

6/23 画像信号処理特論

Schedule

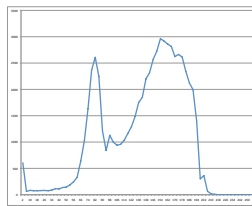
- 5/12 "Hello World!" of image processing
- 5/19 Image filtering
- 5/26 Binarization
- 6/2 (Prof. Tehrani)
- 6/9 (Prof. Tehrani)
- 6/16 (Prof. Tehrani)
- 6/23 Histogram ← 1st report deadline
- 6/30 Discrete Cosine Transform
- 7/7 JPEG
- 7/14 (Prof. Fujii)
- 7/21 (Prof. Fujii) ← 2nd report deadline

Today's Issue

- Create a histogram



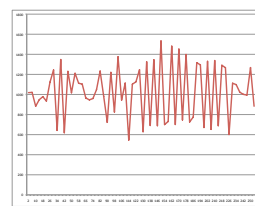
Input



Output

Today's issue

- Histogram Equalization



Histogram

- Histogram consists of bins
 - Pixel values are quantized into several levels
- Each bin has
 - Range of pixel values
 - Number of pixels

Creating a histogram

```
int binsize;    // the quantization step
int nBins = 256/binsize; // number of bins

int * hist = new int [nBins]; // histogram data

// initialize
for idx = 0; idx < nBins; idx++
    hist[idx]=0;
```

Creating a histogram

```
// count pixels
For all pixels (x,y)
    int value = I(x,y);
    int bin = value/binsize;

    // into which bin the value falls
    hist[bin]++;

    // increase the count by one
```

Output the histogram

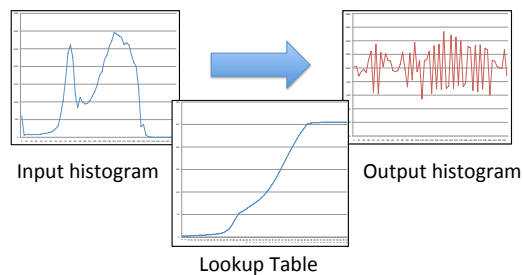
```
for idx = 0; idx < nBins; idx++
    int mid = idx*binsize+binsize/2;

    // median of the bin
    fprintf(fp, "%d\t%d\n", mid, hist[idx]);

    // write the median and pixel count
```

Histogram equalization

- Correspondence between the input and output values



Histogram equalization

- Create a table representing the correspondence

| | | | | | | | | |
|--------|---|---|---|---|-----|-----|-----|-----|
| input | 0 | 1 | 2 | 3 | ... | 127 | ... | 255 |
| output | 0 | 0 | 1 | 2 | ... | 100 | ... | 250 |

```
int table[256]; // to keep correspondences
```

Histogram equalization

- Convert pixel values according to the table

```
int table[256]; // to keep correspondences
For all pixels (x,y)
    int value = I(x,y);
    O(x,y)=table[value];
```

Histogram equalization

- Design of the table

```
int hist[256]; //histogram with 256 bins
int sum = 0; // num of pixels not greater than value
for value = 0; value < 256; value++
    sum += hist[value];
    double r = sum/(W*H);

    // ratio of pixels not greater than value
    table[value] = round(255*r);
```

Exercises

- Build and execute “sample3” and try histogram generation with several images
- Implement [histogram equalization](#) and compare the images and histograms before and after the equalization
- How can a histogram be used [for image binarization](#)? Think and try.
 - Hint: Otsu’s Method, percentile method