COS 426: PRECEPT 2

Riley Simmons-Edler
Assignment 1: Image Processing

• Structure of the assignment
• Implementation of filters operations
  – Luminance
  – Color
  – Filter
  – Composite
  – Resampling
Structure

cos426-assign1

Folders
- css
- images
- js
- libjs
- results

Developer
- batch.html
- index.html
- morphLines.html
- writeup.html

Other
- marker.json

- batch.js
- filters.js
- gui.js
- image.js
- main.js
- morphLines.js
- parser.js
- student.js
- writeup.js
Structure

• Interactive Mode
  – Photolist (edit it in Gui.js)
  – morphLines

• Batch Mode
  – Gui to Batch
  – Brightness Animation
  – newTab
    • Multiple parameters
    • Multiple images
    • Gif
    • Art
Morph Lines

- Read two images and create your own morph lines correspondence.
- You could modify your morphlines by including `&marker = yourmakerfile` to load it in.
- Read JSON in your code
  ```python
  linek = lines.initial[k]
  - linek.x0, linek.y0, linek.x1, linek.y1,
  ```
Implementation

• Graphica Obscura
  \[\text{out} = (1 - \alpha)\times\text{in0} + \alpha\times\text{in1}\]
- brightness:

- problem: it does not make great use of the full range of the slider
Brightness

Ratio < 0: interpolate with black
Ratio > 0: interpolate with white

See wiki_GIMP_contrast_brightness
Contrast

- Map $[-1, 1)$ to $[0, \infty)$ by
  \[ \text{Ratio} = \tan((\text{Ratio}+1)\pi/4) \]

interpolate with gray

See [wiki_GIMP_contrast_brightness](https://example.com/wiki_GIMP_contrast_brightness)
Saturation

• Map \([-1, 1]\) to \([0, 2]\) by
  \[\text{Ratio} = \text{Ratio} + 1;\]

interpolate with grayscale image

See wiki_GIMP_contrast_brightness
Gray

• Either way is ok:
  – Luminance (standard for certain color spaces): $0.2126*R + 0.7152*G + 0.0722*B$
  – Luminance (perceived option 1): $0.299*R + 0.587*G + 0.114*B$

From piazza@11
Gamma Correction

\[ V_{out} = Math.pow(V_{in}, \gamma) \]

- \( V_{in} \) is the rgb values in \([0,1]\), the result pixel is \( V_{out} \times 255 \)
Vignette

innerR = 0.5 - 0.5 * value[0];
outerR = 0.5 + 0.5 * value[1];
Example: value=[0.5,0.5], innerR=0.25, outerR=0.75

Pixel outside outerR is black
Pixel inside innerR is clear
White balance

- First, map RGB to [0,1]
- RGB -> LMS
  - divided by $L_w M_w S_w$
- LMS -> RGB
- Map back to 0-255
Histogram Matching

• Tips: Choose a reasonable reference image.
Gaussian Filter

• Tips:
  – Weight should be normalized.
  – Border pixels
  – Create new image
Edge

• **Tips:**
  – Weight should not be normalized.
  – Border pixels
  – Create new image
Sharpen

**Tips:**
- Weight could be normalized.
- Border pixels
- Create new image
Median

- RGB vs Luminance
- Bilateral
Bilateral

• Color sigma
  – calculate the distance in rgb [0,1]
• Weighted should be normalized
• Make two sigmas more equalized
Sampling

• Create a new image
• Rotation:
  – Set the alpha of outside pixel as 0
• Swirl:
  – For the outside pixels, find its nearest pixel inside the photo.
More tips

• Don’t worry about minor difference with results in example page.
  – contrast, quantize…
  – Just make sure your results are reasonable.
• Which rgb range this operation should process in. [0,1] or 0-255?
• Need to create new images?
• No 256
Q&A