# Texture Synthesis and Manipulation

## Progress Report

Douglas Lanman

EN 256: Computer Vision

20 November 2006

BROWN

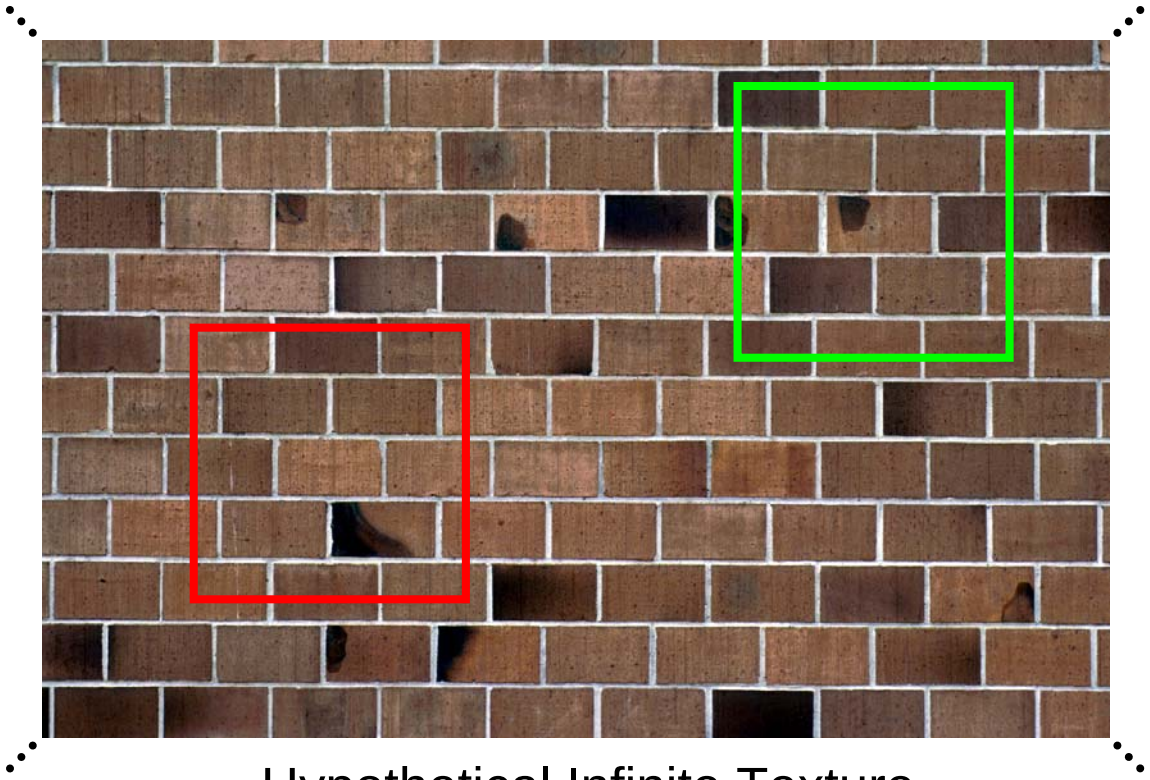
# Outline

- *Review of Texture Synthesis*
- Patch-based Synthesis using Image Quilting
- Texture Transfer and Manipulation
- Revised Project Goals and Timeline

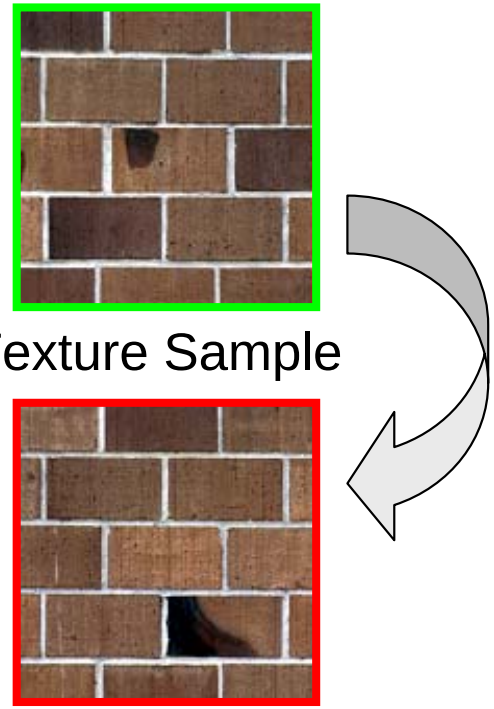
# What is Texture Synthesis?

**The Texture Synthesis Problem:**

- Given a finite texture sample, synthesize additional samples which appear (to a human observer) to be generated from the same underlying stochastic process.



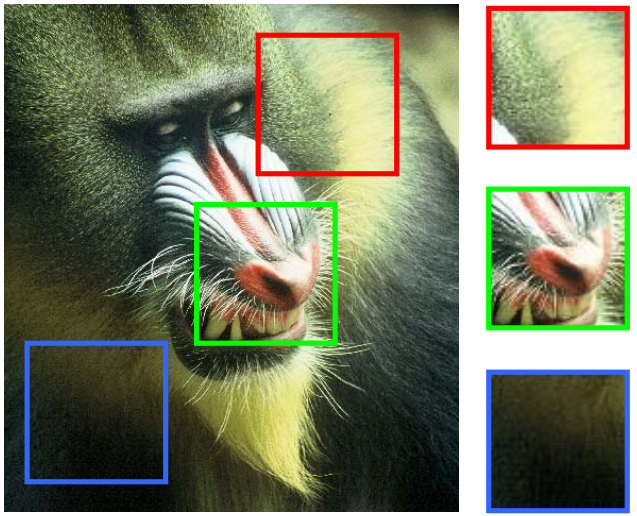
Hypothetical Infinite Texture

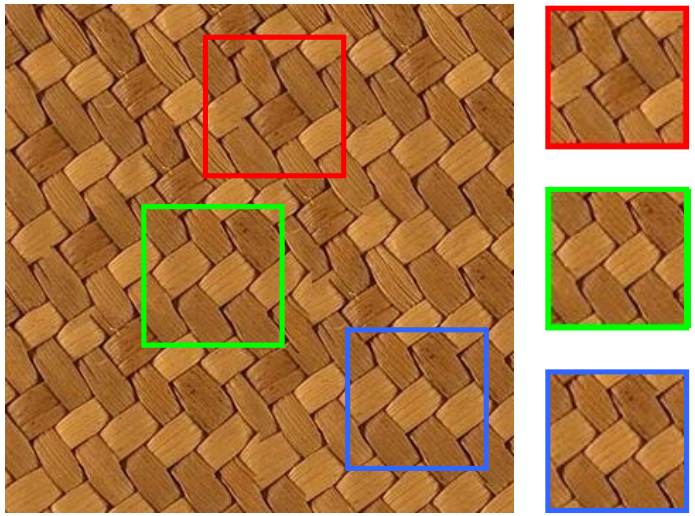



Synthesized Texture

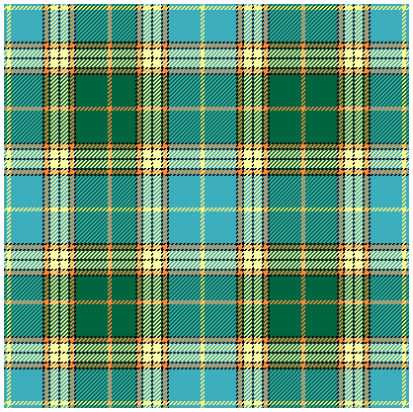
# Characteristics of Natural Textures

## Locality and Stationarity of Random Processes





## Stochastic vs. Regular Textures

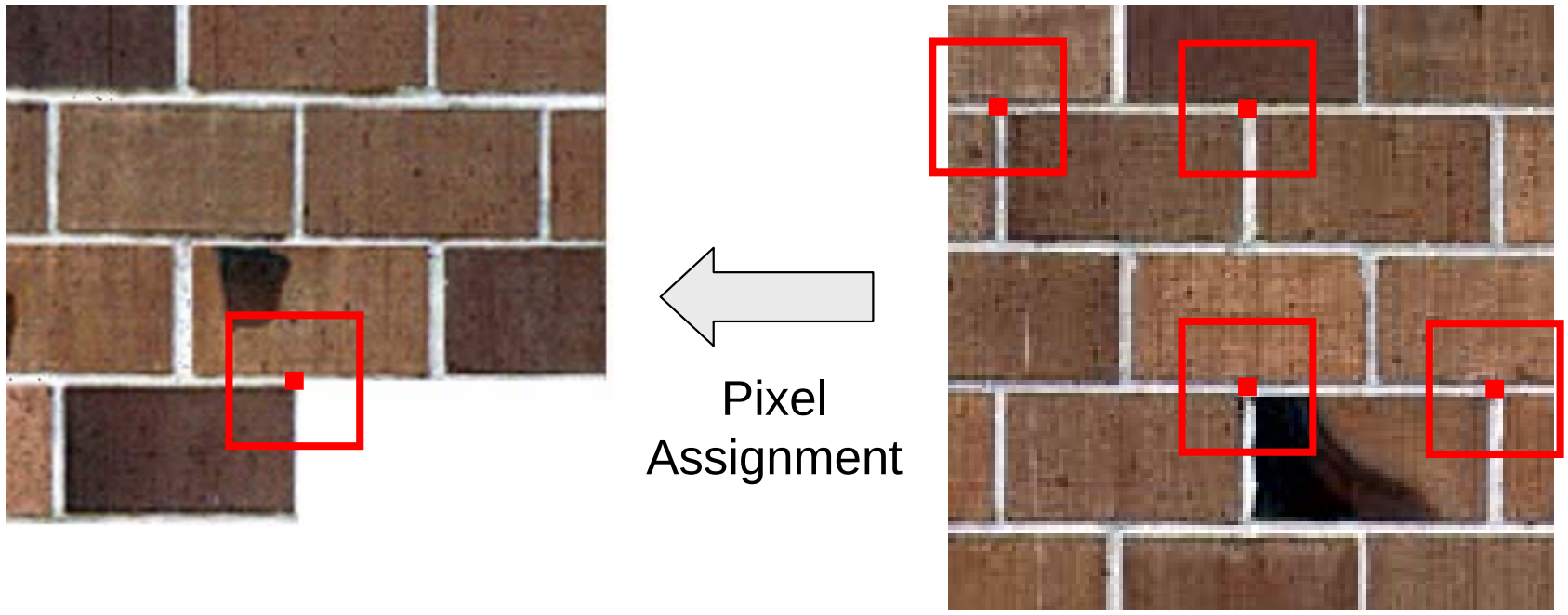


Regular Tiling

Stochastic

Weakly Homogeneous

# Pixel-based Texture Synthesis



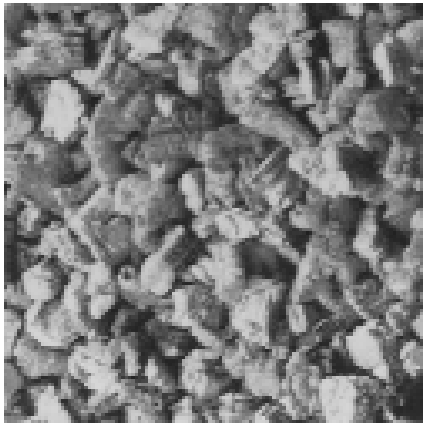
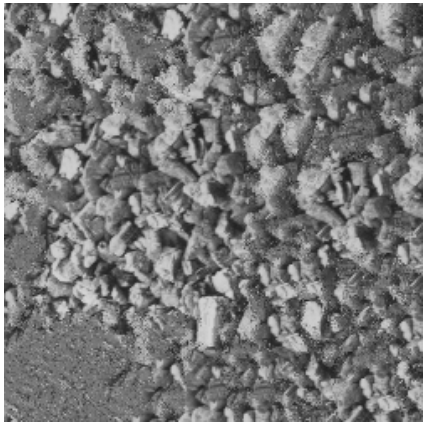

**Pixel-based Synthesis Procedure [Wei and Levoy '00]**

- Assume a Markov Random Field (i.e., local and stationary process)
- Rather than estimating conditional density, simply sample image
- Starting from a set of initial seed values, search the input texture for similar neighborhoods and assign randomly from this set

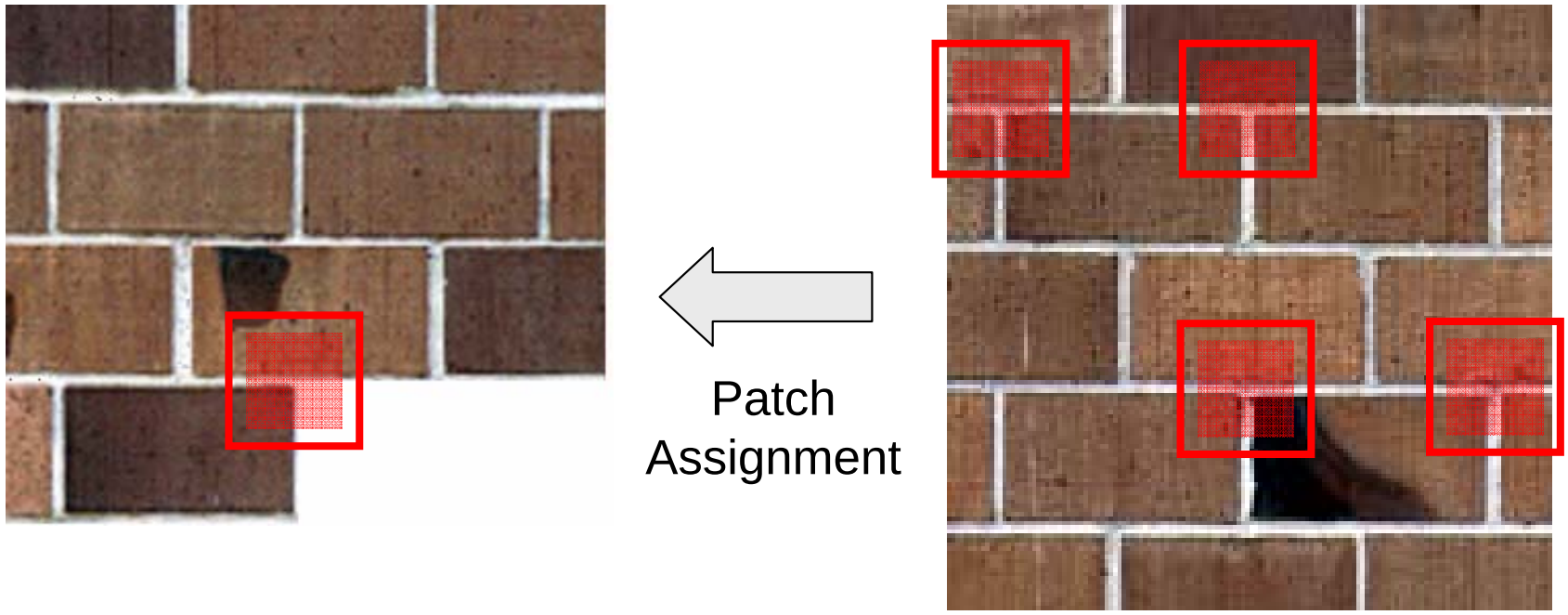
# Pixel-based Texture Synthesis Methods

| Input Sample | Wei and Levoy | Ashikhmin | Hertzmann et al. |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |

# Limitations and Failure Modes

| "Garbage" | Verbatim Copying | Blurring |
|---|---|---|
|  |  |  |
|  |  |  |

# Patch-based Texture Synthesis




**Patch-based Synthesis Procedure [Efros and Freeman '01]**

- Pixel-based methods result in correlated neighboring pixels
- To accelerate synthesis, simply assign patches rather than pixels
- Starting from an initial patch, search the input texture for similar neighborhoods and assign next patch randomly from this set

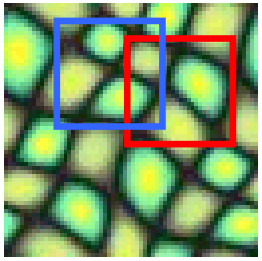
# Patch-based Texture Synthesis Results

| Input Sample | Image Quilting | Graphcut Texture |
|---|---|---|
|  |  |  |
|  |  |  |

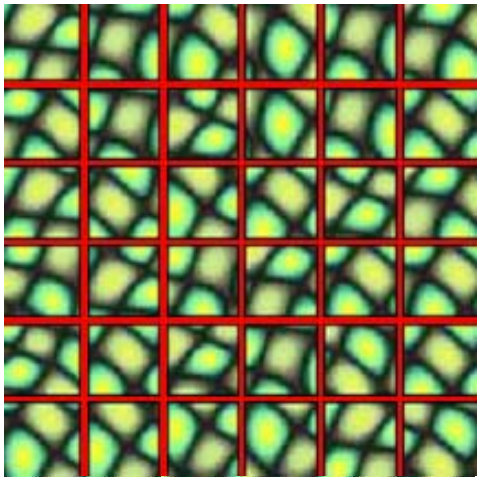
# Outline

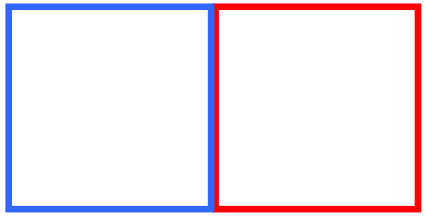
- Review of Texture Synthesis
- *Texture Synthesis using Image Quilting*
  - Implementation Details
  - MATLAB Demo
  - Performance Analysis
  - Comparison to Pixel-based Methods
- Texture Transfer and Manipulation
- Revised Project Goals and Timeline

# Overview of Image Quilting

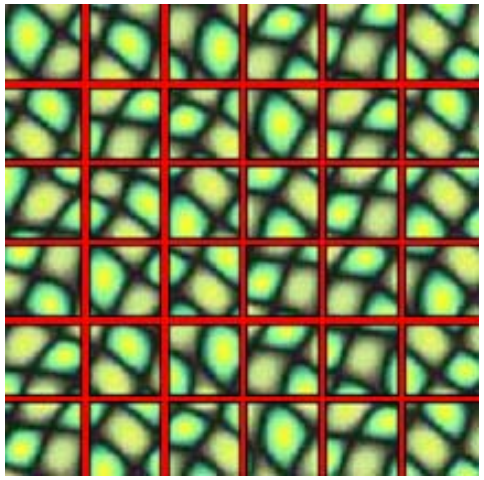


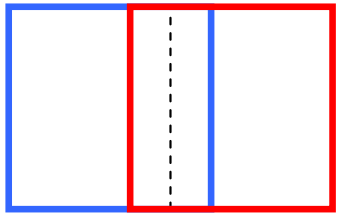
Input Sample



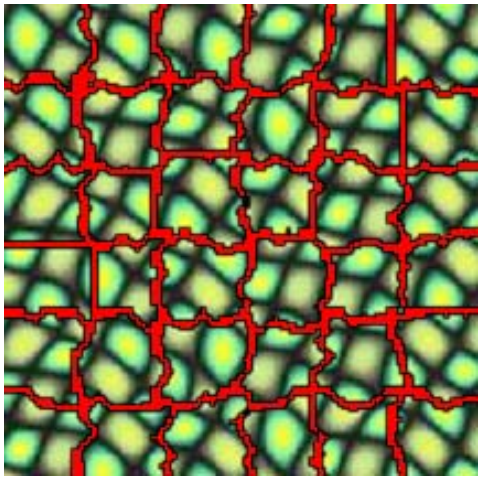


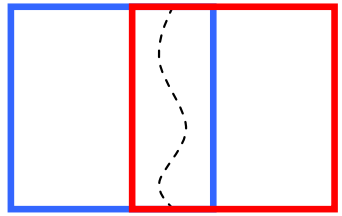
Random Placement





Constrained Overlap





Minimal Error Cut

**Image Quilting Procedure [Efros and Freeman '01]**

- Append blocks to initial seed so that region of overlap is similar
- Define boundary by minimum cost path through overlap error

# Initialization and Constrained Overlap

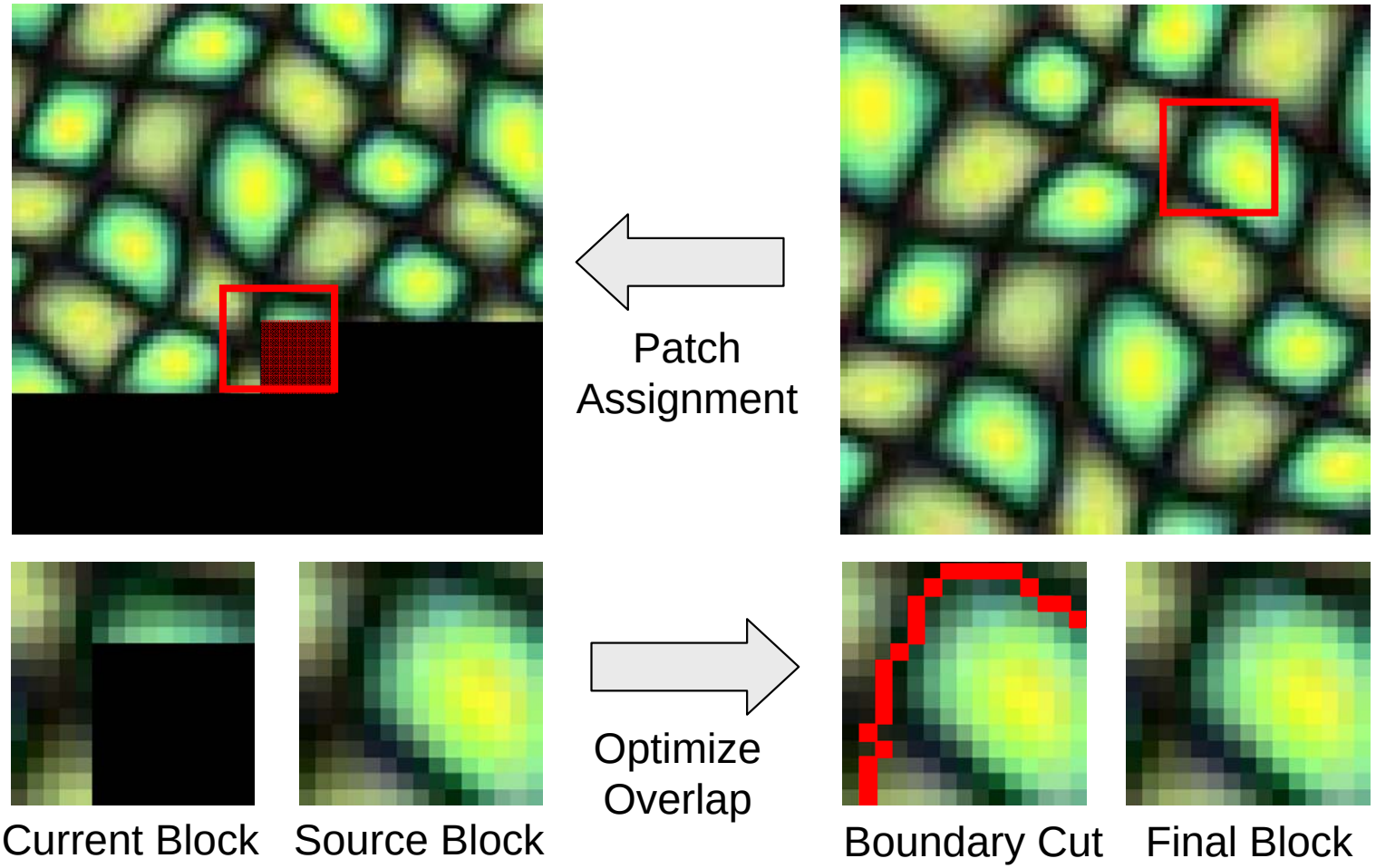
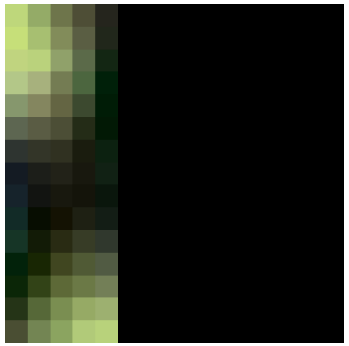


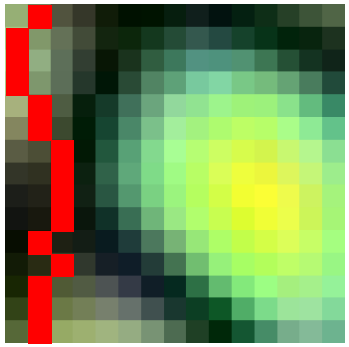

**Image Quilting Procedure**

- Define similarity using $L_2$-norm applied to every pixel/color in block
- Optimize overlap region using minimum error boundary cut

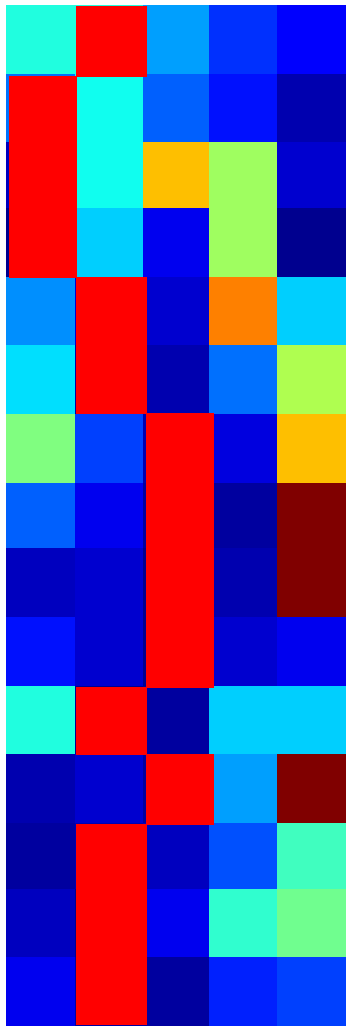
# Finding the Optimal Boundary Cut



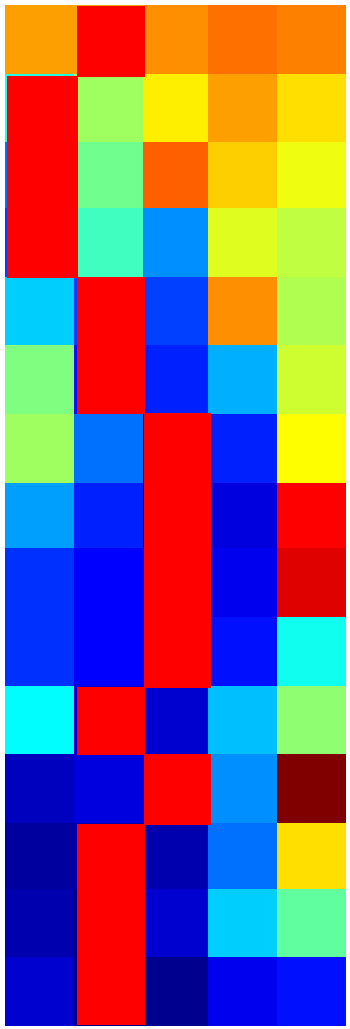
Current Block



Source Block



Overlap Error



Path Costs

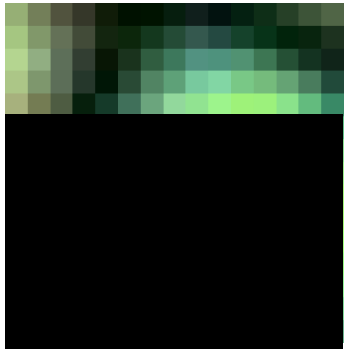
## Case I: Overlapping Columns

- Evaluate overlap error using $L_2$-norm applied to blocks
- Recursively compute path cost matrix using greedy algorithm (bottom-to-top along rows)

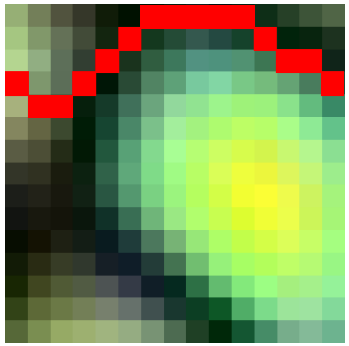
$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

- Trace cost matrix from top-to-bottom (starting at minimum-cost element in the first row)

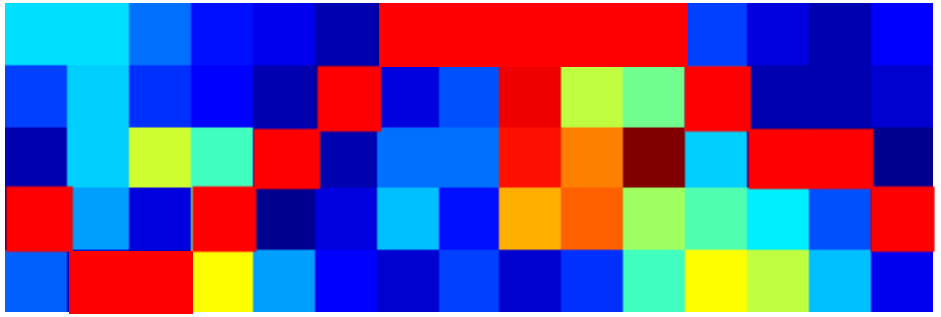
# Finding the Optimal Boundary Cut



Current Block



Source Block



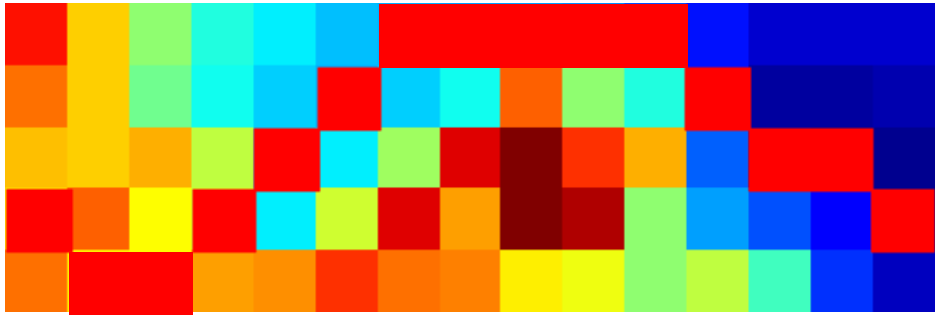
Overlap Error

## Case II: Overlapping Rows

- Evaluate overlap error using $L_2$-norm applied to blocks
- Recursively compute path cost matrix using greedy algorithm (right-to-left along rows)

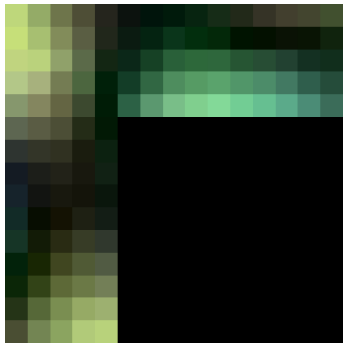
$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$$

- Trace cost matrix from left-to-right (starting at minimum-cost element in the first column)

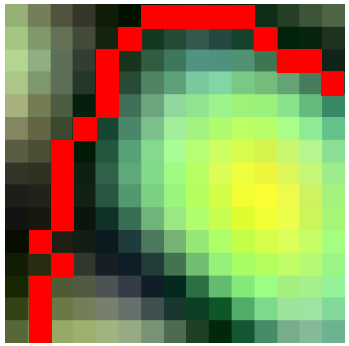


Path Costs

# Handling "L-shaped" Overlap Regions



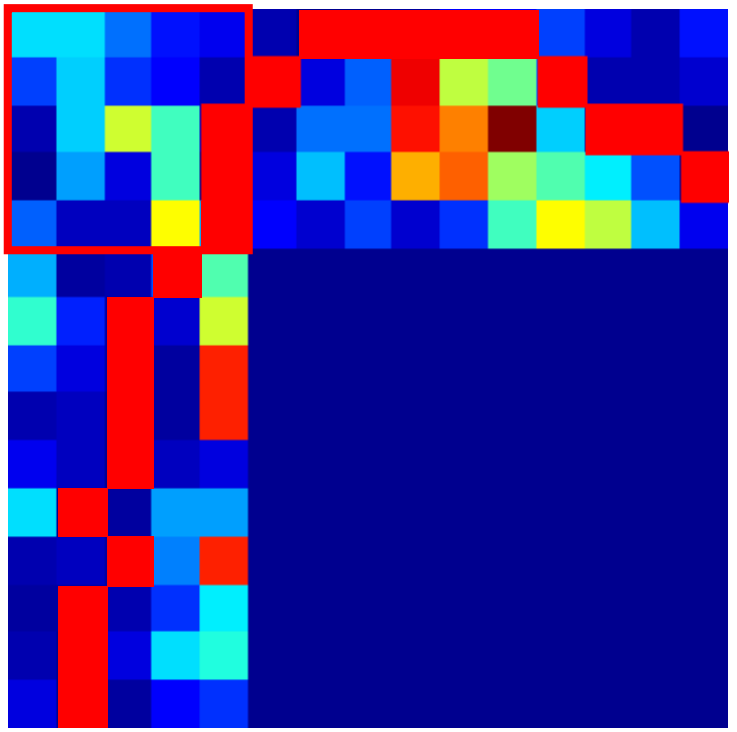
Current Block



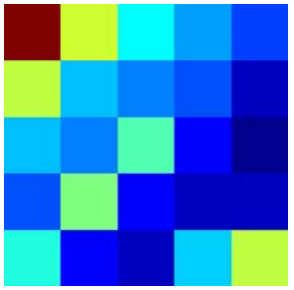
Source Block

**Case III: "L-shaped" Regions**

- Evaluate overlap error using $L_2$-norm applied to blocks
- Recursively compute path cost matrices using greedy method
- Determine starting pixel using overlapping costs in "corner"
- Trace cost matrix from left-to-right and top-to-bottom (from minimum-cost element)



Overlap Error



"Corner" Cost Matrix

# MATLAB Demo

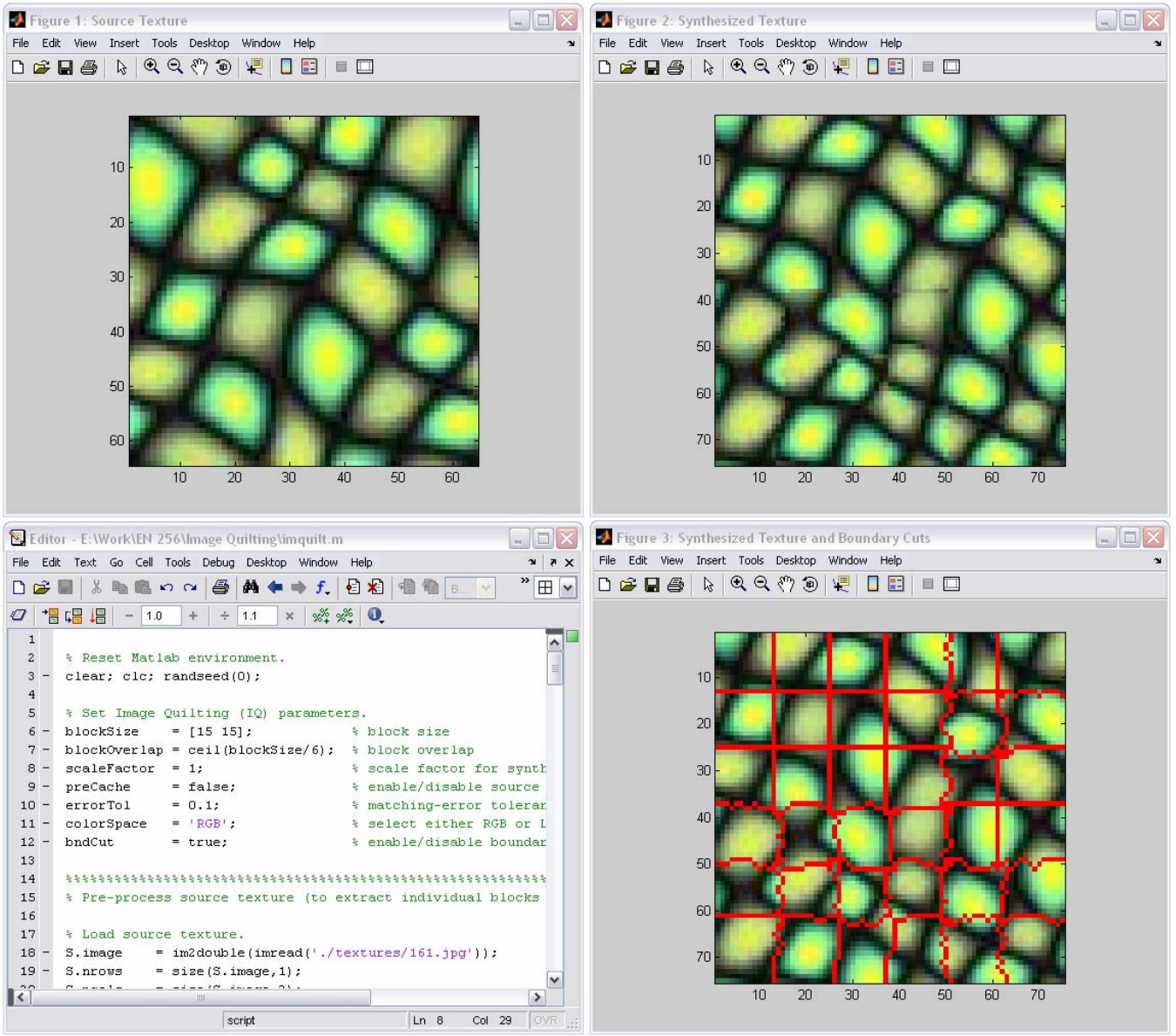


Figure 1: Source Texture
File Edit View Insert Tools Desktop Window Help
Figure 2: Synthesized Texture
File Edit View Insert Tools Desktop Window Help
Editor - E:\Work\EN 256\Image Quilting\imquilt.m
File Edit Text Go Cell Tools Debug Desktop Window Help
% Reset Matlab environment.
clear; clc; randseed(0);
% Set Image Quilting (IQ) parameters.
blockSize = [15 15]; % block size
blockOverlap = ceil(blockSize/6); % block overlap
scaleFactor = 1; % scale factor for synth
preCache = false; % enable/disable source
errorTol = 0.1; % matching-error toleran
colorSpace = 'RGB'; % select either RGB or L
bndCut = true; % enable/disable boundar
% Pre-process source texture (to extract individual blocks
% Load source texture.
S.image = im2double(imread('./textures/161.jpg'));
S.nrows = size(S.image,1);
script
Ln 8 Col 29
Figure 3: Synthesized Texture and Boundary Cuts
File Edit View Insert Tools Desktop Window Help

# Performance Analysis

Input Image

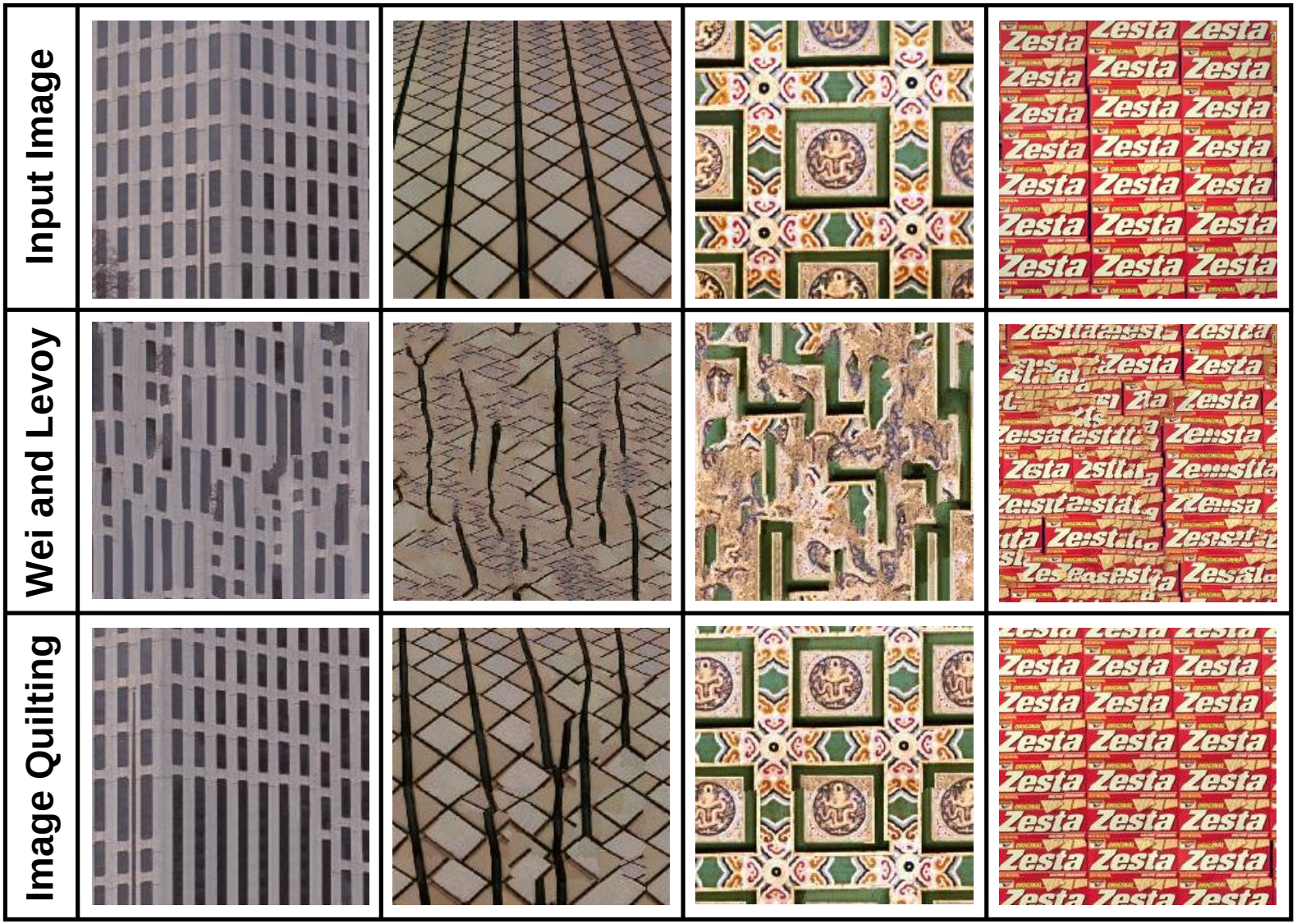



Synthesized Textures

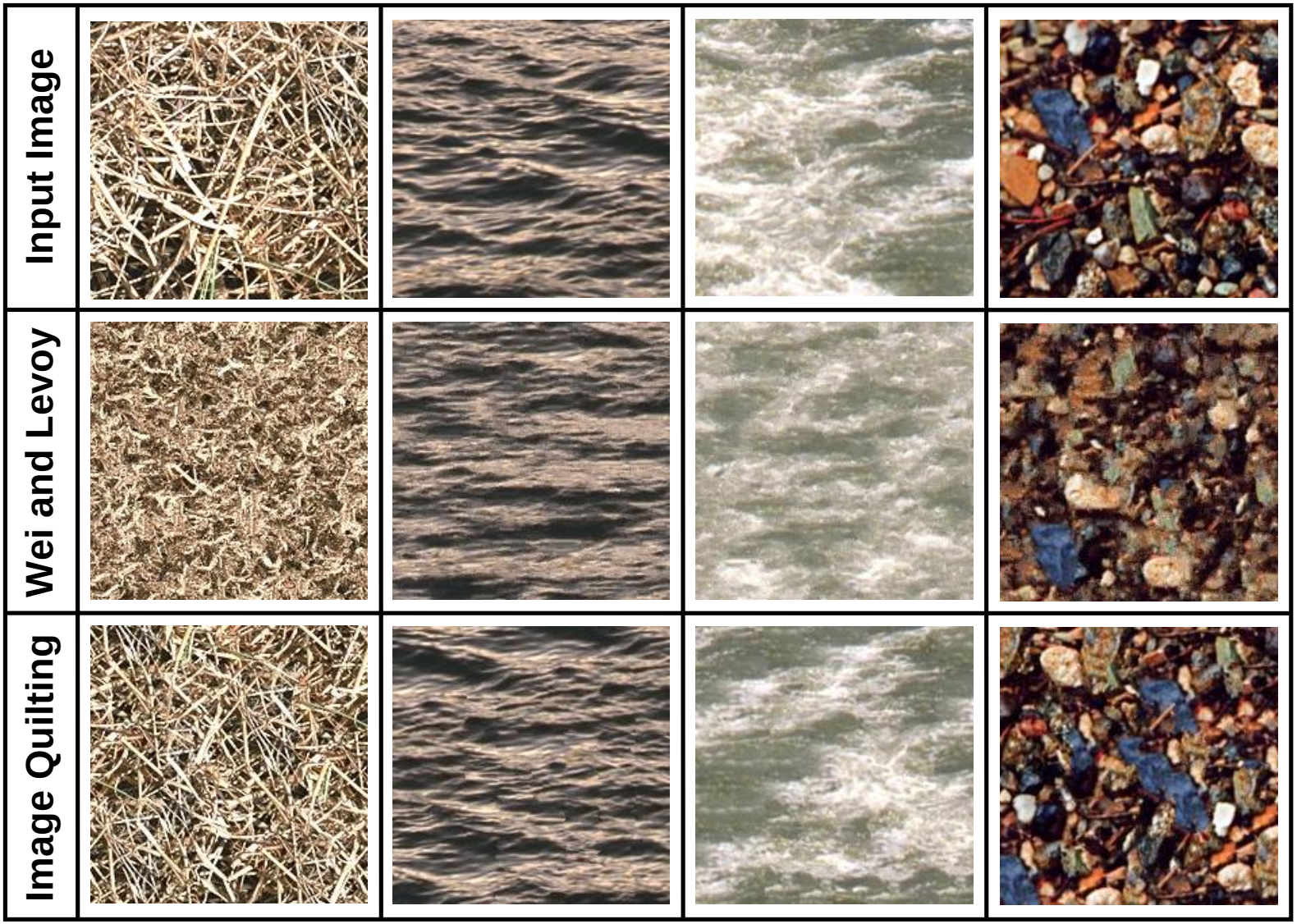
## Image Quilting Parameter Tuning

- Only two parameters: (1) block size and (2) width of overlap edge
- Block size must be large enough to capture relevant structures, yet small enough to allow sufficient samples within the source texture
- [Efros and Freeman '01] use overlaps equal to 1/6 of the block size

# Comparison for Near-Regular Textures



# Comparison for Stochastic Textures



# Comparison for Weakly Homogeneous Textures

# Outline

- Review of Texture Synthesis
- Texture Synthesis using Image Quilting
- *Texture Transfer and Manipulation*
  - Implementation Details
  - Performance Analysis
- Revised Project Goals and Timeline

# Overview of Texture Transfer

**Texture Transfer [Efros and Freeman '01]**

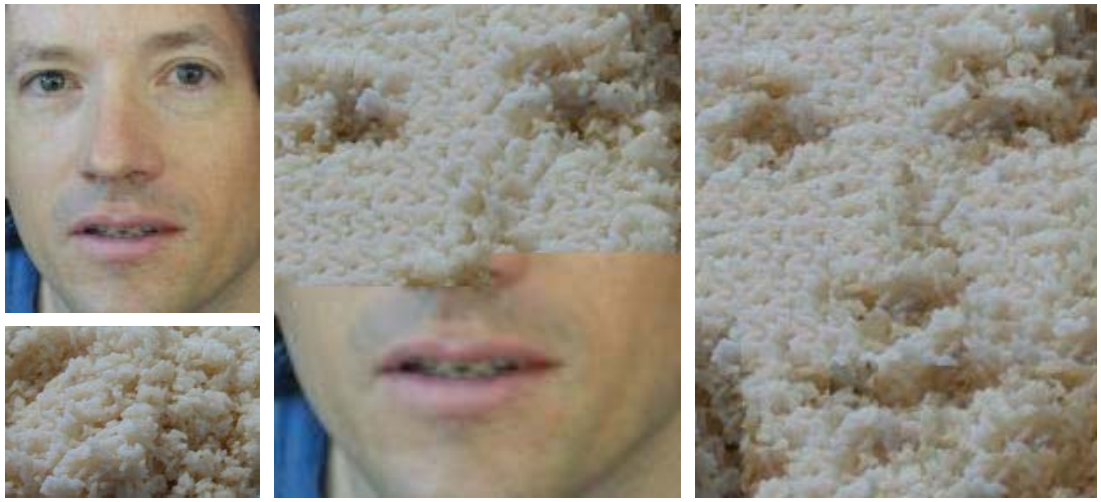
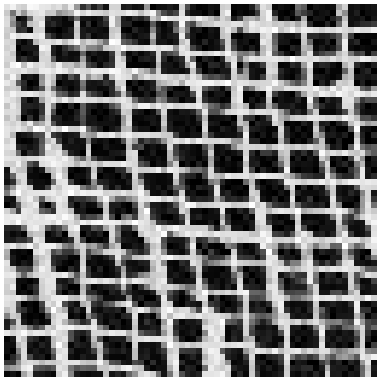


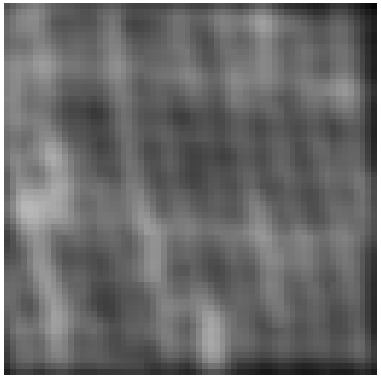
**Image Analogies [Hertzmann et al. '01]**


Source (A) Filtered Source (A′) Target (B) Filtered Target (B′)

# Texture Transfer using Image Quilting

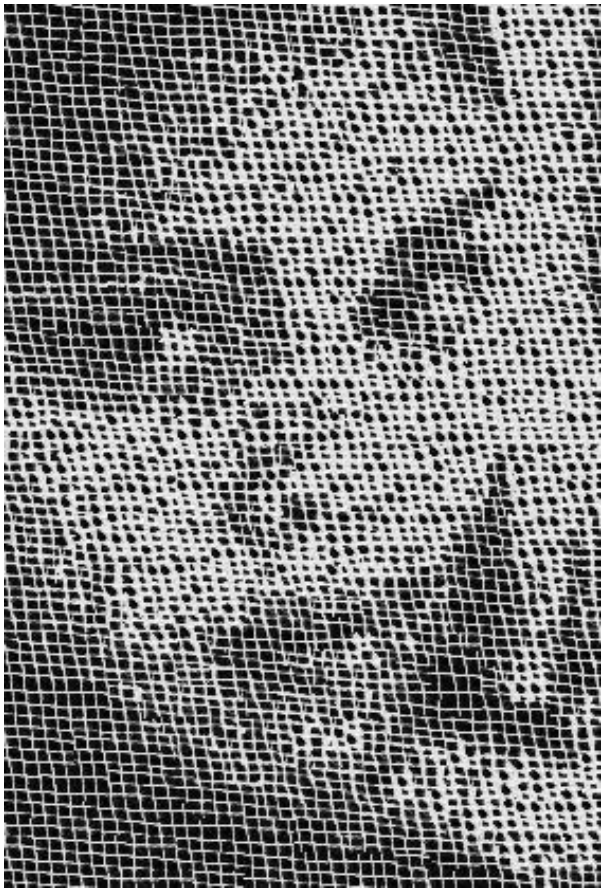


Source Image



Blurred Luminance

Target Image

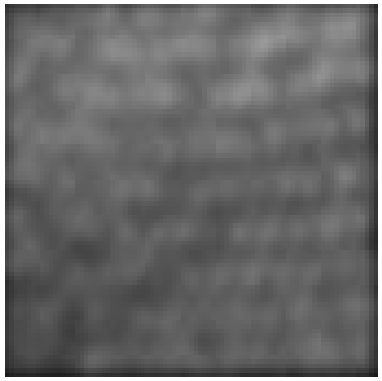


Texture Transfer Result

- From an initial patch, search source texture and luminance map for similar neighborhoods and assign next patch randomly from this set

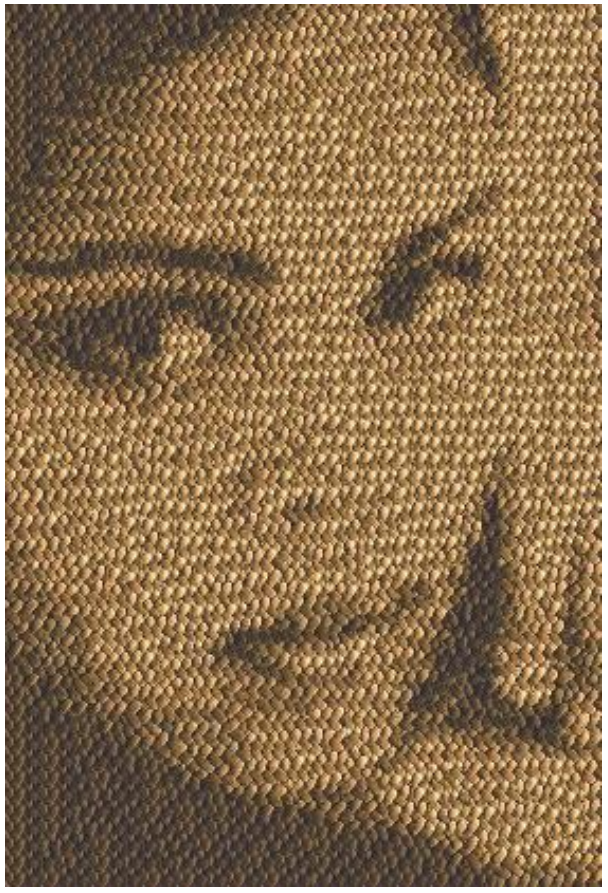
# Additional Texture Transfer Example

Source Image



Blurred Luminance

Target Image



Texture Transfer Result

- From an initial patch, search source texture and luminance map for similar neighborhoods and assign next patch randomly from this set

# Outline

- Review of Texture Synthesis
- Patch-based Synthesis using Image Quilting
- Texture Transfer and Manipulation
- *Revised Project Goals and Timeline*

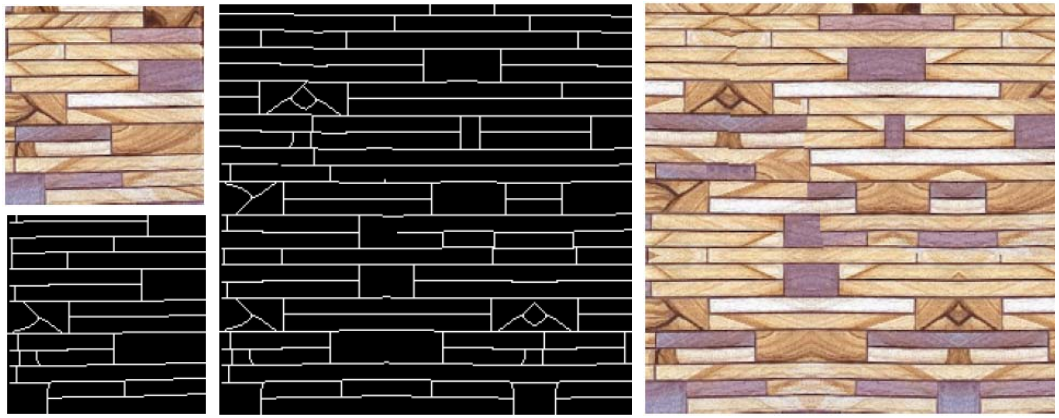
# Revised Project Goals and Timeline

**Proposed and Completed Primary Goals (for Progress Report)**

- ✓ *Implement patch-based texture synthesis using Image Quilting*
- ✓ Use texture transfer to allow user-controlled synthesis
- ✓ Evaluate regular, stochastic, and weakly-homogeneous samples
- ✓ Compare results to existing methods using available implementations

**Secondary Goals (for Final Report)**

- Implement graphcut-based texture synthesis
- Extend texture transfer to allow patch-based image analogies
- *Evaluate feature matching and texture deformation*
- *Examine extensions for inpainting and retouching*
- Apply texture synthesis to surface completion problem



Feature Matching and Image Deformation

# References

## "Early" Approaches: Texture Analysis and Psychophysics


1. D.J. Heeger and J.R. Bergen, "Pyramid-Based Texture Analysis/Synthesis", SIGGRAPH '95.
2. J.S. De Bonet, "Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images", SIGGRAPH '97.


## Pixel-based Texture Synthesis


3. A.A. Efros and T.K. Leung, "Texture Synthesis by Non-parametric Sampling", ICCV, 1998.
4. L. Wei and M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", SIGGRAPH '00.
5. M. Ashikhmin, "Synthesizing Natural Textures", Interactive 3D Graphics (I3D), 2001.
6. A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, D.H. Salesin, "Image Analogies", SIGGRAPH '01.


## Patched-based Texture Synthesis


7. Y. Xu, B. Guo, and H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis", Microsoft Research Technical Report, MSR-TR-2000-32, 2000.
8. A.A. Efros and W. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH '01.
9. V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts", SIGGRAPH '03.
10. Q. Wu and Y. Yu, "Feature Matching and Deformation for Texture Synthesis", SIGGRAPH '04.