Patch-Based Texture Synthesis in 2D and 3D

Presented by Junwen Wu

CSE291-J

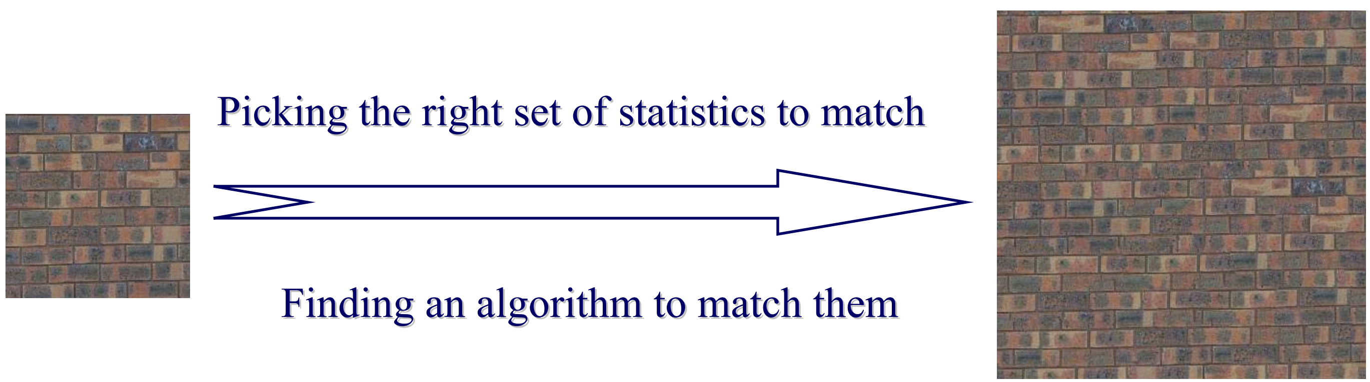
Feb. 27, 2003

Introduction to Texture Synthesis

- Problem Description

  Given: an input texture sample image I_{in}

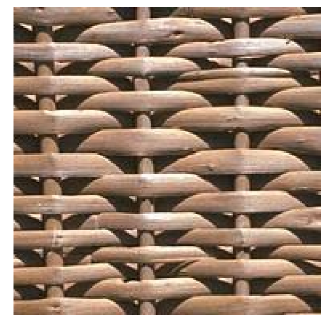
  Target: an unlimited amount of image data I_{out}, which is perceived by the human observer as the same texture



Introduction to Texture Synthesis

- Two classes of textures

Stochastic Texture *V.S.* Structured Texture



•Histogram matching

•Conditional distribution preserving under multi-scale

•Patch pasting

•Pixel-wise non-parametric sampling

•Pixel-wise non-parametric sampling

Greedy strategy, Slow

Patch-based Texture Synthesis

Patch-based Texture Synthesis in Images

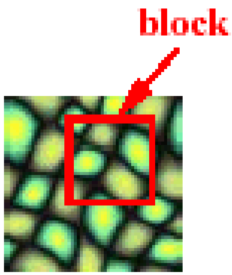
- Alexei A. Efros, William T. Freeman, *Image quilting for Texture Synthesis and Transfer*, SIGGRAPH 2001

  (Image Quilting)

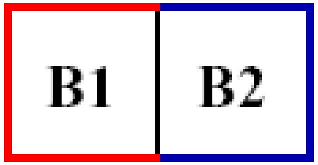
- Lin Liang, Ce Liu, Ying-qing Xu, Baining Guo and Heung-yeung Shum, *Real-time texture synthesis by patch-based sampling*, ACM Transaction on Graphics, Vol. 20, No. 3, July 2001, Pages 127-150

  (Microsoft Paper)

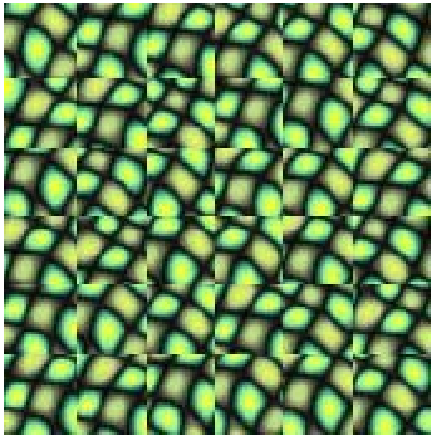
Basic Idea

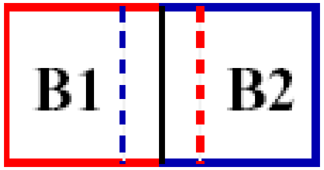


input texture

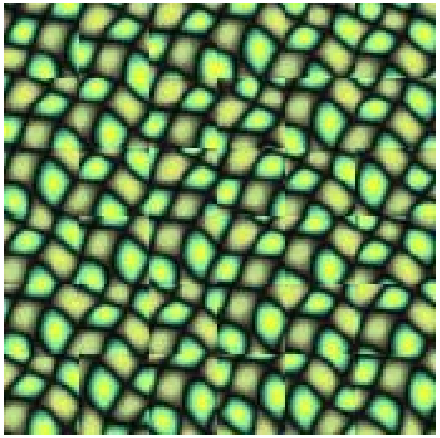


random placement of blocks



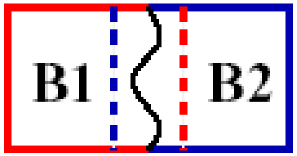


neighboring blocks constrained by overlap

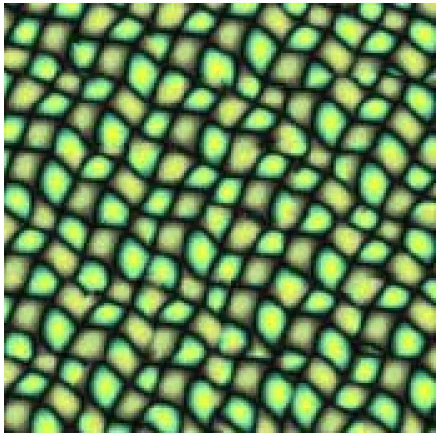


Basic Idea

Image Quilting: Minimum Error Boundary cut:



minimum error boundary cut



Microsoft Paper: Image Feathering (blending):

- Be performed along seams
- Makes the sharp changes occurring at the cut line appear more gradual

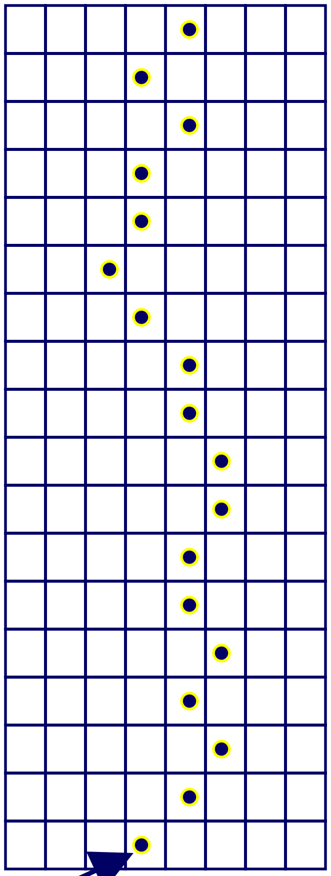
Minimum Error Boundary Cut:

E_{11}	E_{12}	E_{13}	E_{14}
E_{21}	E_{22}	E_{23}	E_{24}

$$e_{22} \quad + \quad \mathrm{Min}\{E_{11}, E_{12}, E_{13}\}$$

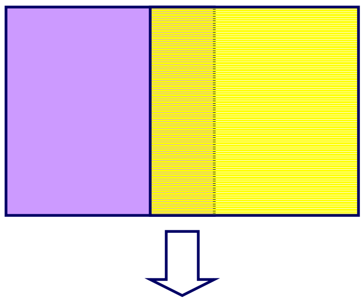
$$e_{ij} = \left(P_{in}(i,j) - P_{Match}(i,j)\right)^2$$

$$E_{ij} = \begin{cases} e_{ij}, & i = 1 \\ e_{ij} + Min\left\{E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}\right\}, & i > 1 \end{cases}$$



Minimum of last row

Image Feathering:



$$P_{out}(i,j) = \alpha_{ij} P_{in}(i,j) + (1 - \alpha_{ij}) P_{Match}(i,j)$$

- Using α_{ij} to control the blending effect
- One possible criterion to select α_{ij} is by their distance to the cutting line

How to choose the matching patch:

- Image Quilting: The most similar patch in the input image, which is defined as:

$$B_{opt} = \arg\min_{B_{in}} \{ dist(\partial B_{in}, \partial B_{out}), B_{in} \in I_{in} \},$$

∂B_{in} and ∂B_{out} are the overlapped region in the patch B_{in} and B_{out}

- Microsoft Paper: Randomly select from the set S_{Match}; In case $S_{Match} = \Phi$, select the most similar patch

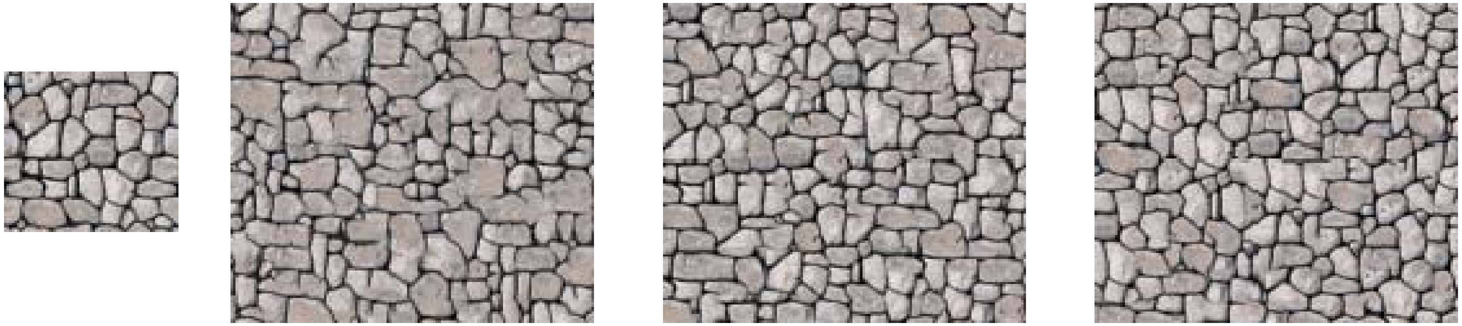
$$S_{Match} = \{ B_{in} : dist\{ \partial B_{in}, \partial B_{out} \} < d_{\max} \}$$

$$d_{\max} = \varepsilon \left(\frac{1}{A} \sum_{i,j} (P_{out}(i,j))^2 \right)^{\frac{1}{2}},$$

$P_{in/out}(i,j) \in \partial B_{in/out}$, A is the area of $\partial B_{in/out}$

Common Parameters in Both Approaches:

- Patch size: Smaller means more matching possibility, yet weaker statistical constraint

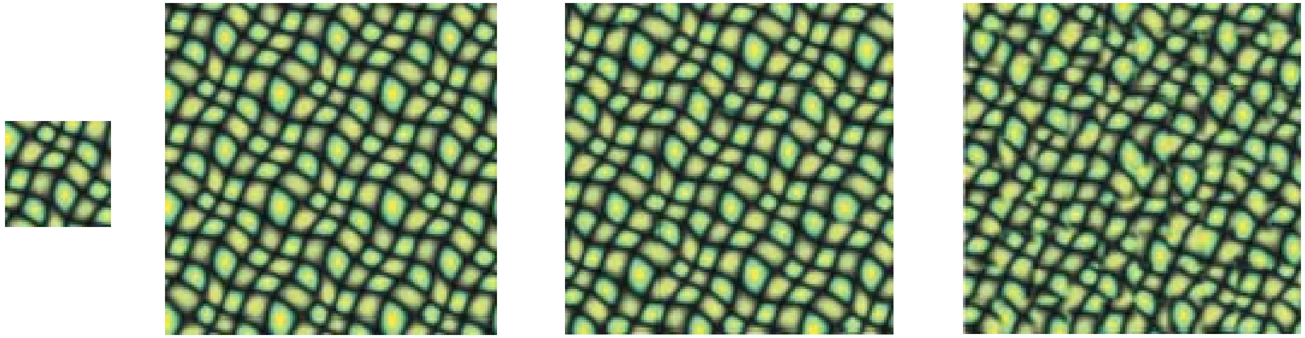


From left to right: I_{in}, I_{out} by patch size 16x16, 24x24, 32x32

- Width of Overlapped Band: Wider band implies stronger statistical constraints, yet less matching possibilities

More Parameters in Microsoft Paper:

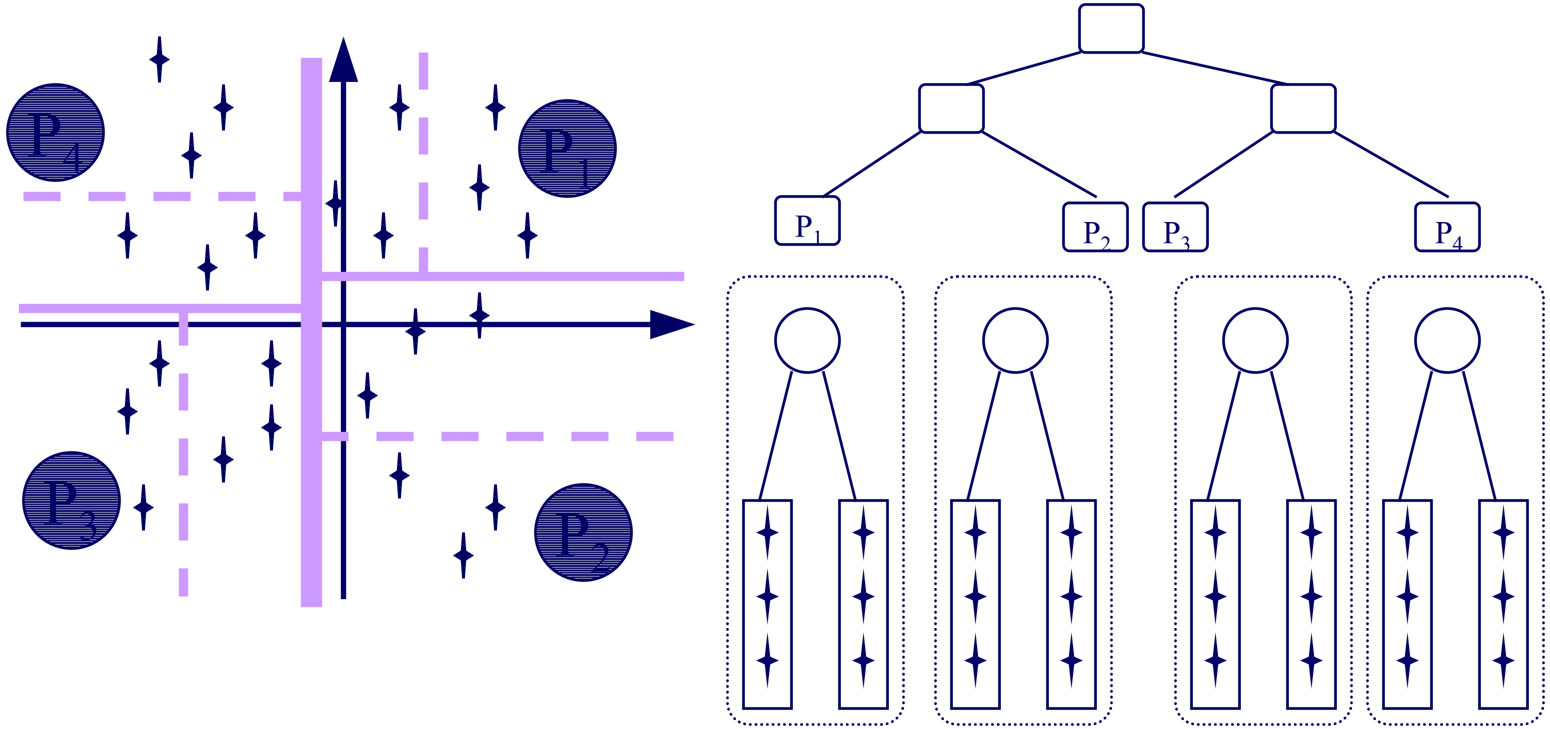
- Distance tolerance d_{max} *(Controlled by ε)*:
  Control the similarity between the synthesis texture and the input texture.
  Smaller ε: More similarity in local structures;
  Greater ε: Discontinuous transition between patches



From left to right: I_{in}, I_{out} with ε=0, 0.2, 1 respectively

Three Steps of Accelerating Technique----Step 1:

- Optimized KD-tree for the data-points



Three Steps of Accelerating Technique---- Step 1 (Contd.):

- Optimized KD-tree for the data-points

  •Partition the data-space into hypercubes by axis-orthogonal hyperplanes

  •Node: hypercubes enclosing a set of data-points

  •Construction rule: Sliding mid-point rule *V.S.* standard KD-tree splitting rule

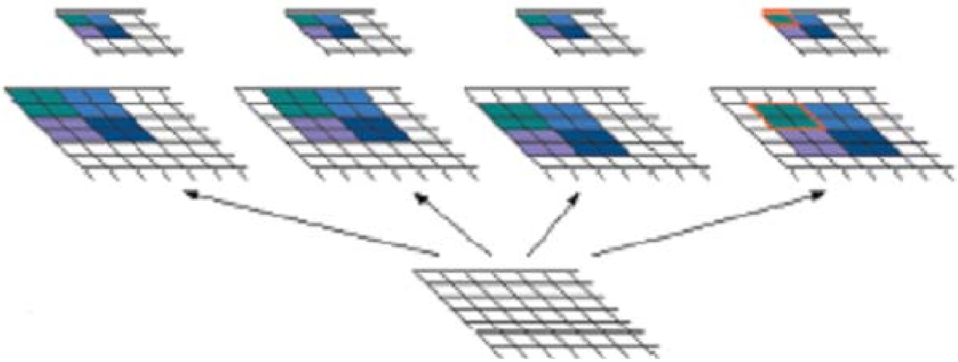
  ☆ Both will produce cubes with high-aspect ratio

  ☆ High aspect ratio will increase error

  ☆ Sliding mid-point rule can prevent it from causing problems

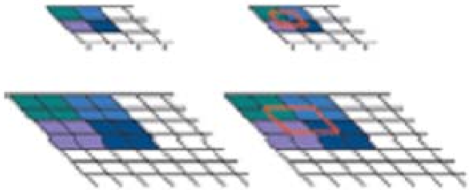
Three Steps of Accelerating Technique----Step 2:

- Quadtree Pyramid---Take advantage of image data



Gaussian Pyramid

Reduce + Expand



•Calculated over I_{in}; all other data-points can be extracted from filtered $I_{in;}$

•At every level, four children (higher level images) are computed over images with different shifting along *x*- and *y*- directions.

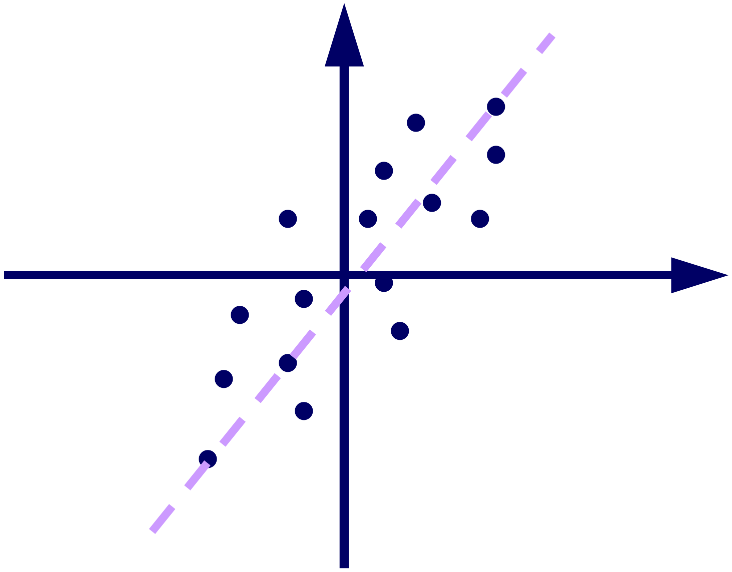
•Compare with Gaussian Pyramid: Same: reduce (Smoothing+sub-sampling) and expand (interpolating); different: Gaussian pyramid cannot always find the corresponding pixel in the higher level to the lower level patches

Three Steps of Accelerating Technique----Step 3:

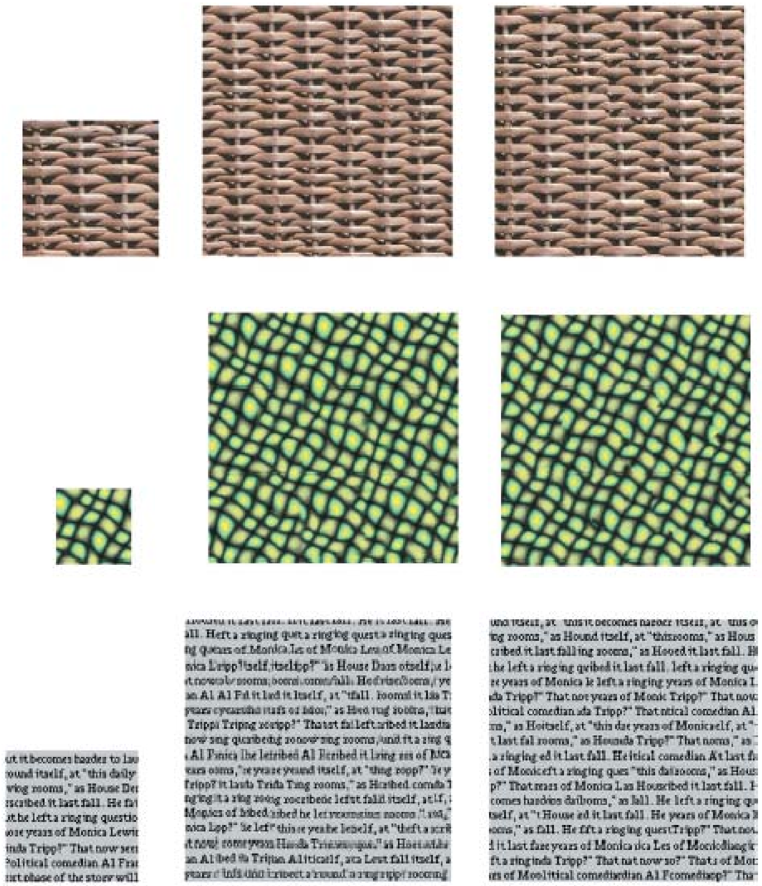
- PCA---Data dimension reduction

•A lower dimension representation

•Expanded by the first several eigenvectors of the covariance matrix



Results



Results from Liang’s paper

From left the right:

Input sample texture images;

Synthesized texture from Liang’s approach;

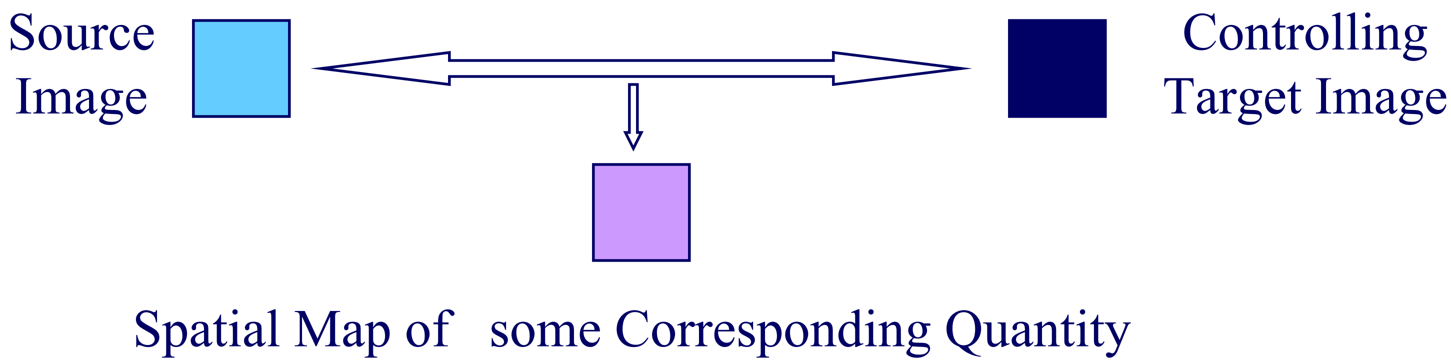
Synthesized texture from Efros and Freeman’s approach

Texture Transfer

input images　　quilting results

Texture Transfer

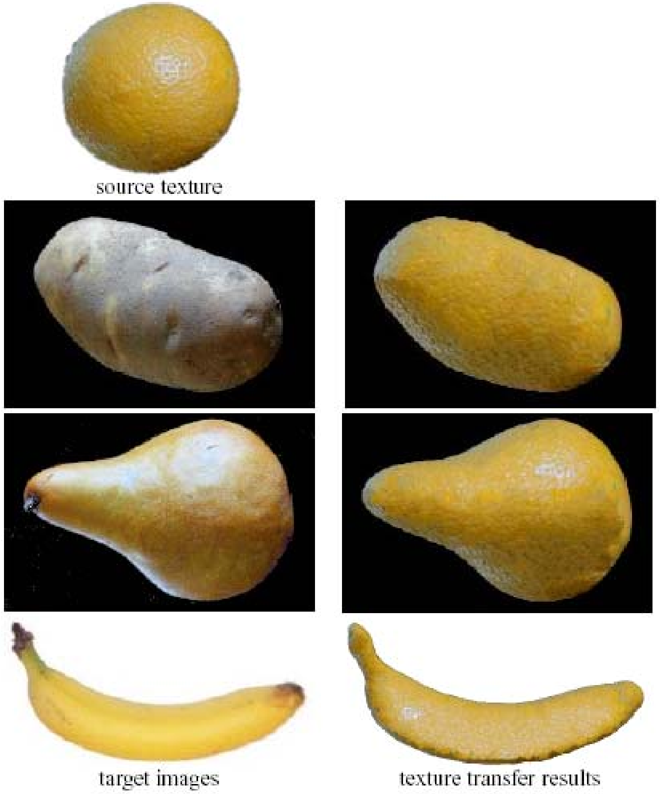
- Image quilting is suitable for it : image quilting is based on local image information
- A desired *correspondence map* should be satisfied as well as the texture synthesis requirement.
- *Correspondence map* (C(•)):

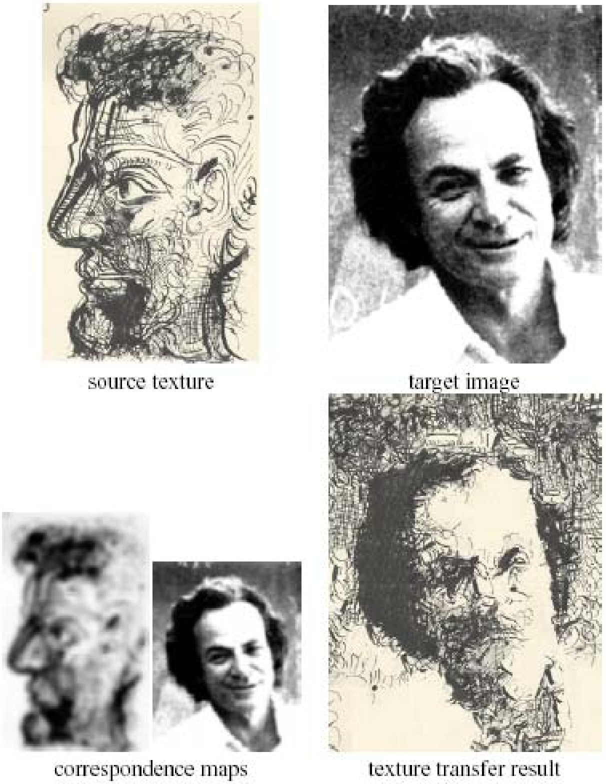


- Error term is modified to be:

$$\tilde{e}_{ij} = \alpha\left(P_{in}(i,j) - P_{out}(i,j)\right)^2 + (1-\alpha)\left(C(P_{in}(i,j)) - C(P_{out}(i,j))\right)^2$$

Results of Image Transfer:





Correspondence map:
Luminance value

Correspondence map:
Blurred Luminance value

QUESTIONS?