Kyungpook National University

Artificial Brain Research - ABR

BEP

Lesson and Assignment 2

Statistics and Histogram

Gwenaelle Cunha Sergio

Winter 2014

1 Summary of Lecture Material

1.1 Statistics

Statistics is the study of the collection, organization, analysis, interpretation, and presentation of data.

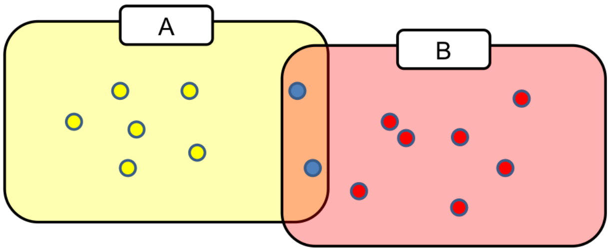


Figure 1: Collection of Data

Take the collection of data A. We have that:

$$Mean(A) = \sum A(n)$$

Variance is a term that indicates how far the elements of a collection are spread out.

$$\begin{aligned} Variance(A) &= E((X-\mu)^2) \\ &= \frac{1}{N}\sum(A(n) - Mean(A))^2 \end{aligned}$$

Covariance is a "measure of how much two random variables change together" [1]. N is the total number of elements in A and M in B.

$$Covariance(A,B) = \sum N \sum M (A(n) - Mean(A)) * (B(m) - Mean(B))$$

Correlation "refers to any of a broad class of statistical relationships involving dependence" [2], any statistical relationship between two sets of data.

$$Correlation\_coefficient = Covariance(A,B)/STD(A) * STD(B)$$

Joint probability is the probability that A and B happen, $P(A \cap B)$. Conditional probability is the probability that A occurs given that B has already occurred, $P(A|B)$.

$$P(A \cap B) = \frac{blue}{N+M}$$
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

[1] http://en.wikipedia.org/wiki/Covariance

[2] $http://en.wikipedia.org/wiki/Correlation\_and\_dependence$

The notion of dependent and independent events is very simple. The first one referes to events that affect the following events and the second one is referent to ones that don't affect the following events.

A probability density function (pdf) is a function that describes the likelihood of a certain event happening on a given time.

Cumulative distribution function (cdf) is a function that "describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x" [3].

1.2 Histogram

A histogram is a graphical representation of the distribution of data, as shown in **Fig. 2**. Histograms are important because they can be used to find properties and pre-process images. Next, we're going to mention some histogram applications.

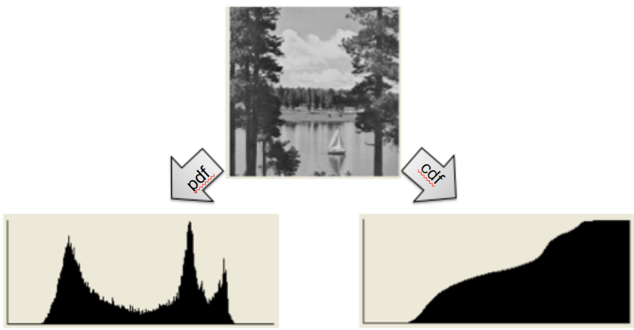


Figure 2: pdf and cdf

1.2.1 Sliding

Consists on moving a histogram by an offset value, **Fig. 3**.

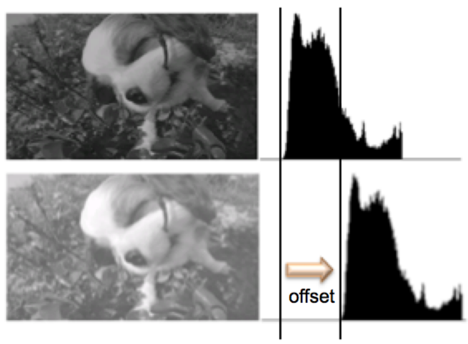


Figure 3: Sliding

[3] $http://en.wikipedia.org/wiki/Cumulative\_distribution\_function$

1.2.2 Stretching

Used to normalize values of the histogram from 0 to 255, **Fig. 4**.

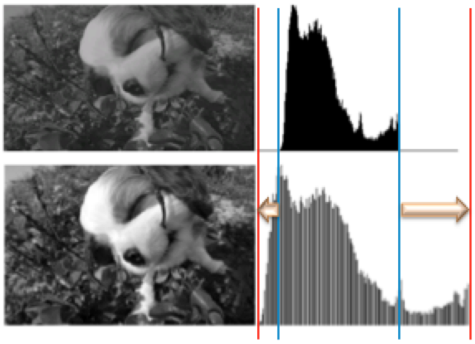


Figure 4: Stretching

1.2.3 Shrink

Normalizes the values of the histogram to a specified min and max, **Fig. 5**.

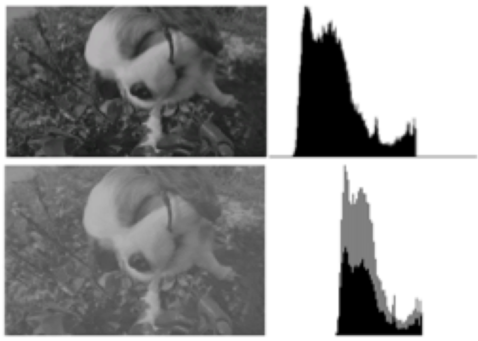


Figure 5: Shrink

1.2.4 Equalize

Adjusts the image's contrat, **Fig. 6**. The cdf is increases uniformly and the values are normalized between 0 and 255.

1.2.5 Quantization

Represents an image using a relatively smaller discrete set of values, like less pixels, colors or intensity, **Fig. 7**. Some of the methods to determine the threshold value at quantization are: Dynamic Bayesian Networks and Reveal [4] and Self-Organizing Lloyd-Max Histogram Quantization [5].

[4] $http://www.biomedcentral.com/1471-2105/6/S4/S11$

[5] $http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.8619\&rep=rep1\&type=pdf$

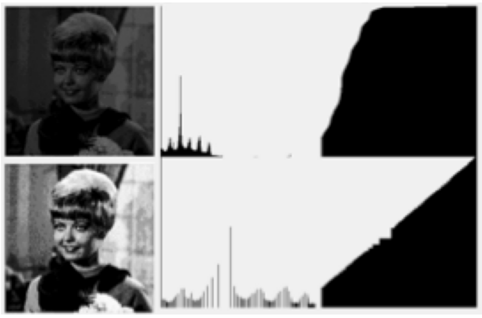


Figure 6: Equalize

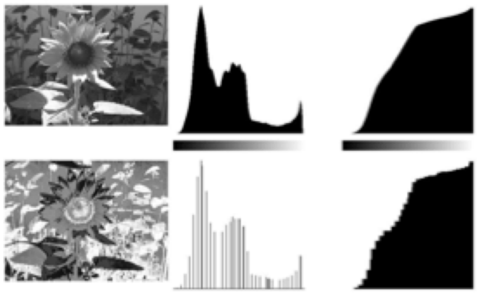


Figure 7: Quantization

2 Assignment

2.1 Implement histogram methods

2.1.1 Sliding

The algorithm for the sliding histogram method is:

```
int offset = 30;
int sliding = 0;
for (int i = 0; i < mat.rows; i++){
    for (int j = 0; j < mat.cols; j++){
        sliding = mat.at<uchar>(i,j) + offset;
        if (sliding < 0) {
            sliding = 0;
        } else if (sliding > 255) {
            sliding = 255;
        }
        slideImage.at<uchar>(i,j) = sliding;
    }
}
```

Fig. 8 shows a demonstration of its execution. From top left to bottom right, we have the original image and the resulting images for the offset equals to 30, 90 and 150, respectively.

Figure 8: Sliding

2.1.2 Stretching

The algorithm for the stretching histogram method consists of only one function, *normalize* and a demonstration of its execution is shown in **Fig. 9**.

```
cv::normalize(mat, stretchImage, 0, 255, CV_MINMAX);
```

Figure 9: Stretching

2.1.3 Shrink

Similarly to the stretching algorithm, the shrinking histogram method consists of only one function, *normalize* and a demonstration of its execution is shown in **Fig. 10**.

```
int min = 0, max = 255;
int minImage = 80, maxImage = 230;
cv::normalize(mat, shrinkImage, minImage, maxImage, CV_MINMAX);
```

2.1.4 Equalization

The equalization histogram method consists of only one function, *equalizeHist* and a demonstration of its execution is shown in **Fig. 11**.

```
equalizeHist(mat, equalizeImage);
```

Figure 10: Shrink

(a) Girl

(b) Woman

Figure 11: Equalize

2.1.5 Method to determine the threshold value of quantization

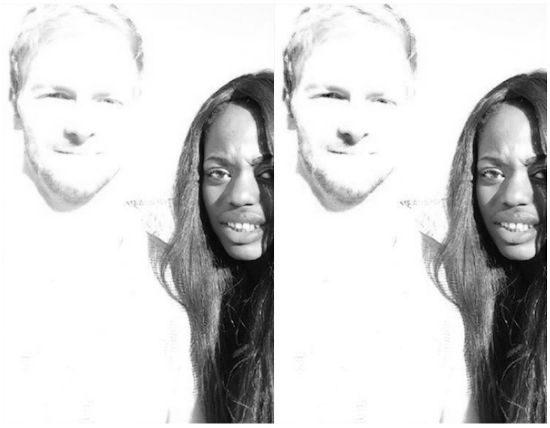
This algorithm uses the *LUT* function and its demonstration is shown in **Fig. 12**.

```
size_t div = 32;
uchar buffer[256];
for (size_t i = 0; i != 256; ++i){
    buffer[i] = i / div * div + div/2;
}
cv::Mat table(1, 256, CV_8U, buffer, sizeof(buffer));
cv::LUT(mat, table, quantizationImage);
```

Figure 12: Quantization

2.2 Local histogram equalization

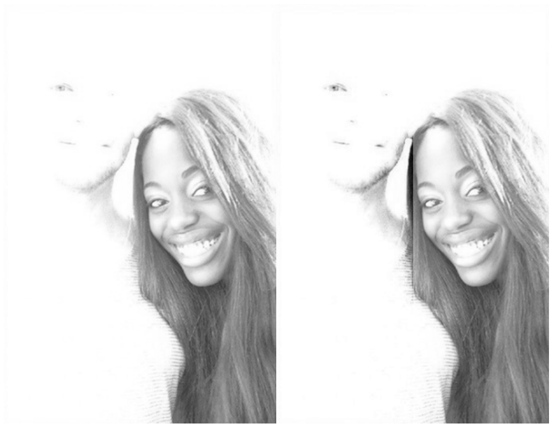
This section can be done using the stretching and equalizing methods already implemented. In **Figs. 13(a)** and **13(d)** there's need to apply a stretching technique. As for **Figs. 13(b)** and **13(c)**, there's need for an equalization on the right half of the image.



(a) Couple 1: stretch

(b) Couple 2: equalize

(c) Couple 3: equalize



(d) Couple 4: stretch

Figure 13: Couple result