Image representation, sampling and quantization

António R. C. Paiva

ECE 6962 – Fall 2008

1 Image representation

Image as a function

- An image is a function of the space.

- Typically, a 2-D projection of the 3-D space is used, but the image can exist in the 3-D space directly.

Image as a single-valued function

- The function can be single-valued

\[ f : \mathbb{R}^m \rightarrow \mathbb{R}, \quad m = 2, 3, \]

quantifying, for example, intensity.
Image as a *multi-valued* function

- ... or, be multi-valued, \( f : \mathbb{R}^m \rightarrow \mathbb{R}^3, m = 2, 3 \), in which each corresponds to a color intensity.

2-D vs. 3-D images
Images are analog

- Notice that we defined images as functions in a continuous domain.
- Images are representations of an analog world.
- Hence, as with all digital signal processing, we need to digitize our images.

2 Digitalization of images

Digitalization

- Digitalization of an analog signal involves two operations:
  - Sampling, and
  - Quantization.
- Both operations correspond to a discretization of a quantity, but in different domains.

Sampling

- Sampling corresponds to a discretization of the space. That is, of the domain of the function, into $f : [1, \ldots, N] \times [1, \ldots, M] \rightarrow \mathbb{R}^m$. 

\[ \text{Figure 2.17} \ (a) \text{ Continuous image projected onto a sensor array,} \ (b) \text{ Result of image sampling and quantization.} \]
• Thus, the image can be seen as matrix,

\[ f = \begin{bmatrix}
  f(1,1) & f(1,2) & \cdots & f(1,M) \\
  f(2,1) & f(2,2) & \cdots & f(2,M) \\
  \vdots & \vdots & \ddots & \vdots \\
  f(N,1) & f(N,2) & \cdots & f(N,M)
\end{bmatrix}. \]

• The smallest element resulting from the discretization of the space is called a **pixel** (picture element).

• For 3-D images, this element is called a **voxel** (volumetric pixel).

**Quantization**

• Quantization corresponds to a discretization of the intensity values. That is, of the co-domain of the function.

• After sampling and quantization, we get \( f : [1, \ldots, N] \times [1, \ldots, M] \rightarrow [0, \ldots, L] \).

• Quantization corresponds to a transformation \( Q(f) \)
Typically, 256 levels (8 bits/pixel) suffices to represent the intensity. For color images, 256 levels are usually used for each color intensity.

**Digitalization: summary**

**FIGURE 2.16**
Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

3  **Changes in resolution**

**Which resolution?**

- Digital image implies the discretization of both spatial and intensity values. The notion of resolution is valid in either domain.

- Most often it refers to the resolution in sampling.
  - Extend the principles of multi-rate processing from standard digital signal processing.

- It also can refer to the number of quantization levels.

**Reduction in sampling resolution**

- Two possibilities:
– Downsampling

– Decimation

Increase in sampling resolution

• The main idea is to use interpolation.

• Common methods are:
  – Nearest neighbor
  – Bilinear interpolation
  – Bicubic interpolation
Decrease in quantization levels
To mitigate the “false contour” effect we can use *dither*.

- Basically, we add noise before quantization to create a more natural distribution of the new intensity values.

![Original - Undithered - Dithered Images](Images from Wikipedia.)

4 Matlab tutorial

Reading images

- Use `imread` to read an image into Matlab:

```matlab
» img = imread('peppers.jpg','jpg');
» whos
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Bytes</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>img</td>
<td>512x512x3</td>
<td>786432</td>
<td>uint8</td>
</tr>
</tbody>
</table>

8
– Format is: \( A = \text{IMREAD}(	ext{FILENAME}, 	ext{FMT}) \). Check the \texttt{help imread} for details.
– Note that data class is \texttt{uint8}. Convert to double with \( \text{img} = \text{double(img)}; \). This is necessary for arithmetic operations.

### Displaying images

- With Image Processing toolbox: use \texttt{imshow} to display the image.

  ```matlab
  imshow(img);
  imshow(img(:,:,1));
  \% Shows only the red component of the image
  ```

  – The image must be in \texttt{uint8} or, if double, normalized from 0 to 1.

- Without the Image Processing toolbox: use \texttt{image} to display the image.

  ```matlab
  image(img);
  ```

  – The image must have 3 planes. So, for grayscale images do,

  ```matlab
  image(repmat(gray_img, [1 1 3]));
  ```

### Saving images

- Use \texttt{imwrite} to save an image from Matlab:

  ```matlab
  imwrite(img,’peppers2.jpg’,’jpg’);
  imwrite(img(:,:,1),’peppersR.jpg’,’jpg’);
  \% Saves only the red component of the image
  ```

  – Format is: \texttt{IMWRITE(A,FILENAME,FMT)}. Check the \texttt{help imwrite} for details.

  – The image should be in \texttt{uint8} or, if double, normalized from 0 to 1.

### Reading

#### Reading

- Sections 2.4 and 2.5 of the textbook.