CPSC 425: Computer Vision

Lecture 10: Texture

( unless otherwise stated slides are taken or adopted from Bob Woodham, Jim Little and Fred Tung )
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
(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**
Question: Is texture a property of a point or a property of a region?
Texture **Segmentation**

**Question:** Is texture a property of a point or a property of a region?

**Answer:** We need a region to have a texture.
Texture Segmentation

**Question**: Is texture a property of a point or a property of a region?

**Answer**: We need a region to have a texture.

There is a “chicken–and–egg” problem. Texture segmentation can be done by detecting boundaries between regions of the same (or similar) texture. Texture boundaries can be detected using standard edge detection techniques applied to the texture measures determined at each point.
Features:

— Raw Intensity
— Orientation Energy
— Brightness Gradient
— Color Gradient
— Texture gradient
Texture Segmentation

**Question:** Is texture a property of a point or a property of a region?

**Answer:** We need a region to have a texture.

There is a “chicken–and–egg” problem. Texture segmentation can be done by detecting boundaries between regions of the same (or similar) texture. Texture boundaries can be detected using standard edge detection techniques applied to the texture measures determined at each point.

We compromise! Typically one uses a local window to estimate texture properties and assigns those texture properties as point properties of the window’s center row and column.
Texture Representation

**Question:** How many degrees of freedom are there to texture?
Question: How many degrees of freedom are there to texture?

(Mathematical) Answer: Infinitely many

(Perceptual Psychology) Answer: There are perceptual constraints. But, there is no clear notion of a “texture channel” like, for example, there is for an RGB colour channel
Texture Representation

**Observation:** Textures are made up of generic sub-elements, repeated over a region with similar statistical properties.

**Idea:** Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region.
Texture Representation

**Observation**: Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

**Idea**: Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

**Question**: What filters should we use?

**Answer**: Human vision suggests spots and oriented edge filters at a variety of different orientations and scales
Texture **Representation**

**Observation:** Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

**Idea:** Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

**Question:** What filters should we use?

**Answer:** Human vision suggests spots and oriented edge filters at a variety of different orientations and scales

**Question:** How do we “summarize”?

**Answer:** Compute the mean or maximum of each filter response over the region — Other statistics can also be useful
Texture Representation

Figure Credit: Leung and Malik, 2001
Texture Representation

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th>Win. #1</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide Credit: Trevor Darrell
Texture Representation

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th></th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide Credit: Trevor Darrell
Texture **Representation**

![Graph showing texture representation with dimensions and data points](image)

<table>
<thead>
<tr>
<th>Window</th>
<th>(\text{mean } d/dx) value</th>
<th>(\text{mean } d/dy) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

*statistics to summarize patterns in small windows*

**Slide Credit:** Trevor Darrell
Texture Representation

Chi-square

0.1

0.8
Spots and Bars (Fine Scale)

Forsyth & Ponce (1st ed.) Figures 9.3–9.4
Spots and Bars (Coarse Scale)

Forsyth & Ponce (1st ed.) Figures 9.3 and 9.5
Comparison of Results

Forsyth & Ponce (1st ed.) Figures 9.4–9.5
A Short **Exercise**: Match the texture to the response

Filters

1

2

3

Mean abs responses

A

B

C

*Slide Credit: James Hays*
A Short Exercise: Match the texture to the response

Filters

Mean abs responses

Slide Credit: James Hays
Spots and Bars (Fine Scale)

Forsyth & Ponce (1st ed.) Figures 9.3–9.4
Spots and Bars (Coarse Scale)

Forsyth & Ponce (1st ed.) Figures 9.3 and 9.5
What happens to the details?
— They get smoothed out as we move to higher levels

What is preserved at the higher levels?
— Mostly large uniform regions in the original image

How would you reconstruct the original image from the image at the upper level?
— That’s not possible

Forsyth & Ponce (2nd ed.) Figure 4.17
Laplacian Pyramid

Building a **Laplacian** pyramid:
— Create a Gaussian pyramid
— Take the difference between one Gaussian pyramid level and the next (before subsampling)

**Properties**
— Also known as the difference-of-Gaussian (DOG) function, a close approximation to the Laplacian
— It is a band pass filter – each level represents a different band of spatial frequencies
Laplacian Pyramid

At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Why Laplacian Pyramid?

unit - Gaussian ≈ Laplacian

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Laplacian Pyramid

At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Can we reconstruct the original image using the pyramid?
— Yes we can!
At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Can we reconstruct the original image using the pyramid?
— Yes we can!

What do we need to store to be able to reconstruct the original image?
Let’s start by just looking at one level!

$$\text{level 0} = \text{level 1 (upsampled)} + \text{residual}$$

Does this mean we need to store both residuals and the blurred copies of the original?

**Slide Credit:** Ioannis (Yannis) Gkioulekas (CMU)
Constructing a **Laplacian** Pyramid

Algorithm

repeat:
  filter
  compute residual
  subsample
until min resolution reached

---

**Slide Credit:** Ioannis (Yannis) Gkioulekas (CMU)
Constructing a **Laplacian Pyramid**

What is this part?

**Algorithm**

repeat:
  filter
  compute residual
  subsample
until min resolution reached

---

*Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)*
Constructing a **Laplacian Pyramid**

It's a Gaussian Pyramid

**Algorithm**

repeat:
  filter
  compute residual
  subsample
until min resolution reached

**Slide Credit:** Ioannis (Yannis) Gkioulekas (CMU)
Reconstructing the Original Image

Algorithm

repeat:
    upsample
    sum with residual
until orig resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Gaussian vs Laplacian Pyramid

Which one takes more space to store?

Shown in opposite order for space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Aside: Image Blending

Aside: Image Blending

Aside: Image Blending

Algorithm:

1. Build Laplacian pyramid LA and LB from images A and B

2. Build a Gaussian pyramid GR from mask image R (the mask defines which image pixels should be coming from A or B)

3. From a combined (blended) Laplacian pyramid LS, using nodes of GR as weights: $LS(i,j) = GR(i,j) \times LA(i,j) + (1 - GR(i,j)) \times LB(i,j)$

4. Reconstruct the final blended image from LS
Aside: Image Blending

left

right

mask

blended
Aside: Image Blending

© david dmartin (Boston College)
Aside: Image Blending

© Chris Cameron
Oriented Pyramids

Laplacian pyramid is orientation independent

**Idea:** Apply an oriented filter at each layer
- represent image at a particular scale and orientation
- Aside: We do not study details in this course
Oriented Pyramids

Filter Kernels

Image

Coarsest scale

Finest scale

Forsyth & Ponce (1st ed.) Figure 9.13
Oriented Pyramids

Oriental Filters

Forsyth & Ponce (1st ed.) Figure 9.14