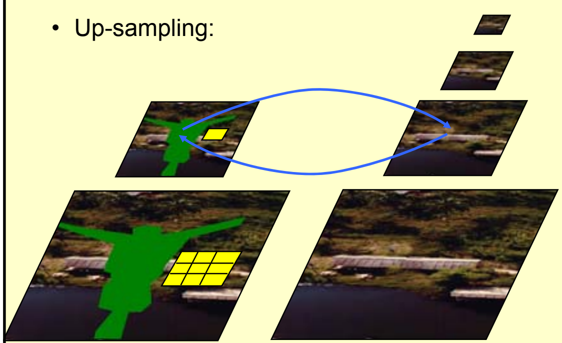


Some Technical Details

- Selecting a new value:
- Final selection by:
  - (Weighted) Mean.
  - Median.
  - (Weighted) Mean-shift.

Some Technical Details

- Up-sampling:



Video Results

Input:

Output:

Input:

Output:

Input:

Output:

Video Results

Input:

Output:

Input: Output: Original: