CS4243 Project
Texture Synthesis

Zhuo Shaojie & Guo Dong

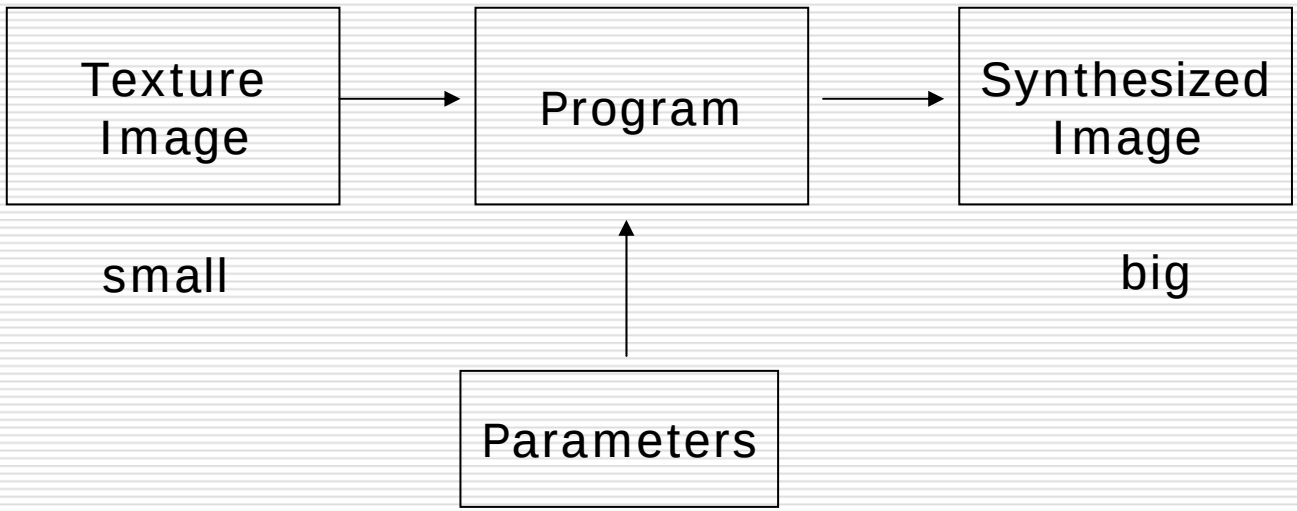
Outline

- Introduction
- Algorithm
  - Find suitable samples
  - Find a cut
  - Feature Map
  - Additional topic
- Results
- Conclusion

Reference

- Image Quilting
  - Efros and Freeman ACM SIGGRAPH 2001
- Graphcut Textures
  - Kwatra et. al. ACM SIGGRAPH 2003
- Feature Matching
  - Qing Wu et. al. ACM SIGGRAPH 2004

Texture Synthesis Problem

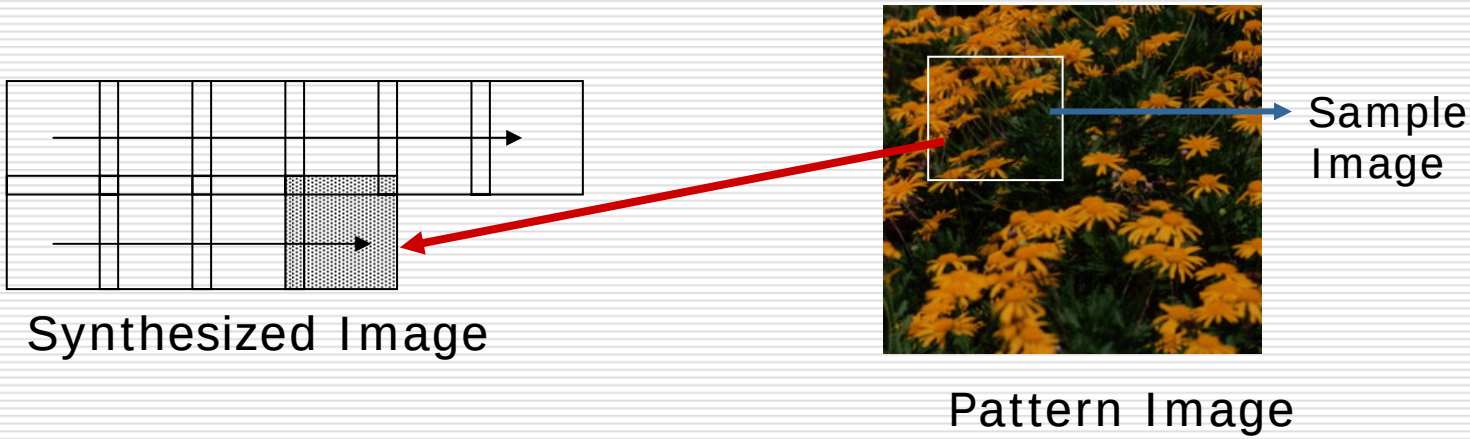


Goals

- Result image should
  - be seamless
  - look similar to input
  - not be obviously regularly repeated

Algorithm

- Synthesize in raster scan order



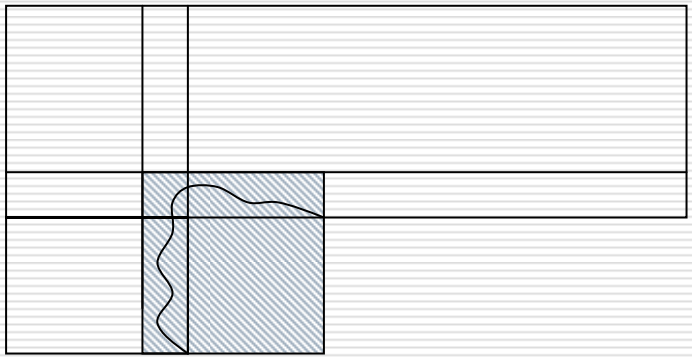
- For each block, find a suitable sample from pattern image

Note that: usually sample image is smaller than pattern image

The left figure is from Efros's slides at ACM *SIGGRAPH*2001
The flower figure (right) is from VisTex

Algorithm (cont.)

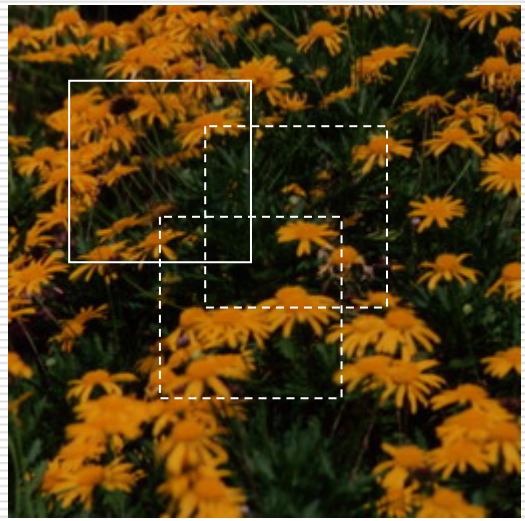
- Find a cut in the overlapping region



- Paste the new sample image
- Repeat

Find suitable samples

- Slide a window over the entire pattern image
- For each position, compute SSD(Sum of Squared Differece) of pixels in overlapping region
- Randomly pick one from those whose SSD<MinSSD*(1+a)



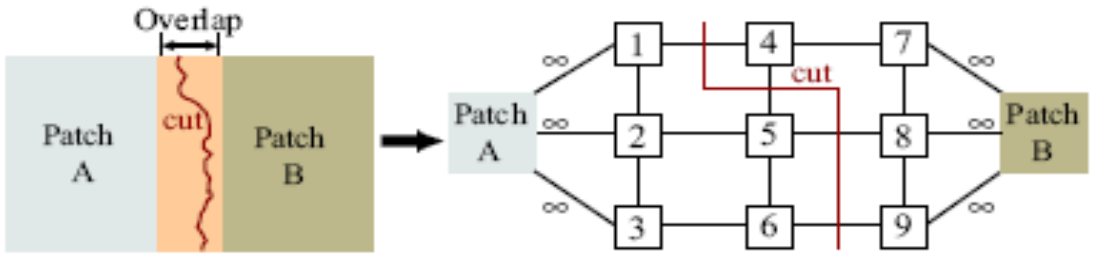
This flower figure is from VisTex

Find suitable samples (cont.)

- 2 Problems:
  - It takes a long time to search all position
  - If randomly pick one, there will be mismatch.
- Our solution:
  - Do not search all; search a part instead
    - Use a step in search
      - Can add a random offset between two step
    - Get an approximate minSSD
  - After randomly picking, search positions around to find a position with partial minimal SSD

Find a cut

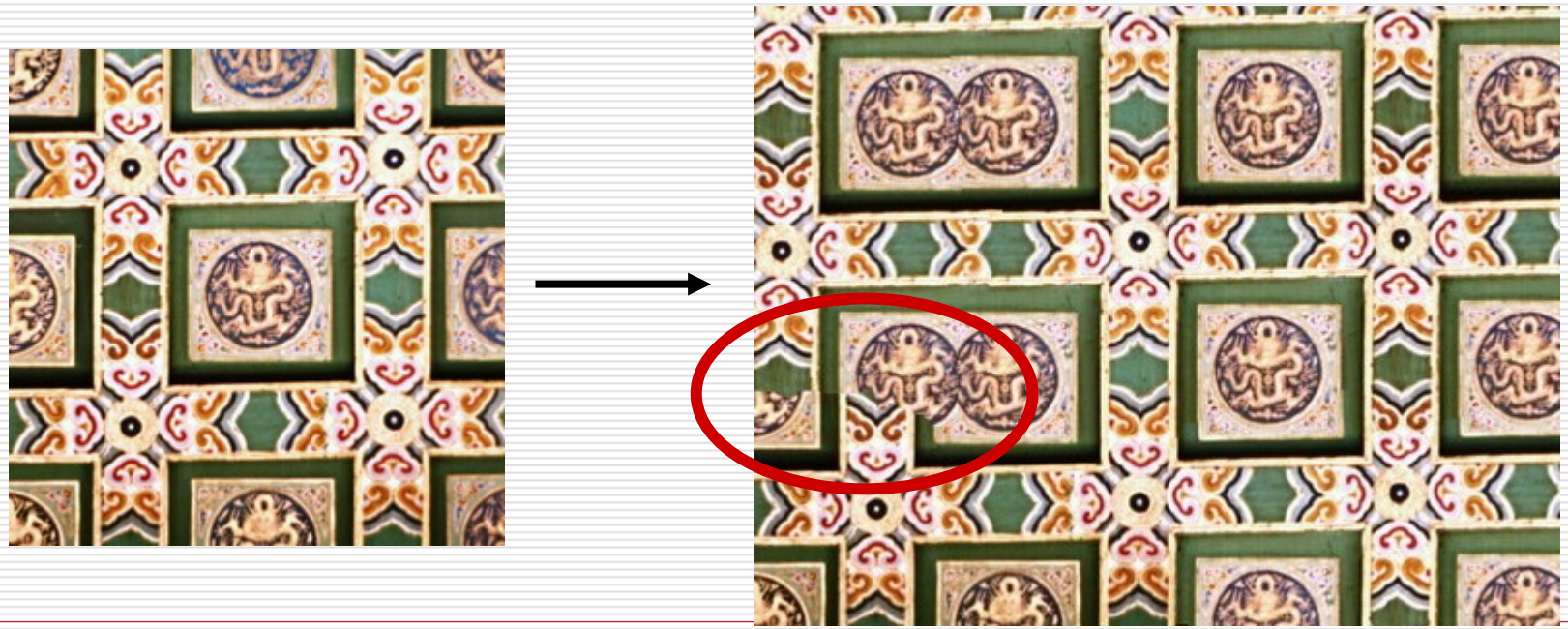
- Use graph cut
  - Construct a graph of all pixels in overlapping region
  - The cost between two adjacent pixels s,t
    cost=||A(s)-B(s)||+||A(t)-B(t)||
    where A(.),B(.) denote old and new image,respectively
  - Run max-flow/min-cut algorithm to find minimal cut



This figure are from [Kwatra2003]
The max-flow/min-cut algorithm code is from Prof. Ramin Zabih

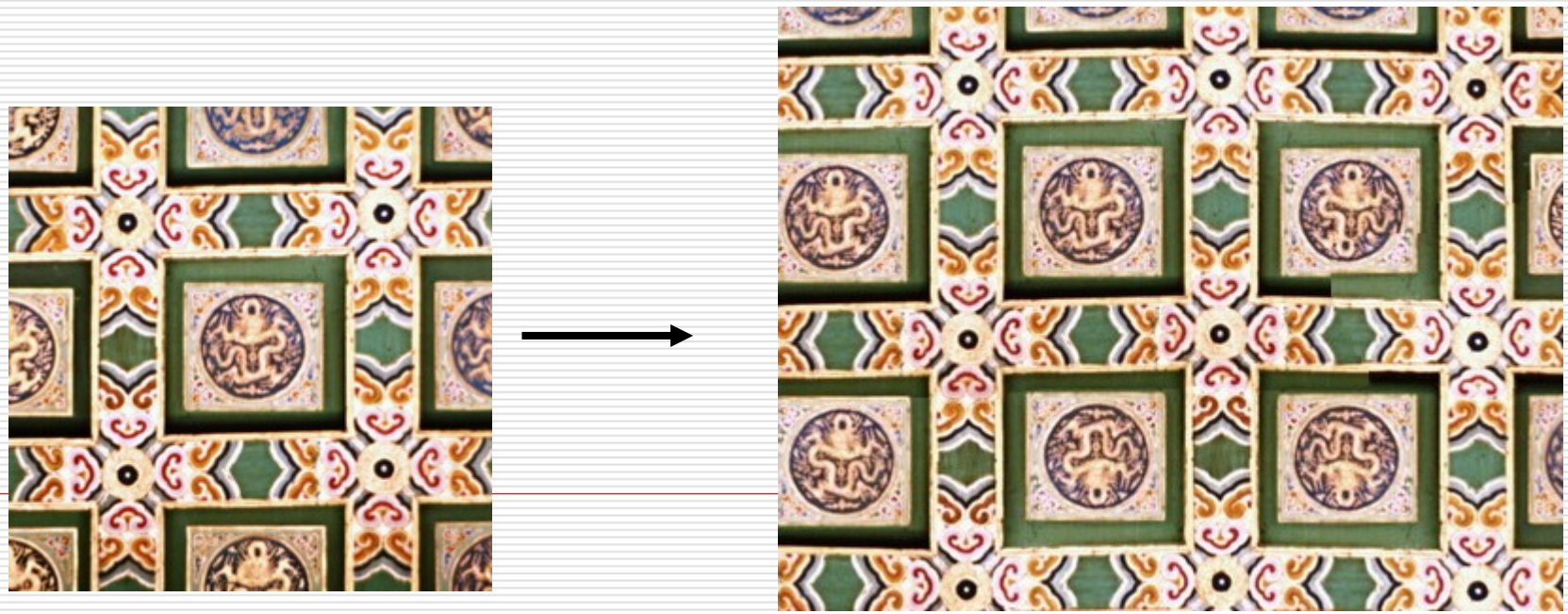
Feature Map

- Some images have strong structure
- The result image has mismatches



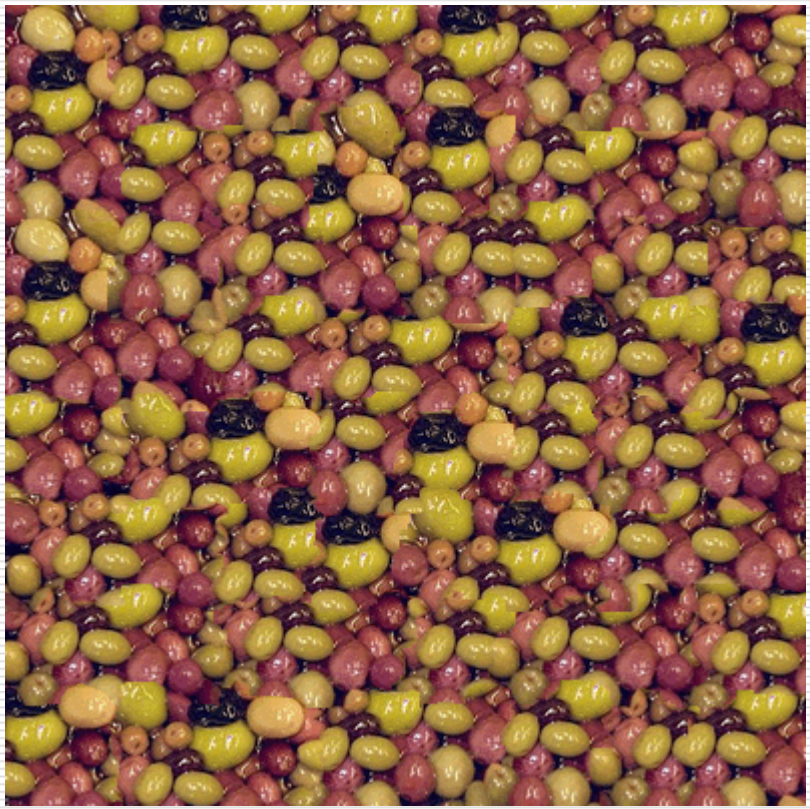
Feature Map (cont.)

- Feature map represents the structure information
- Use weighted SSD of both color and feature as the estimation function

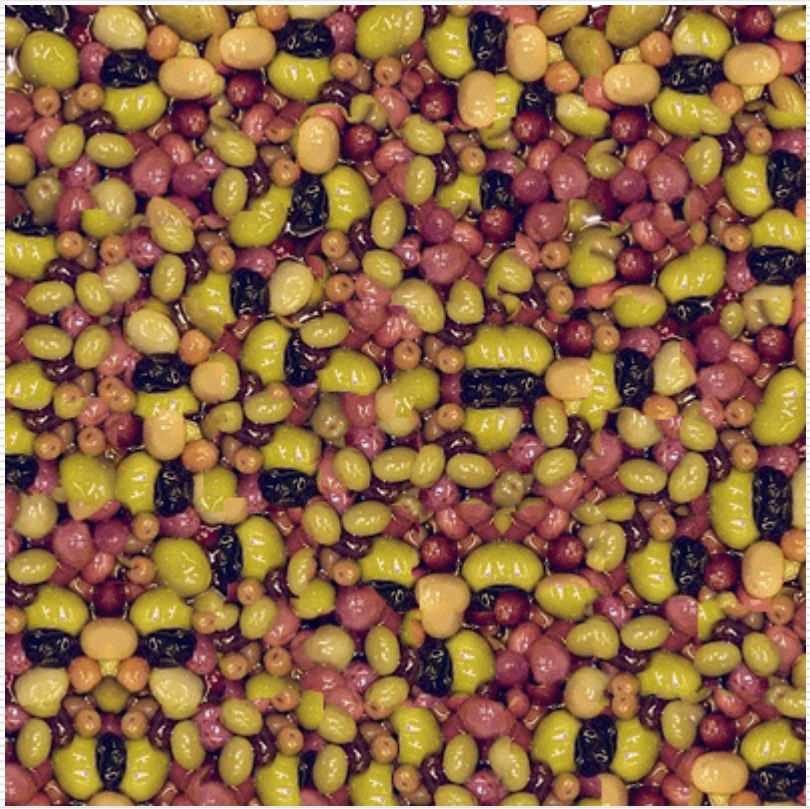


Additional Topic

- Use rotation or mirror of pattern image as candidate samples
  - Give us more choices of samples
  - Give us multiform synthesized results



No rotation or mirror



Rotation and mirror

Results

- Develop platform
  - Visual C++ (Visual Studio.Net 2003)
  - OpenCV Lib
- Go to the web pages

Conclusion

- Our algorithm combined ideas of “image quilting”, “graphcut” and “feature map”
- Our algorithm generates different images from same pattern image every time
- Our algorithm uses rotation and mirror transform to get multiform synthesized results

Conclusion (cont.)

- Our algorithm is fast
- The result is good with texture images including strong-structure images
- The result is good with some general image

Thank you!