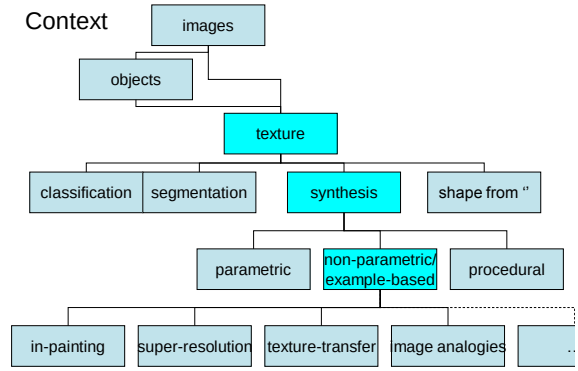
EEE6086 Video Processing and Analysis

Texture Synthesis and Applications

Ling Shao

Context



2

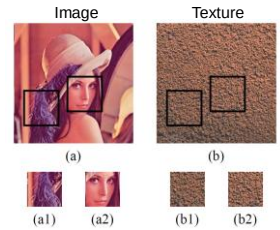
Contents

- What is texture?
- Texture synthesis
  - Intro
  - Basic technique
  - Variations, optimizations
- Applications
  - Inpainting
  - Image analogies
  - Retexturing
- What’s more?, Conclusion

3

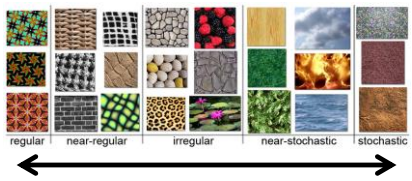
What is texture?

- If observed through a window it looks the same for every location (b), this is not the case for an image (a).
- Texture is:
  - Stationary
  - Repeating



4

What is texture?

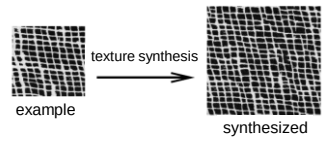


Main texture dimension,

regular/periodic/deterministic <-> stochastic/random

5

Texture synthesis defined



We have an *example* patch and from it we want to synthesize a patch of arbitrary size, typically bigger, same appearance, enough variation and spatially consistent.

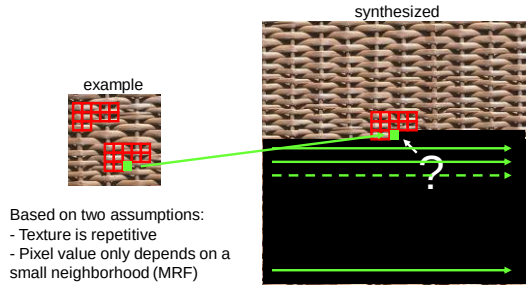
6

Texture synthesis, two main approaches

- Parametric synthesis
  - Model the texture with a fixed number of parameters
  - Stochastic models
    - AR, MA, ARMA, modeling autocorrelation
  - Compact representation
  - Fast
  - Work well on stochastic/random textures not on regular textures
- Non-parametric/example-based synthesis (*this lecture*)
  - Synthesize directly from the example, no parameterization
  - Need full example for synthesis
  - Slow(er)
  - Works well on a wide range of texture given a large enough example

7

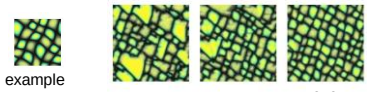
Example-based texture synthesis, basic idea [Wei02]



Based on two assumptions:
- Texture is repetitive
- Pixel value only depends on a small neighborhood (MRF)

8

Texture synthesis, neighborhood size matters

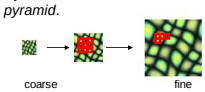


- Neighborhood size matters, too small and the algorithm is unable to capture large structure!
- However, a bigger neighborhood is more expensive, therefore, a multi-resolution variant is proposed.

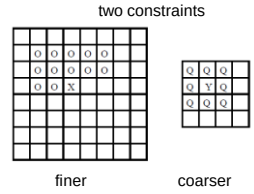
9

Texture synthesis, multi-resolution

- Synthesis in a *multi-resolution pyramid*.



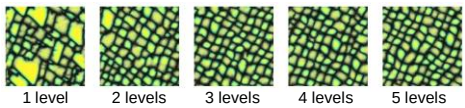
- Output pyramid is initialized with histogram equalized noise.
- Multi-resolution neighborhood
  - two constraints
- Large scale structure is captured better.



10

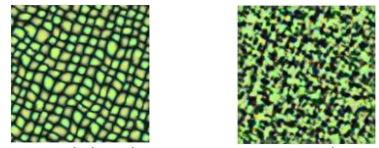
Multi-resolution results

For a 5x5 + 3x3 neighborhood:



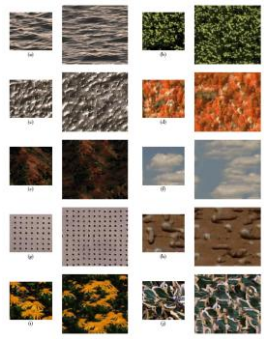
11

Comparison with parametric texture synthesis



12

Multi-resolution synthesis results



13

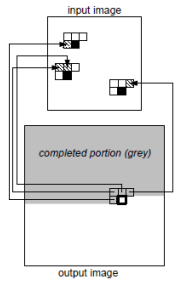
Texture synthesis, variations, optimizations

- Coherency search
- Patch based
  - Not pixel-by-pixel but synthesis on patch basis
    - Fixed size
    - Graph-cut, variable size
- Vector Quantization, Tree look-up, Approximate nearest neighbor search
  - Reduce the search space, accelerate the search
- Texture transfer

14

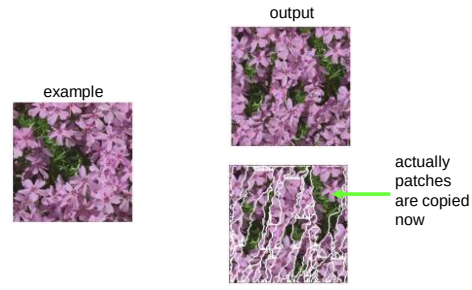
Texture synthesis optimization, coherency search

- Full-search is expensive
- We test only a *few candidates*, indicated by shifted source locations of synthesized neighborhood
- Also test a random candidate to prevent getting stuck
- 3DRS like acceleration



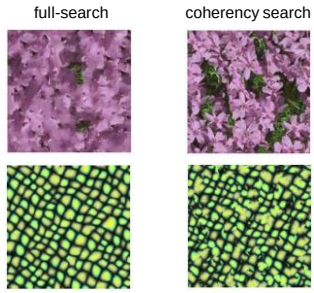
15

Coherency search, result



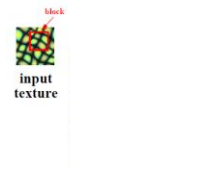
16

Coherency search, compared to full-search



17

Patch-based synthesis, fixed size



18

Patch-based synthesis, fixed size

Efros 2001

19

Patch based synthesis, graph-cut

graph-cut movie

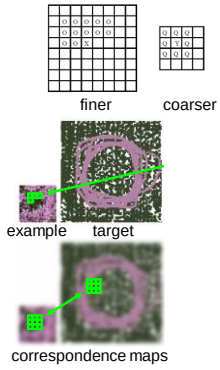
original movie source:
http://www.cc.gatech.edu/cpl/projects/graphcuttextures/

Kwatra 2003

20

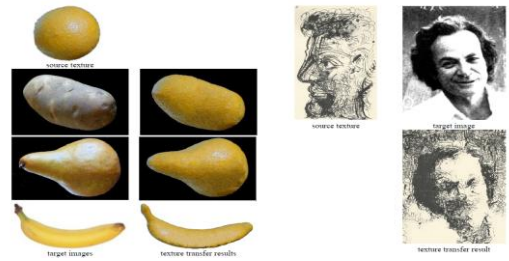
Texture transfer

- Remember the multi-resolution synthesis where a second constraint was the previous coarser scale.
- Another second constraint can be defined.
- A target image.
- This image *steers* the synthesis process.



21

Texture transfer/constrained texture synthesis



22

Applications

- Inpainting
  - *Region filling and object removal by exemplar-based image inpainting* [Criminisi04]
- Image analogies
  - *Image analogies* [Herzmann01]
- Retexturing
  - *Detail preserving shape deformation* [Fang07]

23

Inpainting

desired result:
plausible extension of structure and texture

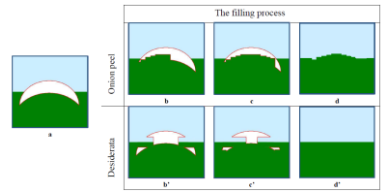
- Two main approaches:
  - *Texture synthesis* works well for texture, not for structures
  - Techniques like *diffusion* work well for structure, not for texture

24

Inpainting with texture synthesis

25

Inpainting, order is important

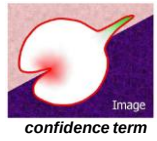


- We can use texture synthesis, however:
  - It turns out that the *order* of the synthesis is important
    - We should first extent structures and confident areas

26

Inpainting

- Give priority based on:
  - Local border shape:
    confidence term
  - Local structure:
    data term

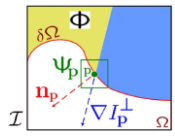


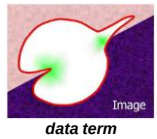
confidence term

- Priority $P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\psi_\mathbf{p}} C(\mathbf{q})}{|\psi_\mathbf{p}|}$$

$$D(\mathbf{p}) = \frac{|\nabla I_\mathbf{p}^\perp \cdot \mathbf{n}_\mathbf{p}|}{\alpha}$$

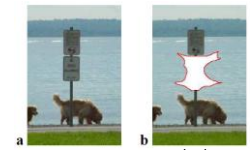




data term

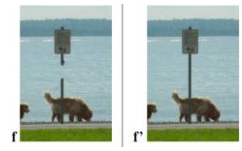
27

Inpainting, examples



movie 1
movie 2

original movie source:
http://mesh.brown.edu/dlanman/courses.html



onion peel

structure-guided

28

Image analogies

- Using example set A and A' and input image B we synthesize *B'* such that:
  - *B'*:B as A':A

Unfiltered source (*A*) Filtered source (*A'*)

Unfiltered target (*B*)

- Uses
  - multi-resolution
  - coherency search
  - 'full-search'
  - texture transfer

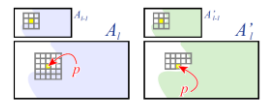
29

Image analogies

function **CreateImageAnalogy**(A, A', B):

Compute Gaussian pyramids for A, A', B

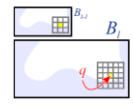
for each level l, from coarse to fine, **do**:
  for each pixel $q \in B_l'$, in scan-line order, **do**:
    $p \leftarrow$ **BestMatch**(A, A', B, B', s, l, q)
    $B_l'(q) \leftarrow A_l'(p)$
    $s_l(q) \leftarrow p$
return B_1'



function **BestMatch**(A, A', B, B', s, l, q):

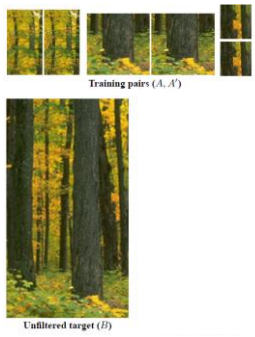
$p_{app} \leftarrow$ **BestApproximateMatch**(A, A', B, B', l, q)
$p_{coh} \leftarrow$ **BestCoherenceMatch**(A, A', B, B', s, l, q)
$d_{app} \leftarrow \|F_l(p_{app}) - F_l(q)\|^2$
$d_{coh} \leftarrow \|F_l(p_{coh}) - F_l(q)\|^2$

if $d_{coh} \le d_{app} \cdot t_l$ **then**
  return p_{coh}
else
  return p_{app}



30

Image analogies results, super-resolution

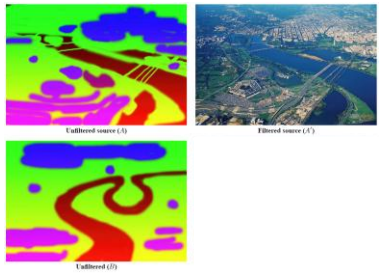


31

Image analogies results, watercolor

32

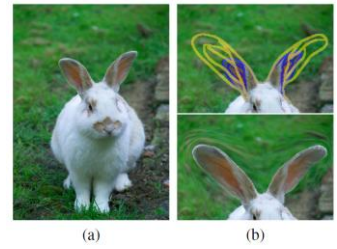
Image analogies results, texture by numbers



33

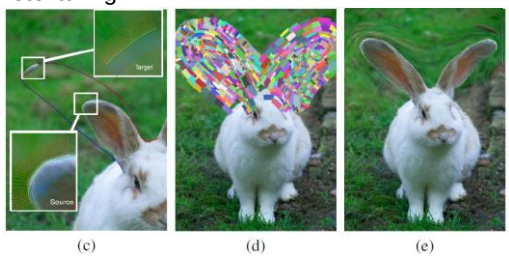
Retexturing

- Retexturing for image deformation
- Deformation can stretch/compress textures, retexturing is needed!



34

Retexturing



- Deformed (target) areas are retextured from the source locations using a patch-based technique
- The retextured image contains no texture deformations

35

Retexturing, more results

36

What's more?

- Video textures
  - By including the temporal domain
- Super-resolution
  - By using relevant high-resolution texture examples
- Compression
  - E.g. the work of Stijn de Waele, modeling texture for compression
- Image editing, retouching
- Texture compression for rendering on graphics cards
- 3D, looking behind objects

- Video epitomes, inverse texture synthesis

37

Concluding….

- Two main approaches to texture synthesis, of which example-based/non-parametric is far more popular
- Simple basic technique
- Many variations, differing in quality and speed
- Many applications
- Most applications consider images instead of videos

38

References

- "Example-based texture synthesis" part I Course SIGGRAPH 2007

- "Texture synthesis by fixed neighborhood searching" Wei 2002
- "Fast texture transfer" Ashikhmin 2003
- "Image quilting for texture synthesis and transfer" Efros 2001
- "Graphcut Textures: Image and Video Synthesis Using Graph Cuts" Kwatra 2003
- "Pyramid-based texture analysis/synthesis" Heeger 1995

- "Region filling and object removal by exemplar-based image inpainting" Criminisi 2004
- "Image analogies" Hertzmann 2001
- "Detail preserving shape deformation" Fang 2007

39

40