This lecture is based on the course materials of the Computational Photography courses given at CMU (Prof. Efros)

Computer Vision for Visual Effects

CVFX 2015

Texture Synthesis

- *Texture Synthesis by Non-Parametric Sampling*
  - Efros and Leung
  - ICCV 1999

- *Fast Texture Synthesis using Tree-Structured Vector Quantization*
  - Wei and Levoy
  - SIGGRAPH 2000

- *Synthesizing Natural Textures*
  - Ashikhmin
  - 2001 Symposium on Interactive 3D Graphics

Texture Synthesis & Transfer

- *Image Quilting for Texture Synthesis and Transfer*
  - Efros and Freeman
  - SIGGRAPH 2001
- *Image Analogies*
  - Hertzmann *et al.*
  - SIGGRAPH 2001

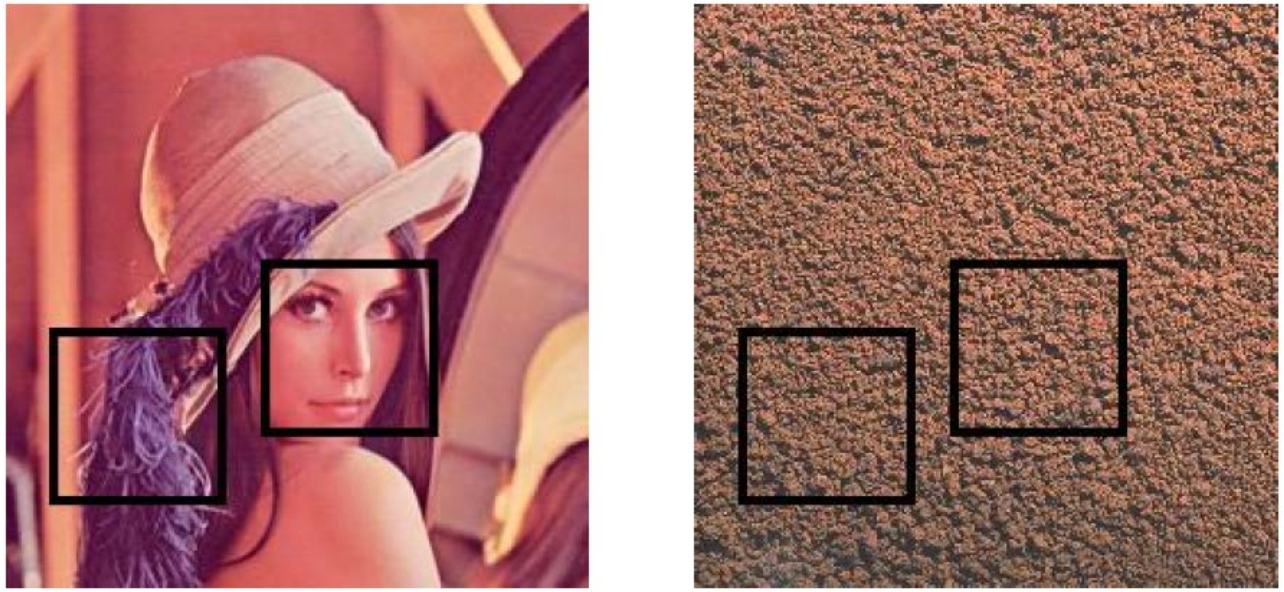
Some Questions

- What are image textures?
- What is texture synthesis?
- What is a non-parametric method?
- What is a Markov random field model?

Texture

- Spatially repeating patterns

影像與材質的不同點?



(a) (b)

(a1) (a2) (b1) (b2)

兩項性質: *stationarity* 和 *locality*

Texture Synthesis

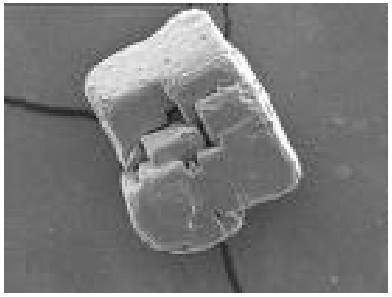
- 從一個樣板產生新的材質圖案

Parametric and Non-Parametric Sampling

- 好的方法必須要能夠描述不同性質的材質:
  - 具有結構性的材質
  - 隨機不具特定結構的材質
  - 或者介於兩者之間

以 “Text” Synthesis 為例對照

- Markov chain
  - One dimensional
  - Given the present state, the future and past states are independent
- *n*-gram: encode *n* words into a state
- Sample text, e.g., a book
- Probability tables
  - *I* → *spent*
  - *spent* → *an*
  - *an* → *interesting*
  - …



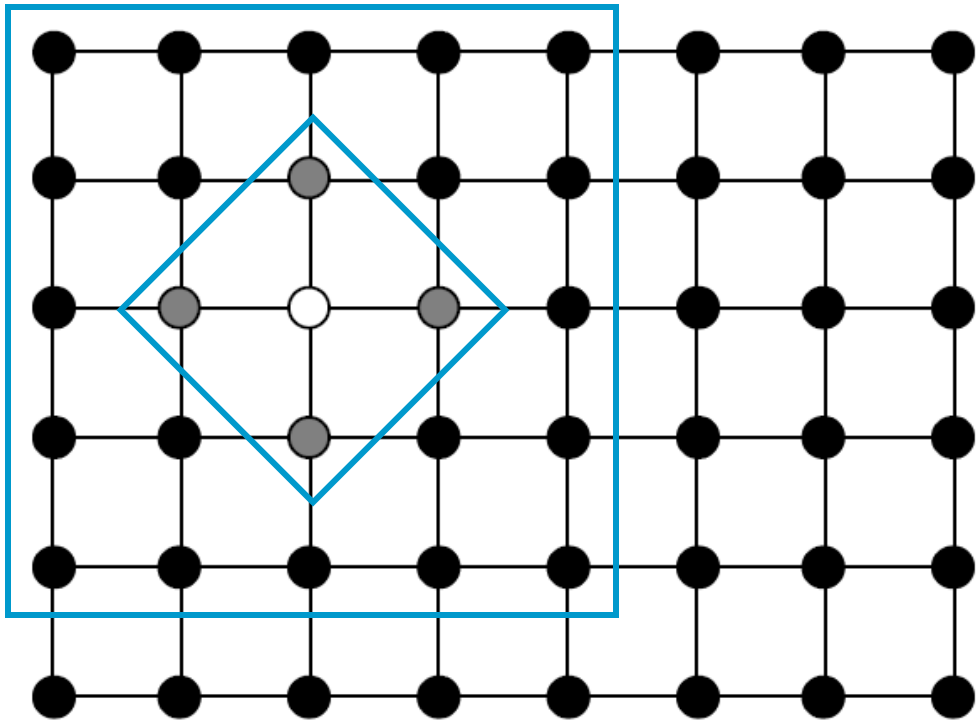
"I spent an interesting evening recently with a grain of salt"

Extension to Two Dimensions

- How to define a unit (a letter or a word) of synthesis and its context (*n*-gram) for texture?
- How to construct a transition probability?
- How to linearize the synthesis process in 2D?

Markov Random Fields and Images

- The probability distribution of brightness values for a pixel given the brightness values of its spatial neighborhood is independent of the rest of the image

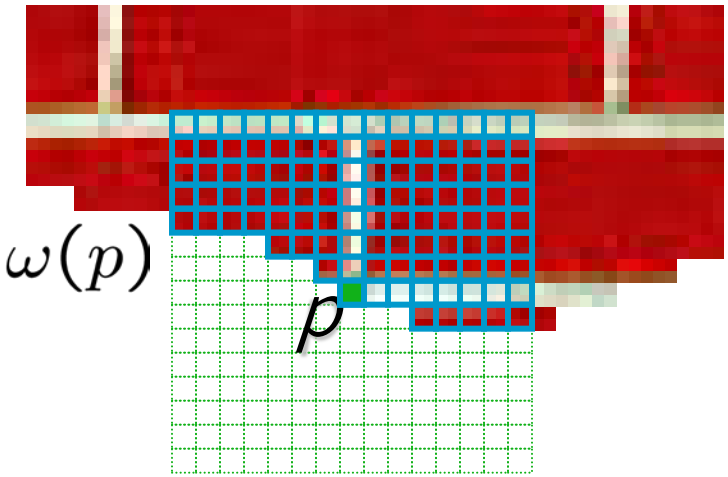


Texture Synthesis

Text	→	Texture
Markov chains	→	Markov random fields
N-gram	→	Square window around a pixel
Sample text	→	Sample texture
Exact match	→	Approximate match

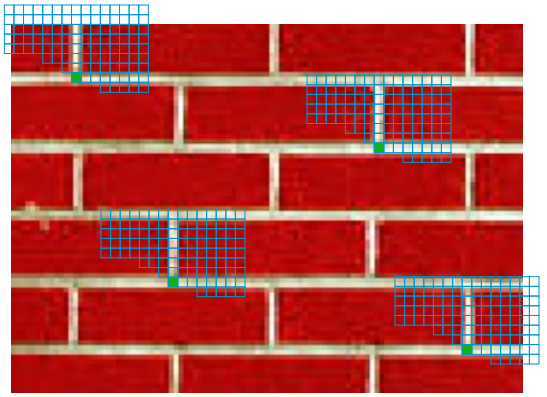
- Probability tables
  - Hard to construct them explicitly
  - Through non-parametric sampling

Efros & Leung Algorithm



Synthesizing a pixel

non-parametric sampling



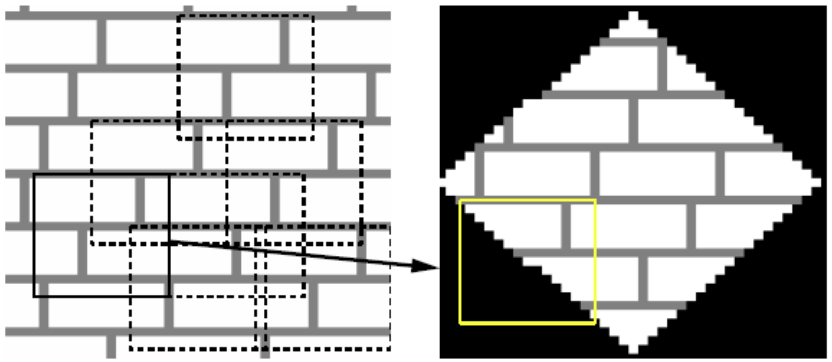
Input image

- Assuming Markov property, compute $P(p|\omega(p))$
- Building explicit probability tables is infeasible
  - Instead, *search the input image* for all similar neighborhoods — *p.d.f.* for *p*
  - To sample from this *p.d.f.* , just pick one match at random

Efros's slide

合成的過程

- 由內而外一層層長出來
  - 在每一層都由最多 neighborhood 的 pixel 優先 (可靠度最高)



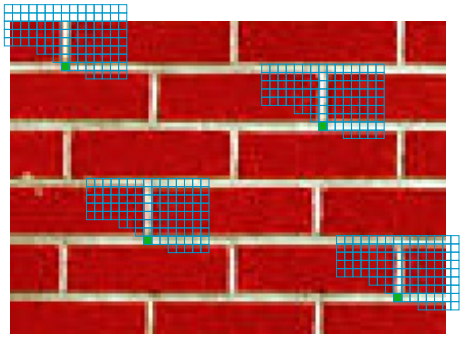
Create a Pool and Get a Histogram

- The pool of patches

$$\{\omega \subset I_{sample} : d(\omega(p), \omega) < (1+\epsilon) d(\omega(p), \omega_{best})\}$$

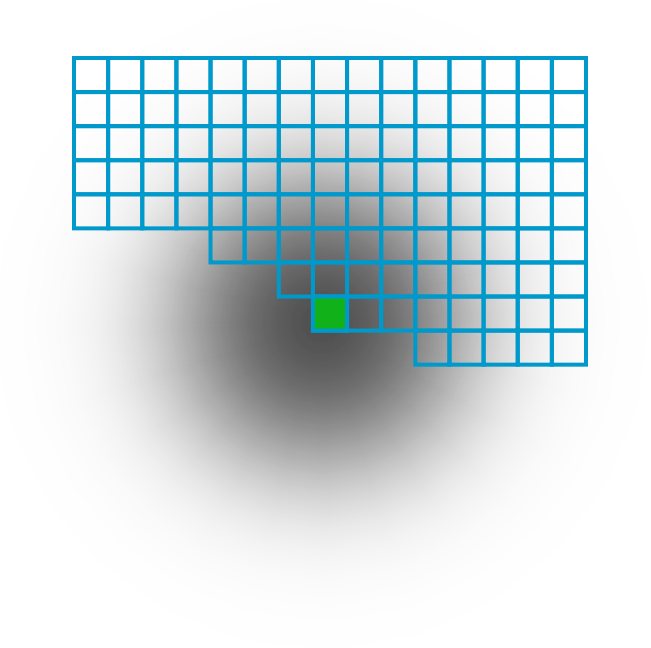
$$\omega_{best} = \arg\min_{\omega} d(\omega(p), \omega) \subset I_{sample}$$

- The center pixel values of patches in the pool give us a histogram for *p*, which can then be sampled, either uniformly or weighted by *d*

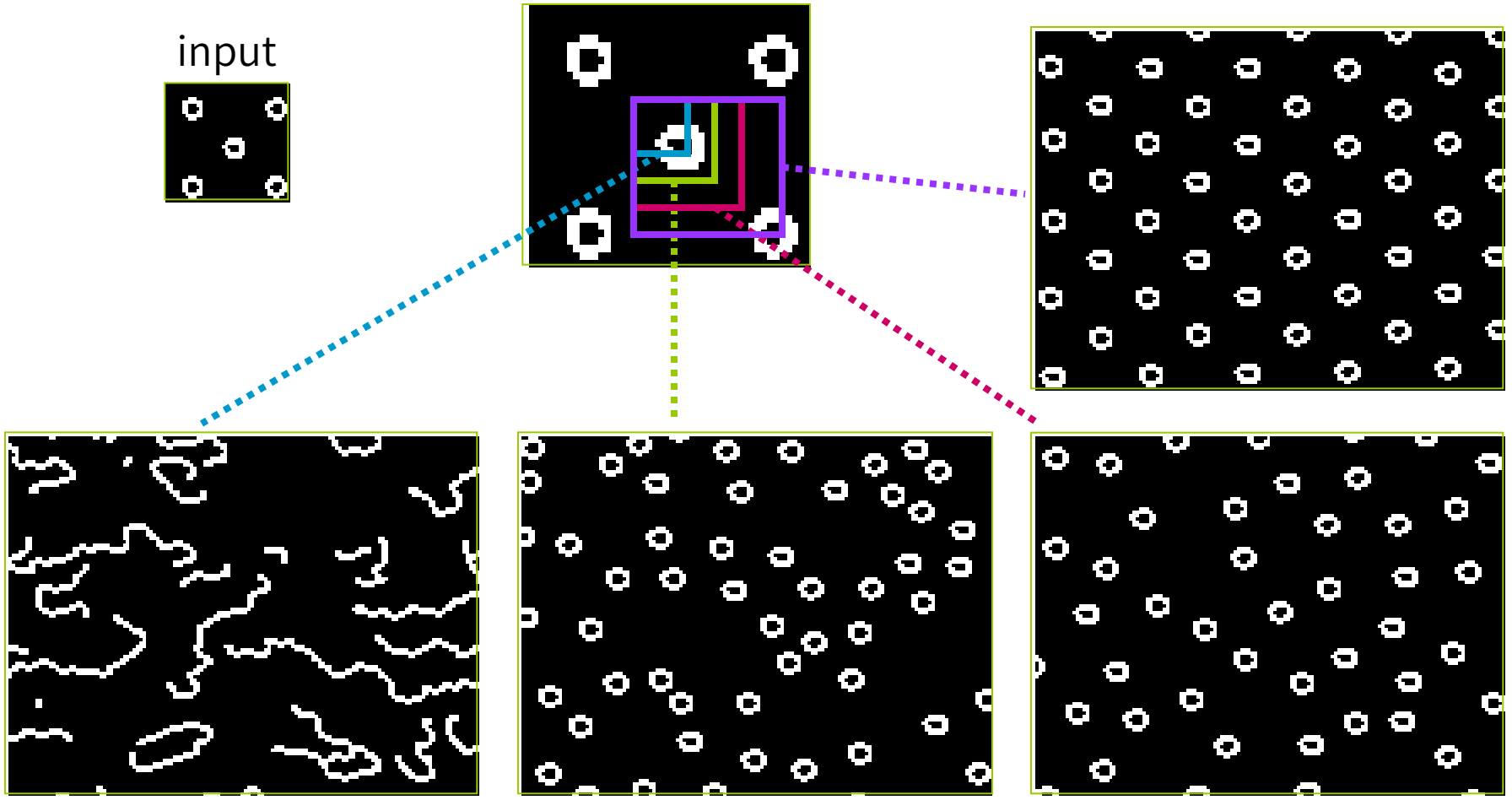


權重

- *Gaussian-weighted* SSD
  - 靠近中間的像素提供的資訊較可靠

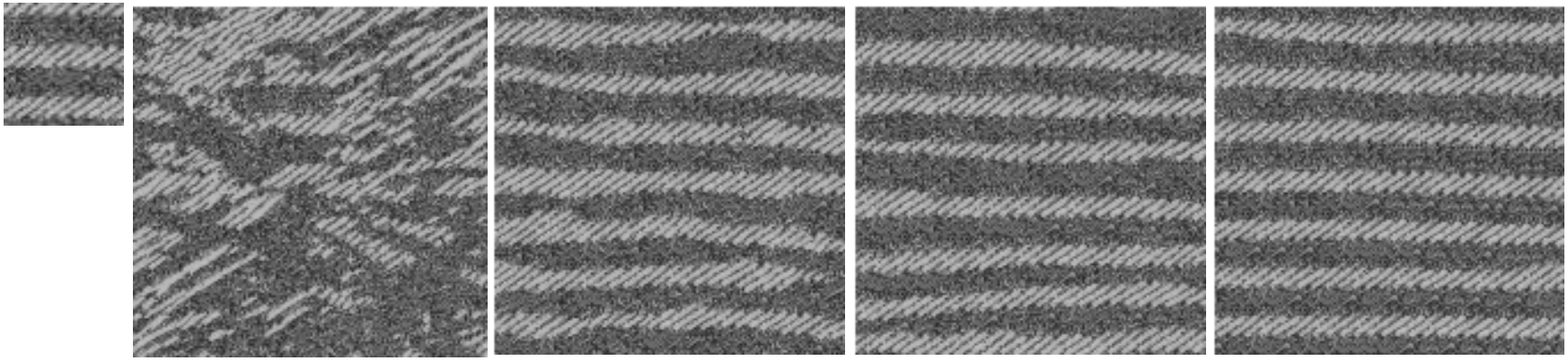


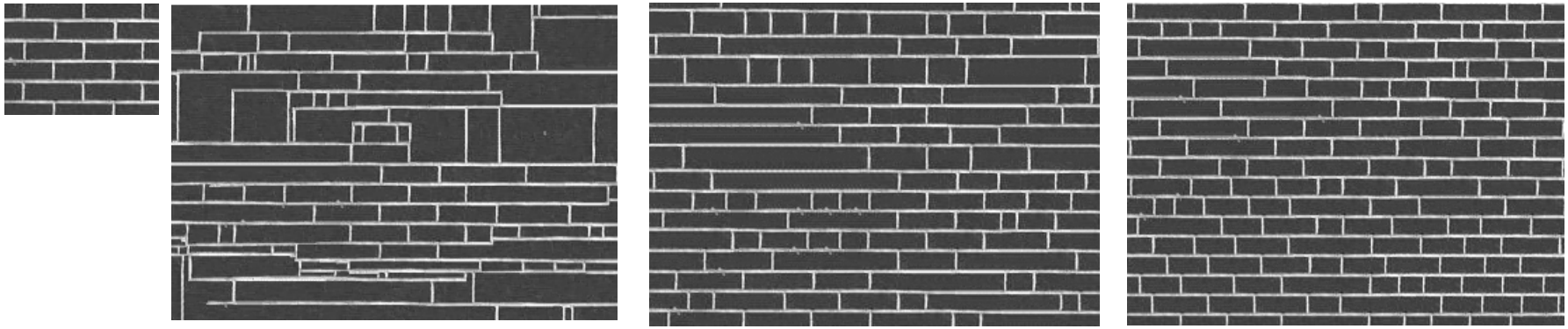
Neighborhood Window



Efros's slide

Window Size

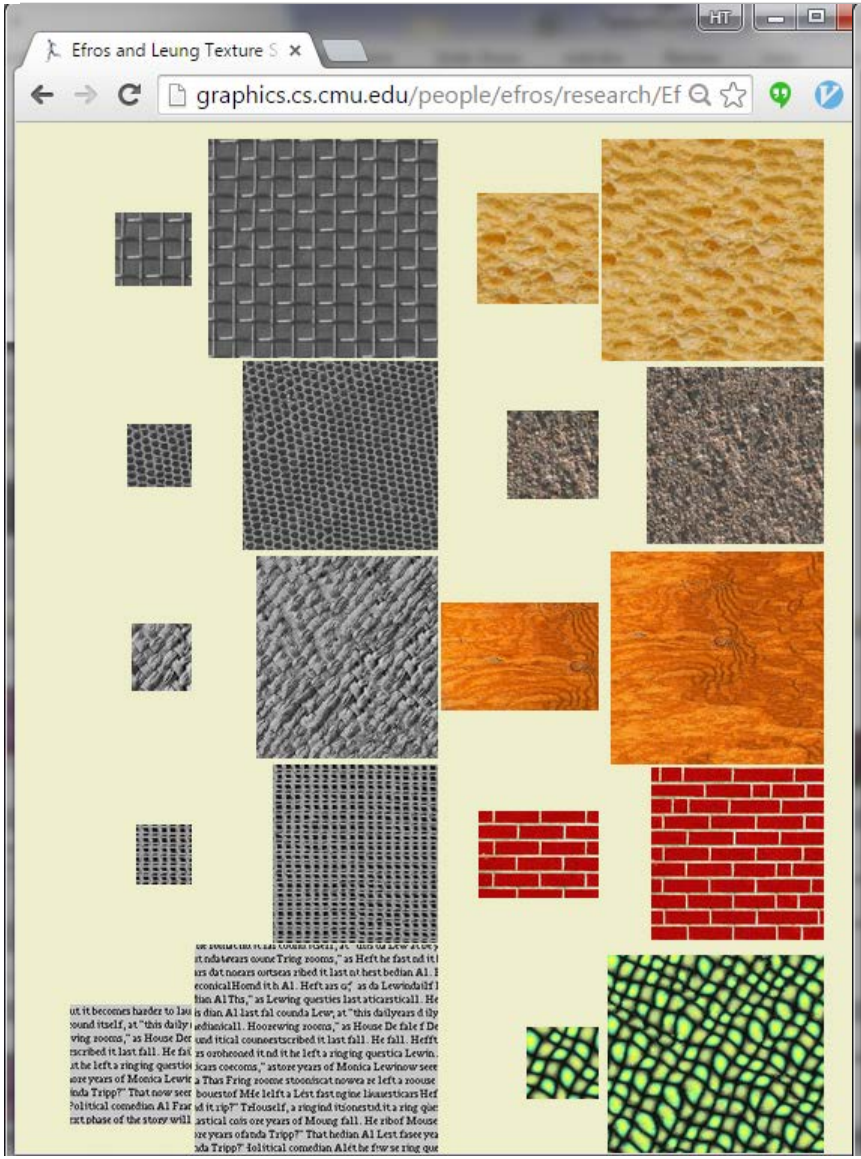


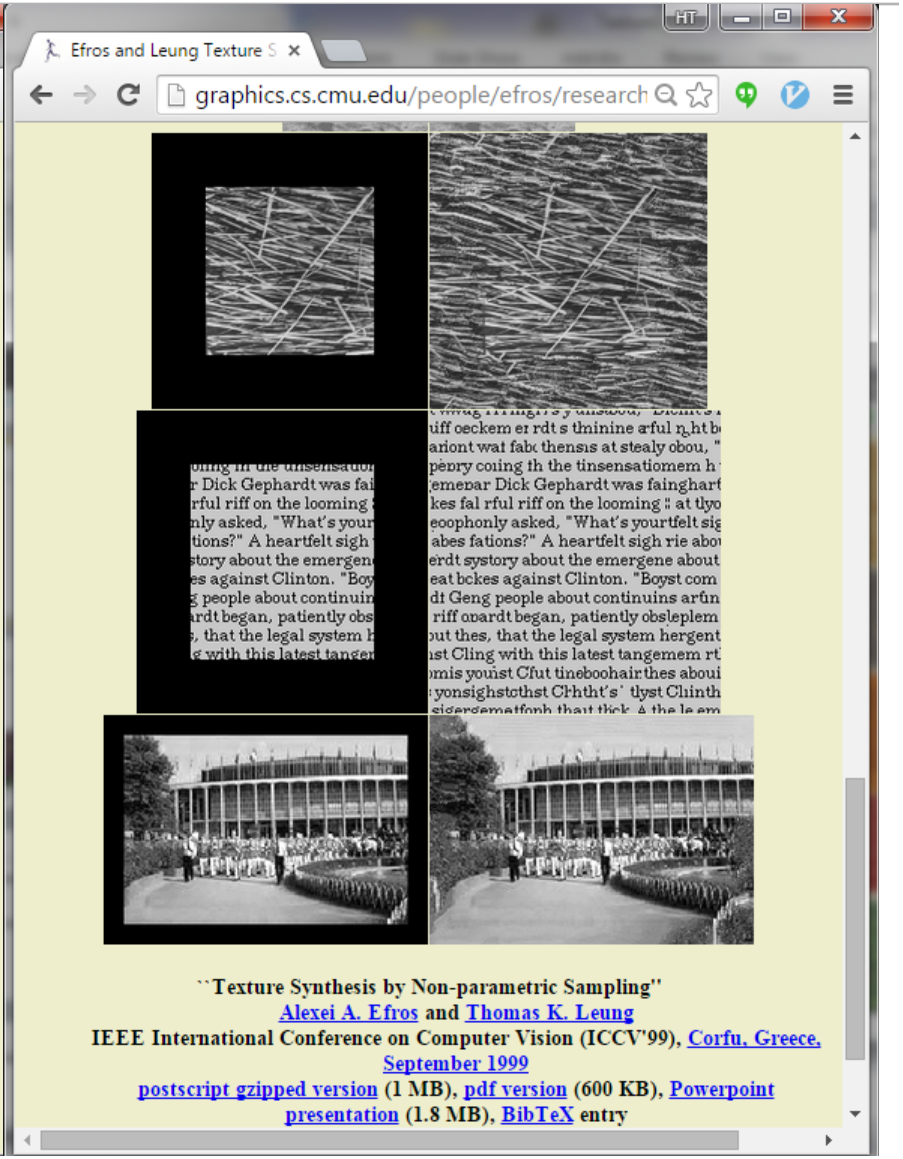


increasing window size

Efros's slide

Results from Efros & Leung Project Webpage





The Efros & Leung algorithm

- Non-parametric sampling
- Results are good
- Limitations:
  - Slow
  - Frontal parallel?

non-stationary

Conclusion

- Markov random field model
  - Seems to be a good model for image textures

- Two important issues
  - Computation time
  - How to incorporate more complicated structures into it

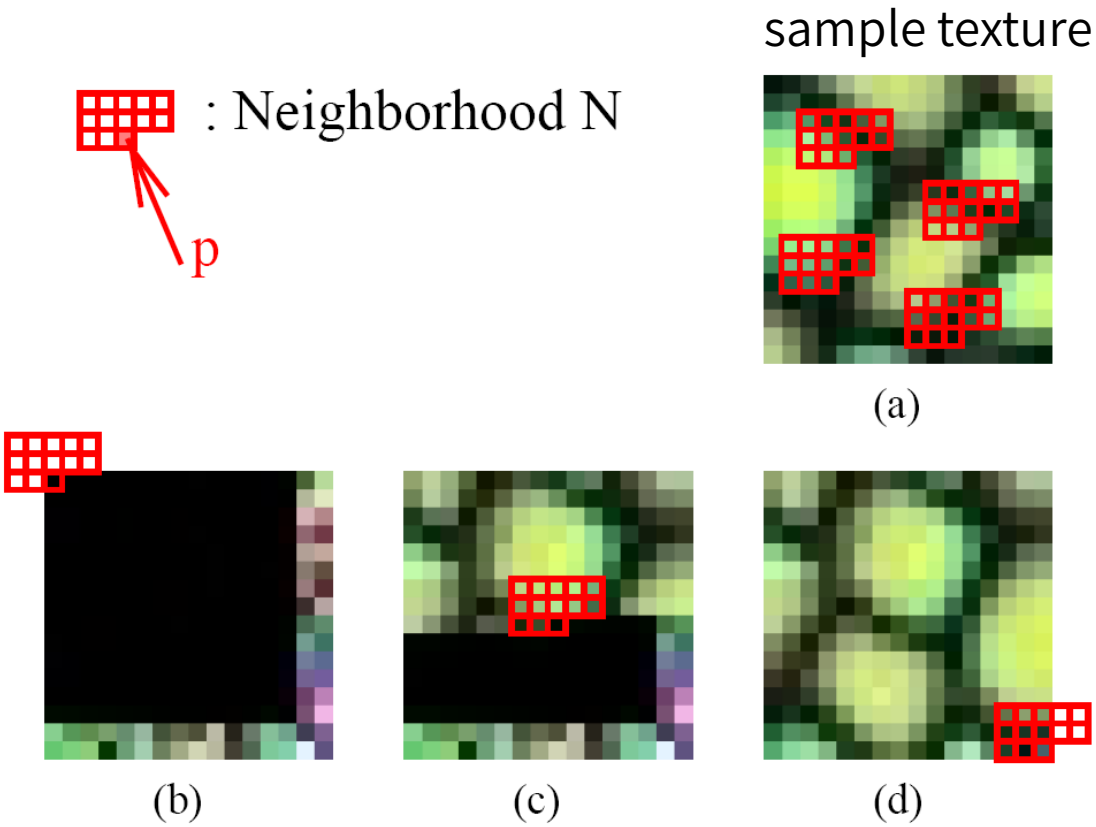
Related Work

- *Fast Texture Synthesis using Tree-Structured Vector Quantization*
  - Wei and Levoy
  - SIGGRAPH 2000

- *Synthesizing Natural Textures*
  - Ashikhmin
  - 2001 Symposium on Interactive 3D Graphics

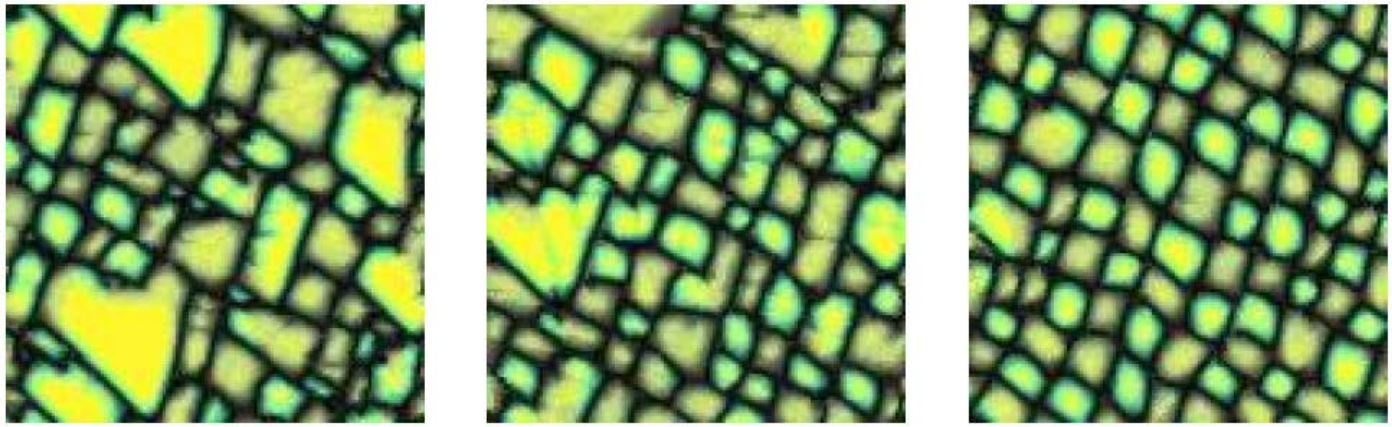
How to Accelerate

- Wei & Levoy
- Raster scan ordering
- L_2 norm



Neighborhood Sizes

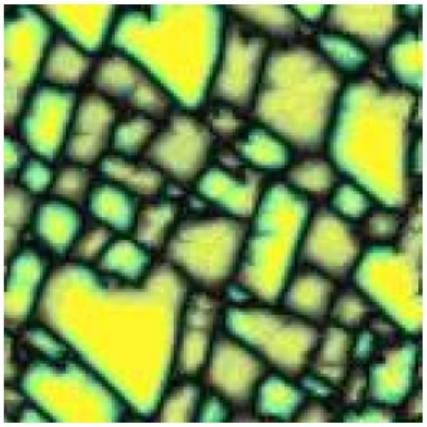
- Synthesis results with different neighborhood sizes



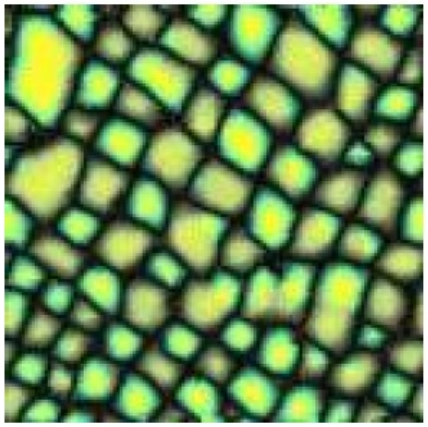
5x5 7x7 9x9

Multiresolution Synhesis –from Low to High

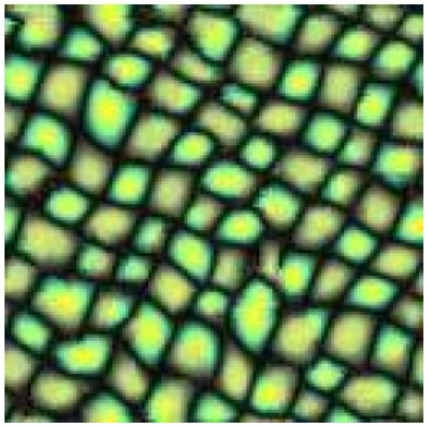
same neighborhood, different numbers of pyramid levels



1 level

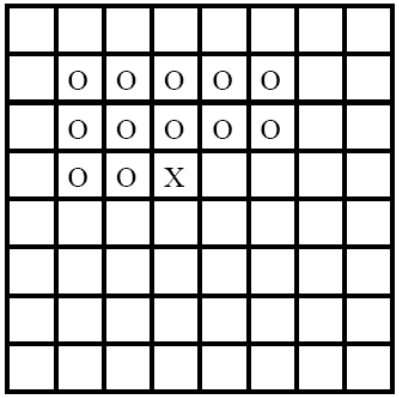


2 levels

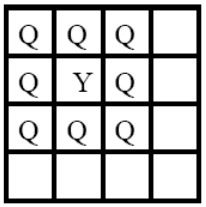


3 levels

Gaussian pyramid: blur and downscale



L

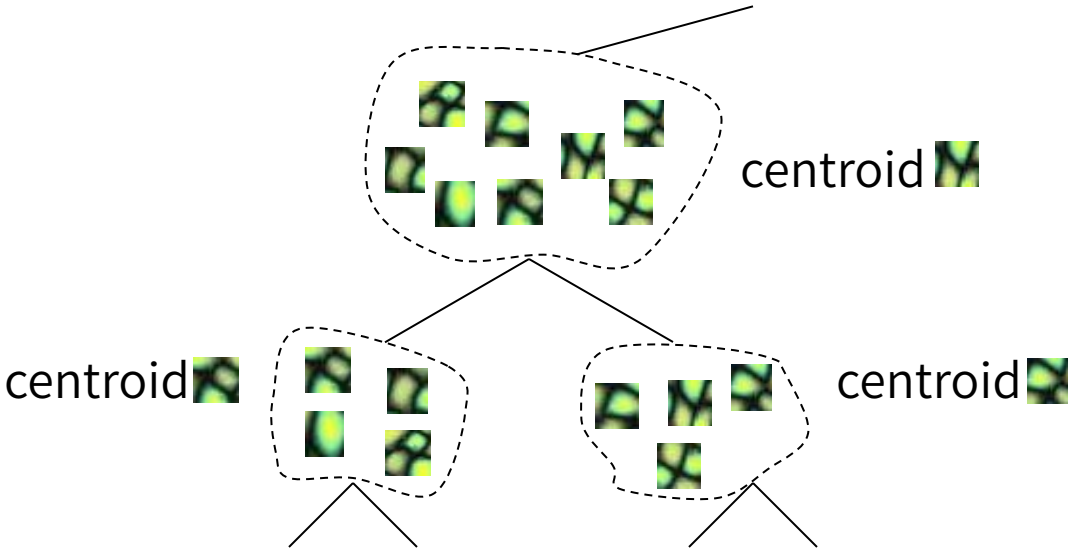


L+1

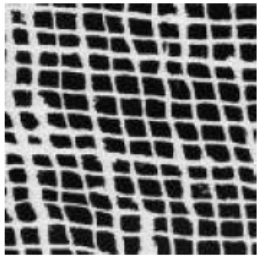
use full neighborhood at low-resolution level

Further Acceleration

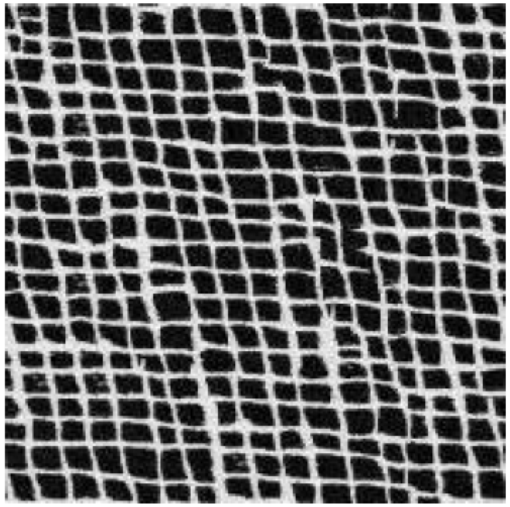
- Find an approximate solution to the nearest-point searching problem: Tree-Structured Vector Quantization (TSVQ)
  - Build a binary-tree-structured codebook
  - Synthesis: best-first traversal
  - Nearest point: centroid at the reached leaf node

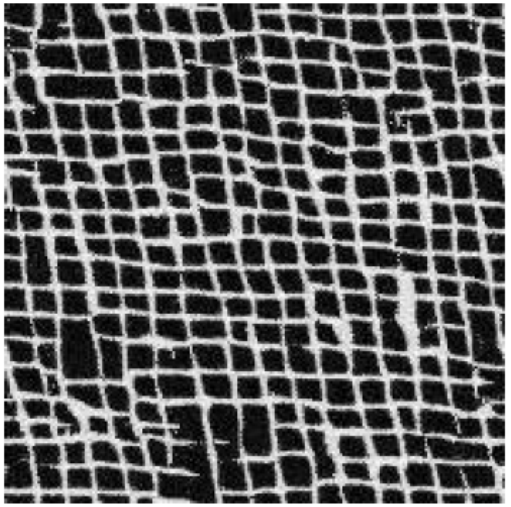


Synthesis Using TSVQ



(a) D103

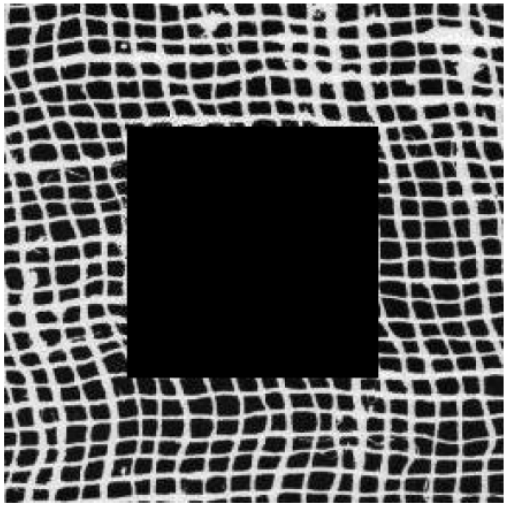


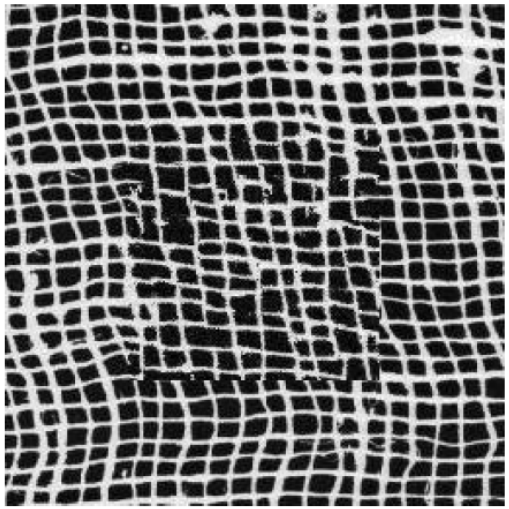


Algorithm	Training Time	Synthesis Time
Efros and Leung	none	1941 seconds
Exhaustive Searching	none	503 seconds
TSVQ acceleration	12 seconds	12 seconds

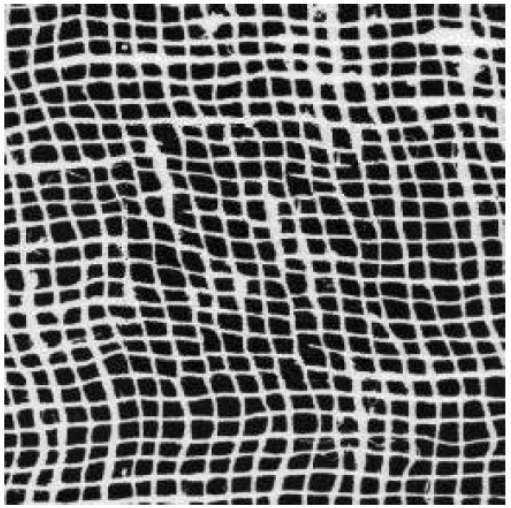
[Wei & Levoy], on a 195MHz R10000 processor

Some Problems





synthesis result using Wei & Levoy algorithm



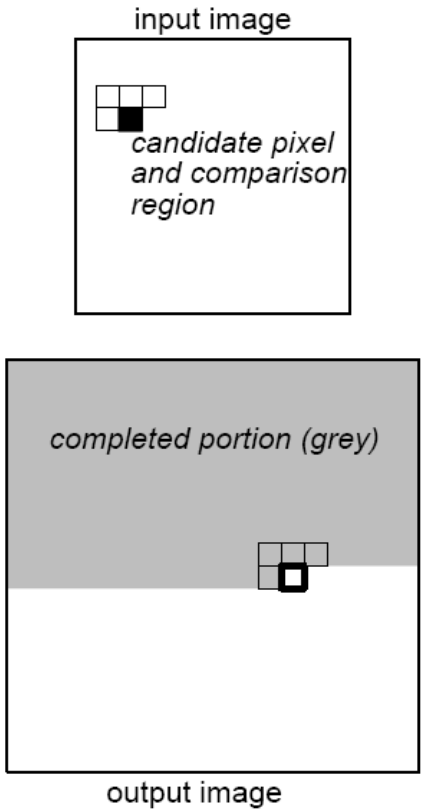
synthesis result using 2-pass Wei & Levoy algorithm plus spiral synthesis ordering

Blur Out Finer Details

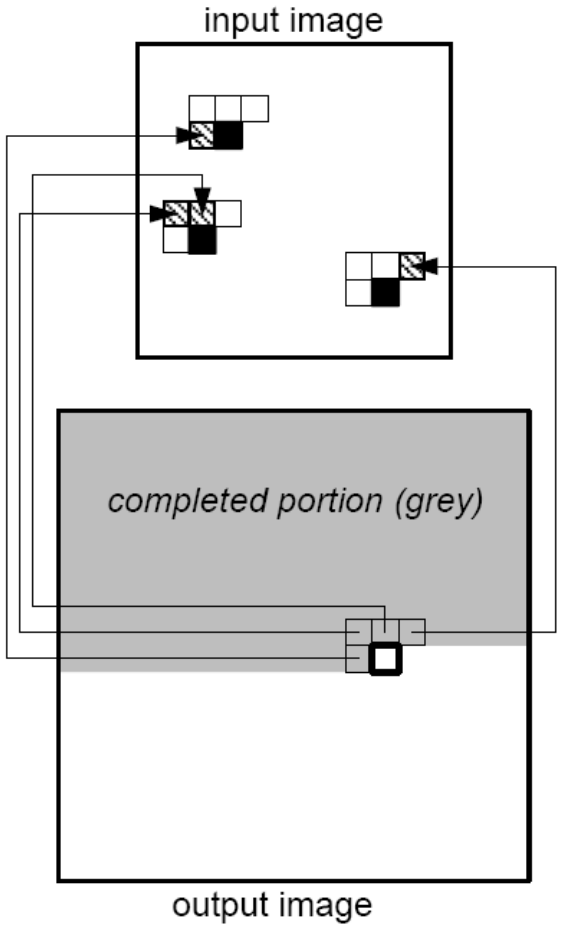
Wei & Levoy algorithm

Sometimes We Do Need Verbatim Copy

Wei & Levoy

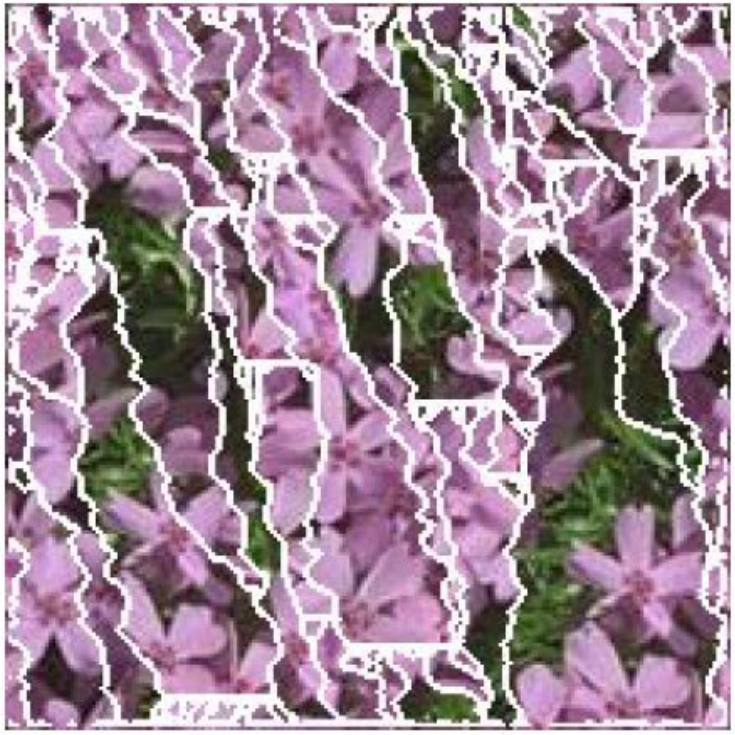


Ashikhmin

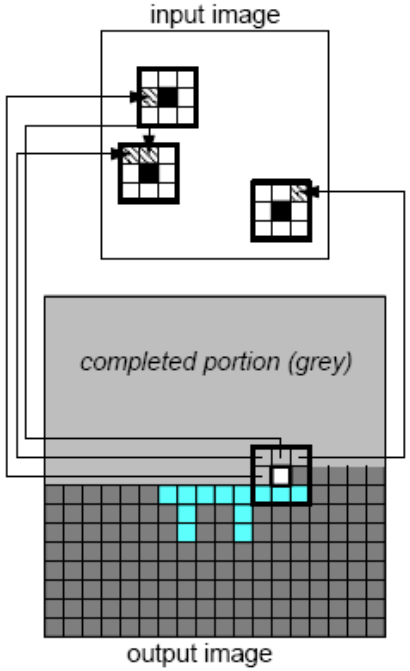


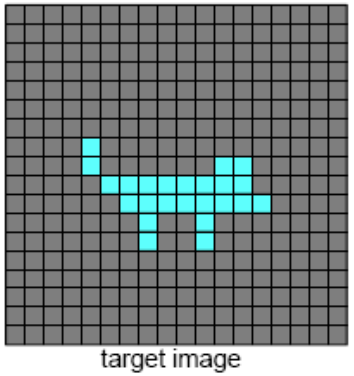
Ashikhmin's algorithm

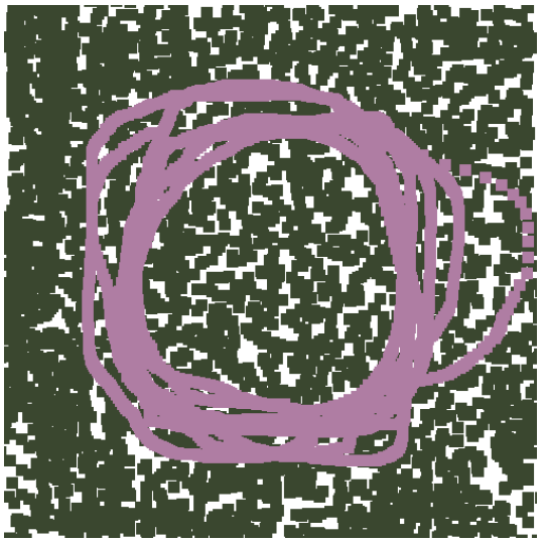
Region-growing Nature of Ashikhmin's Algorithm



Ashikhmin's algorithm with user control







Conclusion

- Markov random field model
  - Seems to be a good model for image textures
- Two important issues
  - Computation time
  - How to incorporate more complicated structures into it

Further Reading

- Graphcut Textures
  - Efficient
  - Visually pleasing results

Graphcut Textures: Image and Video Synthesis Using Graph Cuts

Vivek Kwatra Arno Schödl Irfan Essa Greg Turk Aaron Bobick

GVU Center / College of Computing
Georgia Institute of Technology
http://www.cc.gatech.edu/cpl/projects/graphcuttextures

This banner was generated by merging the source images in Figure 6 using our interactive texture merging technique.