CPSC 425: Computer Vision

Lecture 10: Texture

(unless otherwise stated slides are taken or adopted from Bob Woodham, Jim Little and Fred Tung)
Texture

What is **texture**?

Texture is widespread, easy to recognize, but hard to define

Views of large numbers of small objects are often considered textures
  — e.g. grass, foliage, pebbles, hair

Patterned surface markings are considered textures
  — e.g. patterns on wood

*Figure Credit:* Alexei Efros and Thomas Leung
Definition of **Texture**

*(Functional) Definition:*

**Texture** is detail in an image that is at a scale too small to be resolved into its constituent elements and at a scale large enough to be apparent in the spatial distribution of image measurements.
(Functional) **Definition:**

*Texture* is detail in an image that is at a scale too small to be resolved into its constituent elements and at a scale large enough to be apparent in the spatial distribution of image measurements.

Sometimes, textures are thought of as patterns composed of repeated instances of one (or more) identifiable elements, called **textons**.
— e.g. bricks in a wall, spots on a cheetah
Uses of **Texture**

Texture can be a strong cue to **object identity** if the object has distinctive material properties.

Texture can be a strong cue to an **object’s shape** based on the deformation of the texture from point to point.

— Estimating surface orientation or shape from texture is known as “**shape from texture**”
Texture

We will look at two main questions:

1. How do we represent texture?
   → Texture analysis

2. How do we generate new examples of a texture?
   → Texture synthesis

We begin with texture synthesis to set up Assignment 3
Texture Synthesis

Why might we want to synthesize texture?

1. To fill holes in images (inpainting)
   — Art directors might want to remove telephone wires. Restorers might want to remove scratches or marks.
   — We need to find something to put in place of the pixels that were removed
   — We synthesize regions of texture that fit in and look convincing
Why might we want to synthesize texture?

1. To fill holes in images (inpainting)
   - Art directors might want to remove telephone wires. Restorers might want to remove scratches or marks.
   - We need to find something to put in place of the pixels that were removed
   - We synthesize regions of texture that fit in and look convincing

2. To produce large quantities of texture for computer graphics
   - Good textures make object models look more realistic
Texture Synthesis

Szeliski, Fig. 10.49
Texture Synthesis

Bush campaign digitally altered TV ad

President Bush’s campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.

Original photograph

Photo Credit: Associated Press
Texture Synthesis

Cover of “The Economist,” June 19, 2010

Photo Credit (right): Reuters/Larry Downing
Assignment 3 Preview: Texture Synthesis

**Task:** Make donkey vanish
**Assignment 3** Preview: Texture Synthesis

**Task:** Make donkey vanish

**Method:** Fill-in regions using texture from the white box
Assignment 3 Preview: Texture Synthesis

Task: Make donkey vanish

Method: Fill-in regions using texture from the white box
Objective: Generate new examples of a texture. We take a “data-driven" approach.

Idea: Use an image of the texture as the source of a probability model.
- Draw samples directly from the actual texture.
- Can account for more types of structure.
- Very simple to implement.
- Success depends on choosing a correct “distance”.
Texture Synthesis by Non-parametric Sampling

Alexei Efros and Thomas Leung
UC Berkeley

Slide Credit: http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.ppt
white bread  

brick wall  

Efros and Leung
Like *Copying*, But not Just Repetition
— What is **conditional** probability distribution of $p$, given the neighbourhood window?
— What is **conditional** probability distribution of \( p \), given the neighbourhood window?

— Directly search the input image for all such neighbourhoods to produce a **histogram** for \( p \)
What is conditional probability distribution of $p$, given the neighbourhood window?

Directly search the input image for all such neighbourhoods to produce a histogram for $p$.

To synthesize $p$, pick one match at random.
Efros and Leung: Synthesizing One Pixel

Infinite sample image

— Since the sample image is finite, an exact neighbourhood match might not be present
— Since the sample image is finite, an exact neighbourhood match might not be present

— Find the **best match** using SSD error, weighted by Gaussian to emphasize local structure, and take all samples within some distance from that match
For multiple pixels, "grow" the texture in layers
— In the case of hole-filling, start from the edges of the hole

For an interactive demo, see
(written by Julieta Martinez, a previous CPSC 425 TA)
Randomness Parameter

Slide Credit: http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.ppt
Texturing a Sphere

Sample image

2D

3D

Slide Credit: http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.ppt
Efros and Leung: More Synthesis Results

Forsyth & Ponce (2nd ed.) Figure 6.12
“Big Data” Meets Inpainting

“Big Data” enables surprisingly simple non-parametric, matching-based techniques to solve complex problems in computer graphics and vision.

Suppose instead of a single image, you had a massive database of a million images. What could you do?
“Big Data” Meets Inpainting

Original Image

Input

Figure Credit: Hays and Efros 2007
“Big Data” Meets Inpainting

Figure Credit: Hays and Efros 2007
Effectiveness of “Big Data”
Effectiveness of “Big Data”

10 nearest neighbors from a collection of 20,000 images

Figure Credit: Hays and Efros 2007
Effectiveness of “Big Data”

10 nearest neighbors from a collection of 2 million images

Figure Credit: Hays and Efros 2007
“Big Data” Meets Inpainting

Figure Credit: Hays and Efros 2007
Algorithm sketch (Hays and Efros 2007):

1. Create a short list of a few hundred “best matching" images based on global image statistics

2. Find patches in the short list that match the context surrounding the image region we want to fill

3. Blend the match into the original image

Purely data-driven, requires no manual labeling of images
“Big Data” Meets Inpainting

Figure Credit: Hays and Efros 2007
“Big Data” Meets Inpainting

Figure Credit: Hays and Efros 2007
“Big Data” Meets Inpainting

Figure Credit: Hays and Efros 2007