Universiteit van Amsterdam

Beeldbewerken

Image Processing

February 15, 2016

Student:
Ajit Jena
5730066

Lars Lokhoff
10606165

Docent:
R. v.d. Boomgaard

Cursus:
Beeldbewerken

Vakcode:
5062NURP6Y

1 Introduction

In image science, image processing is the manipulation of input images with mathematical operations to generate a different output image. In this paper we will be focusing on two types of image processing, namely:

- Histogram equalization
- Image warping

1.1 Histogram Equalization

Histogram equalization is used to increase the global contrast of an image. The intensities of the original histogram are distributed across the whole histogram. This makes it that the areas with lower contrast gain a higher contrast. Histogram equalization is useful in images with both very dark or light backgrounds. It can also lead to a better view of bone structures in x-ray images and it can help show better detail in images with over or under exposure.

1.2 Image Warping

The two warping methods that will be used for this assignment will be Affine- and Perspective transformations. The transformations (might) require rotation, stretching and/or shearing of an input image, to produce the proper output image.

Affine Transform

The affine transform will warp a parallelogram within an input image (consisting of four points) to an output image consisting of the parallelogram form the input image. This transformation will make use of rotation and

stretching. The grey value of points that are not known will be calculated with the use of interpolation. For the assignment a tilted parallelogram will be transformed to a square shaped parallelogram.

Perspective Transform

The perspective transform will also warp a parallelogram within an image, however in this case shearing of the input parallelogram will be needed to obtain the needed output image. For the assignment a flyer laying on the floor will be transformed to a rectangular output image only showing the flyer.

2 Theory

In this section the underlying theories and mathematics of achieving the various image manipulations will be discussed.

2.1 Histogram Equalization

Histogram equalization is a point operator such that the histogram of the resulting image is of constant. It is mostly used to correct lighting in images because it can increase the global contrast of an image.

Definition

Let $f : D \rightarrow [0, 1] \ \epsilon \ R$ be a scalar image with histogram h_f. Histogram equalization creates an image point operator Ψ such that the resulting image of Ψf has a constant histogram, meaning that all scalar values in the resulting image are equally probable. In other words, the values of the original image are spread across all possible values $[0, 1]$.
How is Ψ constructed? Because it is assumed that Ψ is a point transform it can be said that $\Psi f(x) = \psi(f(x))$ and becuase the ordering of scalar values is preserved it can be said that $v_1 \leq v_2 \Rightarrow \Psi(v_1) \leq \Psi v_2$. Also, the $p - th$ perscentage of image f should be equal to the p-th percentage of $g = \Psi f$ because the ordering is preserved. In terms of cumultative histograms this can be written as $H_f(v) = H_g(\psi(v))$. Because the histogram is flat and the of the image run from 0 to 1 $H_g(v) = v$ and $\psi(v) = H_f(v)$.

Practical Use

Histogram equalization serves as a preprocessing step in vision systems to compensate for uknown changes in illumination. For instance when an object has to be compared to another object and the lighting in the images is different. This can cause the program to make a faulty decision.

2.2 Image Warping

The two image warping methods (also known as geometrical transformations) that will be explained are the Affine- and Perspective- transform. Both methods require linear algebra to obtain the final result.

Affine Transform

An affine transform requires coördinate (x, y) to be transformed from the input image to the output image coördinate (x', y'). This transformation can be represented by the following formula:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The affine transformation matrix is the 3×3 matrix in the formula above. By obtaining this matrix (since it is invertible) the original coördinate (x, y) can be obtained with the output coördinate (x', y'). The reason that the input coördinate is sought from the output coördinate, is because not every input coördinate corresponds exactly with an output coördinate. When the output coördinate $(0, 10)$ corresponds with the input coördinate $(12.6, 9.7)$ for example the input coördinate does not ´exist´ since the image matrix is discrete. Interpolation is used to determine the value at the input coördinate. When multiple coördinates have to be transformed, three non collinear points are needed to determine the affine transformation matrix. The relationship between a given coördinate (x'_i, y'_i) and the affine transformation matrix is as follows:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix}$$

Since the transformation matrix has to be determined for three points the relationship can be stacked upon eachother as follows:

$$\begin{pmatrix} x'_1 \\ y'_1 \\ x'_3 \\ y'_2 \\ x'_3 \\ y'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix}$$

or:

$$\mathbf{q} = M\mathbf{p}$$

In which vector $\mathbf{p}$ is the vector needed to determine the affine transformation matrix. Instead of using the inverse to determine $\mathbf{p}$, the least squares method is used to determine $\mathbf{p}$, because determining three non collinear points often is inaccurate. To obtain $\mathbf{p}$ the following formula can be used:

$$\mathbf{p} = (M^T M)^{-1} M^T \mathbf{q}$$

With the vector $\mathbf{p}$ the affine transformation matrix can be made and any point in the input image can be transformed to the output image using the transformation matrix.

Perspective Transform

The perspective transformation matrix is similar to the affine transformation, however in the case of a perspective transformation shearing also occurs therefore the following formula is used:

$$s \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

the scaling factor s is needed to compensate for any unwanted stretching caused by the addition of g, h and i. The transformation can be rewritten to the following:

$$x' = \frac{ax + by + c}{gx + hy + i}$$
$$y' = \frac{dx + ey + f}{gx + hy + i}$$

This can be reformulated to:

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x'x - x'y - x' \\ 0 & 0 & 0 & x & y & 1 & -y'x - y'y - y' \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Another difference between affine transform and perspective transform is that four quadrilateral points are used. Stacking up these points gives the

following:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x - x'_1y - x' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x - y'_1y - y' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x - x'_2y - x' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x - y'_2y - y' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x - x'_3y - x' \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x - y'_3y - y' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x - x'_4y - x' \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x - y'_4y - y' \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

or:

$$M\mathbf{p} = 0$$

If all points are not collinear the homogeneous system of equations can be solved, because there will be one non trivial solution for $\mathbf{p}$

If there are more than four point correspondances the singular decomposition value (or SVD) of M is used to compensate for unwanted noise. The vector $\mathbf{p}$ has to be obtained minimizes the absolute length of M$\mathbf{p}$ subject to the constraint that the length of $\mathbf{p}$ is equal to 1. The following steps can be taken to obtain $\mathbf{p}$:

$$min||UDV^T\mathbf{p}|| \text{ s.t. } ||\mathbf{p}|| = 1$$
$$min||DV^T\mathbf{p}|| \text{ s.t. } ||\mathbf{p}|| = 1$$

U is orthogonal and therefore perserves the norm V is also orthogonal. Inserting $\mathbf{q} = V\mathbf{p}$ gives the following:

$$min||D\mathbf{q}|| \text{ s.t. } ||V\mathbf{q}|| = 1$$
$$min||D\mathbf{p}|| \text{ s.t. } ||\mathbf{q}|| = 1$$

D is a diagonal matrix with the sorted singular values on the diagonal and therefore $\mathbf{p} = V\mathbf{q}$ which is the last column of V since $\mathbf{q} = (00...1)^T$. Now that the vector $\mathbf{p}$ is known, the perspective transformation can be determined ($\mathbf{v}$ = (a,b,c...i)^T).

3 Implementation

Now that the theory and mathematics have been discussed the implementation of the various methods will be discussed.

3.1 Histogram Equalization

To implement the histogram equalization a small modification was needed for the original algorithm given in the assignment. Instead of letting the range go from 0 to 1, the range needs to go from 0 to M. This way the algorithm works for an arbitrary range $[0, M]$, although M can be 255 maximum.

3.2 Image Warping

Affine Transform

To obtain the affine transformation matrix A the vector $\mathbf{p}$ must be determined in the following formula (explained in the theory section):

$$\mathbf{p} = (M^T M)^{-1} M^T \mathbf{q}$$

Matrix M is the 6×6 matrix containing the three points. In the implementation X and Y are the width and height of the output image respectively. Vector $\mathbf{q}$ contain the points to which the points in $\mathbf{p}$ transform to. $\mathbf{q} = (0, Y-1, 0, 0, X-1, 0)^T$, which transforms points (x_1, y_1) to $(0, Y-1)$(bottom left corner), (x_2, y_2) to $(0, 0)$(top left corner) and (x_3, y_3) to $(X-1, 0)$(top right corner) (the ´-1´ is added to match the array indeces).

Using the least squares method, vector $\mathbf{p}$ is calculated and then reshaped to a 2×3 matrix which is the affine transformation matrix A. With the cv2.warpAffine() function and A the original image is warped into the output image of dimensions $X \times Y$.

Perspective Transform

For the perspective transform first the matrix M is created:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x - x'_1y - x' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x - y'_1y - y' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x - x'_2y - x' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x - y'_2y - y' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x - x'_3y - x' \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x - y'_3y - y' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x - x'_4y - x' \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x - y'_4y - y' \end{pmatrix}$$

where:

$$(x'_1, y'_1) = (0, 0)(topleftcorner)$$
$$(x'_2, y'_2) = (X, 0)(toprightcorner)$$
$$(x'_3, y'_3) = (X, Y)(bottomrightcorner)$$
$$(x'_4, y'_4) = (bottomleftcorner)$$

The SVD is calculated with the built in function svd(M). With the last column of the V matrix vector $\mathbf{p}$ is determined. Vector $\mathbf{p}$ is reshaped into a 3×3 matrix which is the perspective transformation matrix. With use of the obtained matrix and the function cv2.warpPerspective() the input image is transformed into the output image with dimensions $X \times Y$.

4 Results

In this section the results of the implemented algoriths will be shown. The resulting images and histograms will be shown and discussed to show the end results. For the histogram assignment the answers to the 2 questions are also discussed in this section.

4.1 Histogram Equalization

This section is split up in 4 parts, respectively the parts of the assignment.

Part 1: Equalization

As shown in the theory section, the expression $\psi(v) = H_f(v)$ was given for the range $[0, 1]$. When the range is changed to $[0, M]$, the expression can be easily changed to $M * \psi(v) = H_f(v)$.

Part 2: Code

Figure 1 shows the result of part 2, where the histogram function is applied to the photgraph of the man with the old camera. As can be seen in the second picture the contrast of the image has changed. The changes can also be seen in figure 2 where the before and after histograms are shown.

Figure 1: Before and after equalization

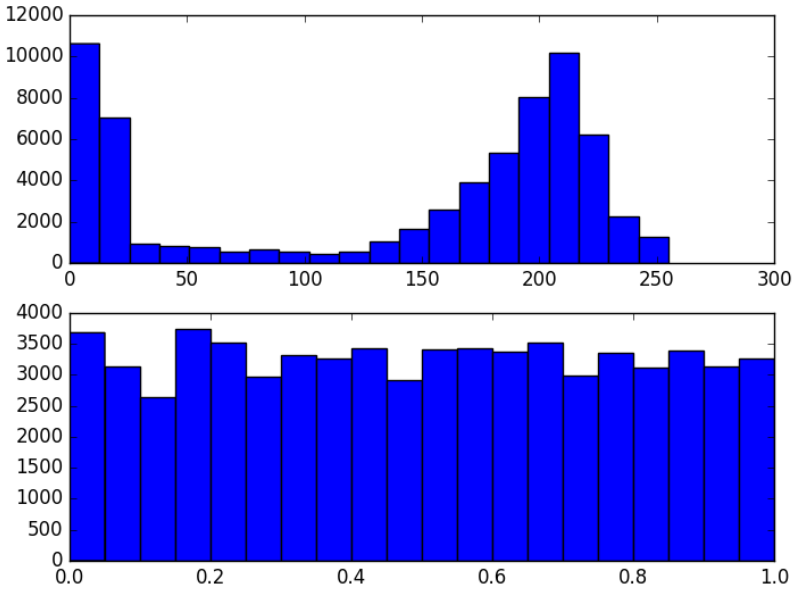

Figure 2: Before and after equalization histograms

Part 3: Practical Use

Figure 3 (see next page) shows the result using histogram equalization on 4 different images made of the same object but each with different angles and lighting. As can be seen the images all show the same contrast.

Part 4: Table Lookup

The efficiency of a simple interpolater on a list xs (sorted) is log(N) with N being the number of samples. The efficiency is achieved by using a binary search tree to find the two closest points which can be used to calculate the weighted average. The average is then used to interpolate. This results in an overall efficiency log(N).

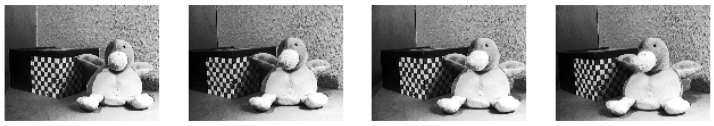

Figure 3: Before and after equalization of different angle/lighting images

4.2 Image Warping

Here the results of the implementations of the two warping methods will be displayed.

Affine Transform

Figure 4: Before and after Affine transformation

Perspective Transform

Figure 5: Before and after Perspective transformation 1

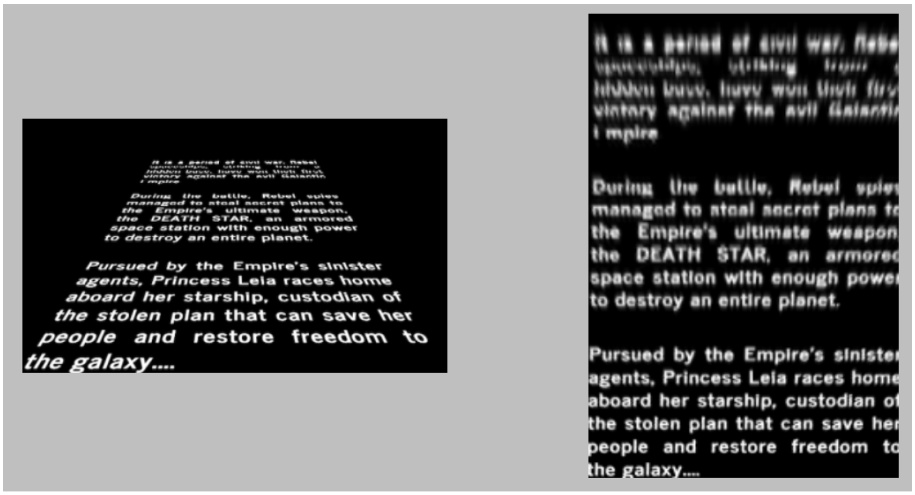

Figure 6: Before and after Perspective transformation 2