CS 89.15/189.5, Fall 2015

Computational Aspects of Digital Photography

Assignment 0: C++ Refresher

Wojciech Jarosz
wojciech.k.jarosz@dartmouth.edu
First programming assignment

Programming assignment 0 available on class website

- just a warm up
- familiarize yourself with C++ and the basecode
- compile
- change brightness/contrast of an image
C++
Why C++?

More efficient than Java (compilation, memory)
- Ridiculously more efficient than Python

Standard language for many domains where performance matters (graphics, imaging)

Good experience
Online resources

http://cs.brown.edu/courses/cs123/docs/java_to_cpp.shtml


and many more...
Images in C++
Digital images

Can be encoded as 3D arrays
- 2D (x,y) grid of pixels
- for each (x,y), have a number of channels (e.g. R, G, B)

Formally:
- Domain: 2D plane
- Range: RGB space

Other color spaces possible

Values often encoded as 8- or 16-bit integers ([0..255] or [0..65535])
- But we will use floats in [0..1] to make life simpler

Modeled after a slide by Frédo Durand
Arrays

C++ vector
- dynamically sized
- templatized by type, float in our case
- e.g.: data = std::vector<float>(size, initialValue);

Array3D (array3d.h)
- our templatized wrapper to access a C++ vector as a 3D array
1D to 2D

vectors only have one 1D index

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

turn 2D index into 1D through strides

- pixel at x, y stored at y * width + x

2D array

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

1D vector encoding

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

Modeled after a slide by Frédo Durand
1D to 3D

vectors only have one 1D index

$\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
\end{array}$

likewise for 3D where $z$ is the color channel

- pixel at $x,y,z$ stored at $z*\text{width}*\text{height} + y*\text{width} + x$

- other choices possible, e.g. $y*\text{width}*\text{depth} + x*\text{depth} + z$
  - why favor one over the other?

Modeled after a slide by Frédo Durand
Our FloatImage class

Inherits from Array3D<float>

Stores a vector of pixel values

- size width*height*numChannels

size & number of dimensions

- could be used to represent 2D images with single channel, or even 1D arrays

access data using operator(...), e.g.: image(x,y,z)
File formats

We’ll use PNG

- simple to read, no big library needed
- only lodepng.(h|cpp) in ext subdirectory
- easy to convert to/from other formats

We’ll talk about how JPEG and other formats work later
Programming Assignment 0

Just a warm up

familiarize yourself with C++, the FloatImage class

compile

change brightness & contrast of an image
Next…

History of photographic technology
Slide credits

Frédo Durand