Image Processing I

15-463: Rendering and Image Processing
Alexei Efros

...with most slides shamelessly stolen from Steve Seitz and Gonzalez & Woods

Today

Image Formation
Range Transformations
  • Point Processing

Programming Assignment #1 OUT

Reading for this week:
  • Gonzalez & Woods, Ch. 3
  • Forsyth & Ponce, Ch. 7
Image Formation

\[ f(x,y) = \text{reflectance}(x,y) \times \text{illumination}(x,y) \]
Reflectance in \([0,1]\), illumination in \([0,\infty]\)

Sampling and Quantization
What is an image?

We can think of an image as a function, $f$, from $\mathbb{R}^2$ to $\mathbb{R}$:

- $f(x, y)$ gives the intensity at position $(x, y)$
- Realistically, we expect the image only to be defined over a rectangle, with a finite range:
  - $f: [a, b] \times [c, d] \rightarrow [0, 1]$

A color image is just three functions pasted together. We can write this as a “vector-valued” function:

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$
What is a digital image?

We usually operate on **digital (discrete)** images:
- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)

If our samples are $\Delta$ apart, we can write this as:

$$f[i, j] = \text{Quantize}\{ f(i \Delta, j \Delta) \}$$

The image can now be represented as a matrix of integer values:

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>76</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>58</td>
<td>197</td>
</tr>
<tr>
<td>176</td>
<td>135</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>89</td>
</tr>
<tr>
<td>255</td>
<td>252</td>
</tr>
<tr>
<td>166</td>
<td>63</td>
</tr>
<tr>
<td>127</td>
<td>17</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
An image processing operation typically defines a new image $g$ in terms of an existing image $f$. We can transform either the range of $f$:

$$g(x, y) = t(f(x, y))$$

Or the domain of $f$:

$$g(x, y) = f(t_x(x, y), t_y(x, y))$$

What kinds of operations can each perform?

---

Point Processing

The simplest kind of range transformations are these independent of position $x,y$:

$$g = t(f)$$

This is called point processing.

What can they do? What’s the form of $t$?

**Important**: every pixel for himself – spatial information completely lost!
Basic Point Processing

Figure 3.3 Some basic gray-level transformation functions used for image enhancement.

Figure 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2.1).
(Courtesy of G.E. Medical Systems.)
Log

Power-law transformations
Gamma Correction

Gamma Measuring Applet:  
http://www.cs.berkeley.edu/~efros/java/gamma/gamma.html

Image Enhancement
Contrast Stretching

Image Histograms
Cumulative Histograms

Histogram Equalization

Figure 3.18: Transformation functions (1) through (4) were obtained from the histograms of the images in Fig. 3.17(a) using Eq. (3.3.8).
Histogram Matching

Match-histogram code

```python
Match-histogram (im1, im2)
im1-cdf = Make-cdf(im1)
im2-cdf = Make-cdf(im2)
inv-im2-cdf = Make-inverse-lookup-table(im2-cdf)
Loop for each pixel do
    im1[pixel] =
        Lookup(inv-im2-cdf,
        Lookup(im1-cdf, im1[pixel]))
```
Neighborhood Processing (filtering)

Q: What happens if I reshuffle all pixels within the image?

A: It's histogram won’t change. No point processing will be affected…

Need spatial information to capture this.

Programming Assignment #1

Easy stuff to get you started with Matlab
  • James will hold tutorial this week

Distance Functions
  • SSD
  • Normalized Correlation

Bells and Whistles
  • Point Processing (color?)
  • Neighborhood Processing
  • Using your data (3 copies!)
  • Using your data (other images)