Boston University
CAS CS 585:
Image and Video Computing

Image Formats

Slides are part of 1st Lecture by Margrit Betke
1/18/2024
Learning Objectives

- Understand formats of images used as inputs to computer vision models.
  Pixel values may be greyscale, color, or attenuation (medical scans)

- Know how to access a single pixel in an image
- Know how to convert color images into greyscale images
- Know about standard computer vision library: OpenCV
What is an image?

- Images are fields of colored dots
- Each dot is called a **pixel** = picture cell
- Standard test image with detail, shading, texture, sharp & blurry regions:

Lena Soderberg ‘72
Controversy!

Slide credit: Diane Theriault
Images can be gray scale, color, or color with an alpha (transparency) channel.

Most common color representation is RGB (Red, Green, Blue). This is the representation used to put pixels on the screen.

Other models include CMYK (used for print) and YUV (often used for input from cameras, compression, and transmission).
What is an image?

- Images are 2 dimensional arrays of data, with an associated width, height, and color depth.
- Images typically use one byte per color channel per pixel.
- Gray images have 1 color channel. RGB images have 3 color channels. RGBA images have 4 color channels.

Slide credit: Diane Theriault
Digital Image File Formats

**Image:**

<table>
<thead>
<tr>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
</tr>
<tr>
<td>= Map</td>
</tr>
<tr>
<td>= Raster</td>
</tr>
<tr>
<td>of Pixel Values</td>
</tr>
</tbody>
</table>

**Size of table, color, compression scheme**

- Gray-scale images: generally 1 byte per pixel
- Color images: 3 numbers (each 1 byte) per pixel
- Medical images, e.g., CT, MRI: typically 2 bytes per voxel
Example: PGM Image

<table>
<thead>
<tr>
<th>Image file</th>
<th>Image ??</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td></td>
</tr>
<tr>
<td>3  3  255</td>
<td></td>
</tr>
<tr>
<td>0  255  0</td>
<td></td>
</tr>
<tr>
<td>220  0  20</td>
<td></td>
</tr>
<tr>
<td>0  130  0</td>
<td></td>
</tr>
</tbody>
</table>
Example: PGM Image

<table>
<thead>
<tr>
<th>Image file</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>255</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>255</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>220</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>130</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
Example: PGM Image

<table>
<thead>
<tr>
<th>Image file</th>
<th>Image</th>
</tr>
</thead>
</table>

```
P2
3 3 255
0 255 0
220 0 20
0 130 0
```

Note how indistinguishable
Light: Electromagnetic Waves

Wavelength $\lambda$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>450</td>
<td>500</td>
<td>550</td>
<td>600</td>
<td>650</td>
</tr>
</tbody>
</table>

Visible Range

X-rays  | Violet | Blue   | Yellow  | Green  | Red     | Radio    |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50 nm</td>
<td>400</td>
<td>450</td>
<td>500</td>
<td>550</td>
<td>600</td>
<td>650</td>
</tr>
</tbody>
</table>
RGB Color Space

Additive Space

- Cyan
- Green
- Yellow
- Blue
- Magenta
- White
- Red
- Black
### Example: PPM Image

**Image file**

```
P3
3 3 255
```

```
0 0 0 255 0 0 0 0 0
0 255 0 0 0 0 255 255 0
0 0 0 0 0 255 0 0 0
0 0 0 0 0 255 0 0 0
```

**Image ??**
Example: PPM Image

<table>
<thead>
<tr>
<th>Image file</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>3 3 255</td>
<td></td>
</tr>
<tr>
<td>0 0 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 255 255 0 0 255 0 0 0 255 0 0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>
How do I get at the data?

- Some image-handling APIs have nice interfaces, but speed can be a problem.

- You will probably have to handle the bytes of data directly at some point.

Slide credit: Diane Theriault
How do I get at the data?

- X = desired row
- Y = desired column
- C = color channel (red, green, blue, …).
- Bpp = Bytes per pixel (color channels)
- Image data is normally stored in row major order
- Note that there may be multiple values associated with each (x,y) pixel

**Data(x,y,c) = y \cdot (width \cdot Bpp) + x \cdot Bpp + c**

Slide credit: Diane Theriault
“Quick and dirty” conversion: Grab the Green Channel

- Average R, G, B: \((R+G+B)/3\)
- Max(R, G, B)
- Weigh them: \(0.3*R + 0.6*G + 0.1*B\)

Slide credit: Diane Theriault
Hue-Saturation-Value (HSV) Color Space

V=(R+B+G)/3 or V= max(R,G,B)

H=
\frac{\pi}{3} \frac{B-R}{V-\text{min}} + \frac{2\pi}{3}

H=
\frac{\pi}{3} \frac{R-G}{V-\text{min}} + \frac{4\pi}{3}

S=(V-\text{min}(R,G,B))/V
Image File Formats

- PPM / PGM is maybe simplest file format ever, but not supported by Photoshop or MS Image Viewer. Uncompressed. ASCII mode lets you open the image in a text editor.
- BMP: Microsoft’s uncompressed image format
- GIF: Images are compressed using run-length encoding to reduce the number of colors used. Previously licensed, now open
- JPEG: Images are compressed by removing high frequency information (also uncompressed version)
Tools of the Trade

- OpenCV is a widely used open-source computer vision library started by Intel
- Provides libraries for image I/O, video I/O and camera capture
- Industrial strength computer vision and image processing implementations
- “Quick and dirty” GUI toolkit

Slide credit: Diane Theriault
Learning Objectives

- Understand formats of images used as inputs to computer vision models.
  Pixel values may be greyscale, color, or attenuation (medical scans)

- Know how to access a single pixel in an image
- Know how to convert color images into greyscale images
- Know about standard computer vision library: OpenCV