# 中国科学技术大学计算机学院
# 《数字图像处理与分析》实验报告

2021.06.15

实验题目：图像处理实验

学生姓名：胡毅翔

学生学号：PB18000290

计算机实验教学中心制

2019 年 9 月

# 目录

# 1 实验目的

本实验的目的是通过实验进一步理解和掌握数字图像处理与分析的原理和方法。通过分析、实现现有的图像处理算法，学习和掌握常用的图像处理与分析技术。

# 2 实验环境

1. PC 一台
2. Windows 10 操作系统
3. Vivado 2019.1
4. Visual Studio Code 1.56.2

# 3 实验内容

数字图像处理的实验内容主要包括五个方面:

1. 图像几何变换。
2. 对图像进行空间域滤波，提高图像视觉质量, 以便于人眼观察、理解或用计算机对其进一步处理。
3. 对图像作频域变换, 进行频率域滤波增强处理。
4. 在空间域和频域提取、描述和分析图像中所包含的特征, 便于计算机对图像作进一步的分析和理解, 经常作为模式识别和计算机视觉的预处理。这些特征包括很多方面, 如图像的频域特性、边界特征等。
5. 了解常见的图像退化模型和相应的图像恢复算法，从本质上改善图像质量；对图像进行分析，采用阈值法、区域分裂合并法等分割算法，获取图像中感兴趣目标区域。

# 4 实验一 图像几何变换

## 4.1 图像的平移

### 4.1.1 实验原理

图像平移就是将图像中所有的点都按照指定的平移量水平、垂直移动。如：设 $(x_0, y_0)$ 为原图像的一点，图像水平平移量为 tx, 垂直平移量为 ty, 则平移后坐标变为 $(x_1, y_1)$, 显然，$(x_0, y_0)$ 和 $(x_1, y_1)$ 有如下关系:

$$\begin{cases} x_1 = x_0 + tx \\ y_1 = y_0 + ty \end{cases}$$

用矩阵表示如下:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathrm{tx} \\ 0 & 1 & \mathrm{ty} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

### 4.1.2 实验内容

输入一幅图像，根据输入的水平和垂直平移量，显示平移后的图像。

输入水平偏移量 100，垂直偏移量 100，得到的结果为:

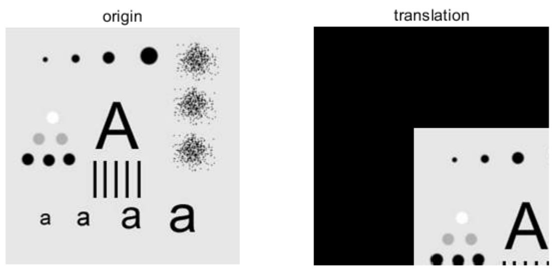


图 1: 图像的平移

## 4.2 图像的旋转

### 4.2.1 实验原理

图像绕中心点（原点）旋转的公式如下: $\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$ 图像如果绕一个指定点 $(a, b)$ 旋转，则先要将坐标系平移到该点，再进行旋转，然后平移回新的坐标原点。则旋转变换表达式为:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathrm{a} \\ 0 & 1 & \mathrm{b} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\mathrm{a} \\ 0 & 1 & -\mathrm{b} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ y_0 \\ 1 \end{bmatrix}$$

### 4.2.2 实验内容

输入一幅图像，根据输入的旋转角度参数，绕图像中心点旋转，分别用最近邻插值和双线性插值显示旋转后的图像。

输入旋转角度参数 60，得到的结果为：

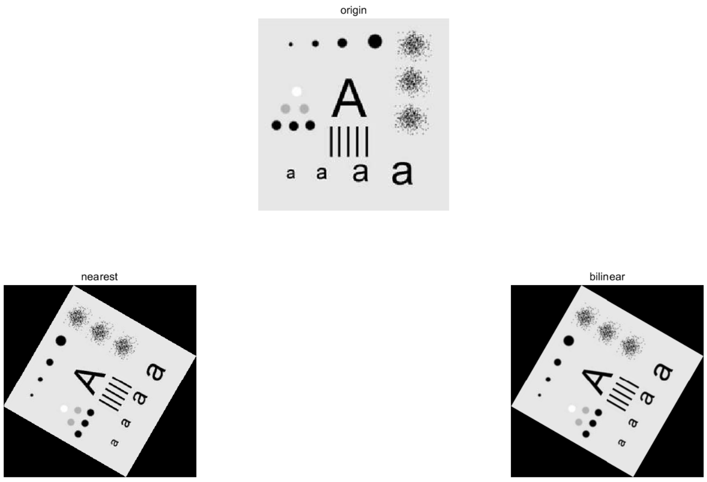


图 2: 图像的旋转

## 4.3 图像的缩放

### 4.3.1 实验原理

假设图像 x 轴方向缩放比率为 c, y 轴方向缩放比率为 d, 那么原图中，点 $(x_0, y_0)$ 对应于新图中的点 $(x_1, y_1)$ 的转换矩阵为:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} c & 0 & 0 \\ 0 & \mathrm{d} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ y_0 \\ 1 \end{bmatrix}$$

### 4.3.2 实验内容

输入一幅图像，根据输入的水平和垂直缩放量，分别用最近邻插值和双线性插值，显示缩放后的图像。

水平方向缩放为原来的 1/3，垂直方向缩放为原来的 1/2，得到的结果为：

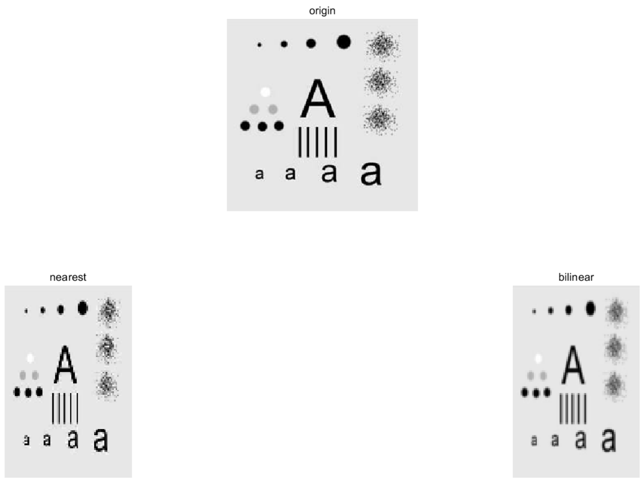


图 3: 图像的缩放

## 4.4 图像几何失真校正

### 4.4.1 实验原理

设原图像: $\mathrm{f(x,y)}$ 失真图像:$\mathrm{g}$ $\mathrm{(x',y')}$

$$\mathrm{x'=s(x,y)}$$

$$\mathrm{y'=t(x,y)}$$

通过选取控制点对, 求出 $\mathrm{(x',y')}$ 与 $\mathrm{(x,y)}$ 坐标之间的关系

1. 线性失真:

$$\mathrm{x'=a_1x+a_2y+a_3}$$

$$\mathrm{y'=b_1x+b_2y+b_3}$$

6 个系数要 3 个控制点对为减少误差, 选取 n>3 个控制点对

$$\begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ . & . & . \\ . & . & . \\ . & . & . \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$
\mathrm{M}=\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ . & . & . \\ . & . & . \\ . & . & . \\ x_n & y_n & 1 \end{bmatrix}
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}=\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ \vdots \\ x'_n \end{bmatrix}
\quad
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}=\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\begin{bmatrix} y'_1 \\ y'_2 \\ . \\ . \\ y'_n \end{bmatrix}
$$

2. 双线性失真:

$$
\begin{cases} x' = a_1xy + a_2x + a_3y + a_4 \\ y' = b_1xy + b_2x + b_3y + b_4 \end{cases}
$$

8 个系数要 4 个控制点对为减少误差, 选取 n>4 个控制点对

$$
\mathrm{M}=\begin{bmatrix} x_1y_1 & x_1 & y_1 & 1 \\ x_2y_2 & x_2 & y_2 & 1 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_ny_n & x_n & y_n & 1 \end{bmatrix}
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}=\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ \vdots'_n \end{bmatrix}
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}=\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\begin{bmatrix} y'_1 \\ y'_2 \\ . \\ . \\ y'_n \end{bmatrix}
$$

3. 二次校正式:

$$
x' = a_1x^2 + a_2y^2 + a_3xy + a_4x + a_5y + a_6
$$

$$
y' = b_1x^2 + b_2y^2 + b_3xy + b_4x + b_5y + b_6
$$

12 个系数要 6 个控制点对为减少误差, 选取 n>6 个控制点对

$$
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}=\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ \dot{x}'_n \end{bmatrix}
$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}} \begin{bmatrix} y_1' \\ y_2' \\ . \\ . \\ y_n' \end{bmatrix}$$

$$\mathrm{M} = \begin{bmatrix} {x_1}^2 & {y_1}^2 & x_1y_1 & x_1 & y_1 & 1 \\ {x_2}^2 & {y_2}^2 & x_2y_2 & x_2 & y_2 & 1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ {x_n}^2 & {y_n}^2 & x_ny_n & x_n & y_n & 1 \end{bmatrix}$$

### 4.4.2 实验内容

输入图像 alphabet1.jpg 及几何失真图像 alphabet2.jpg，设置控制点进行几何失真校正，显示校正后的图像。

选取 4 个控制点：

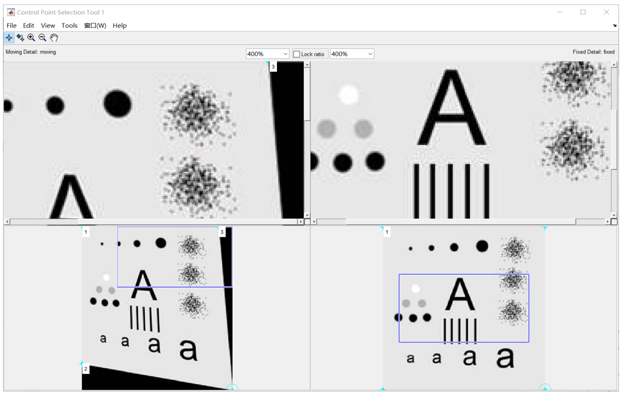

图 4: 控制点的选取

得到的校正结果：

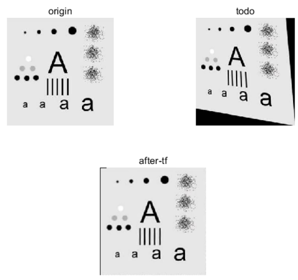


图 5: 图像几何失真校正

# 5 实验二 图像点处理增强

## 5.1 灰度的线性变换

### 5.1.1 实验原理

灰度的线性变换就是将图像中所有的点的灰度按照线性灰度变换函数进行变换。该线性灰度变换函数是一个一维线性函数:

$$f(x) = f_A \cdot x + f_B$$

灰度变换方程为:

$$D_B = f\left(D_A\right) = f_A \cdot D_A + f_B$$

其中参数 $f_A$ 为线性函数的斜率，$f_B$ 为线性函数的在 y 轴的截距，$D_A$ 表示输入图像的灰度，$D_B$ 表示输出图像的灰度。

### 5.1.2 实验内容

输入一幅图像，根据输入的斜率和截距进行线性变换，并显示。

关键函数为:

```
function [new] = LinearTransformFunc(original, k, d)
    new = original;
    [a, b] = size(original);

    for i = 1:a

        for j = 1:b
            tmp = original(i, j) * k + d;

```

```
            if tmp > 255
                tmp = 255;
            elseif tmp < 0
                tmp = 0;
            end

            new(i, j) = tmp;
        end

    end

end
```

输入的斜率为 3，截距为 $-44$，结果为:

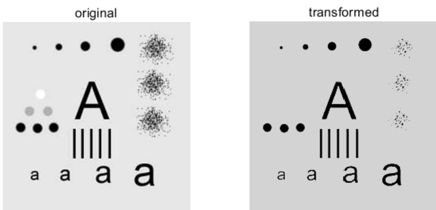


图 6: 灰度的线性变换

## 5.2 灰度拉伸

### 5.2.1 实验原理

灰度拉伸和灰度线性变换相似。不同之处在于它是分段线性变换。表达式如下:

$$f(x) = \frac{y_1}{x_1}x; \quad x < x_1$$

$$f(x) = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1; \quad x_1 \le x \le x_2$$

$$f(x) = \frac{255 - y_2}{255 - x_2}(x - x_2) + y_2; \quad x > x_2$$

其中, $(x_1, y_1)$ 和 $(x_2, y_2)$ 是分段函数的转折点。

### 5.2.2 实验内容

输入一幅图像，根据选择的转折点，进行灰度拉伸，显示变换后的图像。

关键函数为:

```
function [new] = StretchFunc(original, x1, y1, x2, y2)
    new = original;

    w = size(new, 1);
    h = size(new, 2);

    k1 = y1 / x1;

    dk1 = (y2 - y1) / (x2 - x1);
    dk2 = (255 - y2) / (255 - x2);

    for i = 1:w

        for j = 1:h
            x = new(i, j);

            if x < x1
                new(i, j) = k1 * x;
            elseif x < x2
                new(i, j) = dk1 * (x - x1) + y1;
            else
                new(i, j) = dk2 * (x - x2) + y2;
            end

            if new(i, j) > 255
                new(i, j) = 255;
            elseif new(i, j) < 0
                new(i, j) = 0;
            end

        end

```

```
    end

end
```

参数 $x_1, y_1, x_2, y_2$ 分别为 $30, 10, 200, 250$，得到的结果为：


图 7: 灰度拉伸

## 5.3 灰度直方图

### 5.3.1 实验原理

灰度直方图是灰度值的函数，描述的是图像中具有该灰度值的像素的个数，其横坐标表示像素的灰度级别，纵坐标表示该灰度出现的频率 (象素的个数)。

### 5.3.2 实验内容

输入一幅图像，显示它的灰度直方图，可以根据输入的参数（上限、下限）显示特定范围的灰度直方图。

参数上限、下限分别为 $100, 255$，得到的结果为：

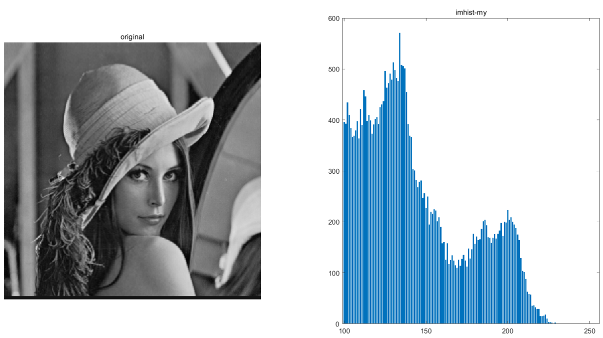


图 8: 灰度直方图

## 5.4 直方图均衡

### 5.4.1 实验原理

1. 统计图像中各灰度级像素个数 $n_k$；
2. 计算直方图: $p_r(r_k) = \frac{n_k}{n}$；
3. 计算累计直方图: $s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$；
4. 取整 $\mathrm{S_k} = \mathrm{int}\left[(\mathrm{L}-1)\mathrm{s_k} + 0.5\right.$；
5. 确定映射对应关系: $\mathrm{k} \rightarrow \mathrm{S_k}$；
6. 对图像进行增强变换 $(\mathrm{k} \rightarrow \mathrm{S_k})$。

   其中 L 是灰度等级，n 是图像总像素数。

### 5.4.2 实验内容

1. 显示一幅图像 pout. bmp 的直方图；
2. 用直方图均衡对图像 pout. bmp 进行增强；
3. 显示增强后的图像及其直方图；
4. 用原始图像 pout.bmp 进行直方图规定化处理，将直方图规定化为高斯分布；
5. 显示规定化后的图像及其直方图。

得到的结果为:

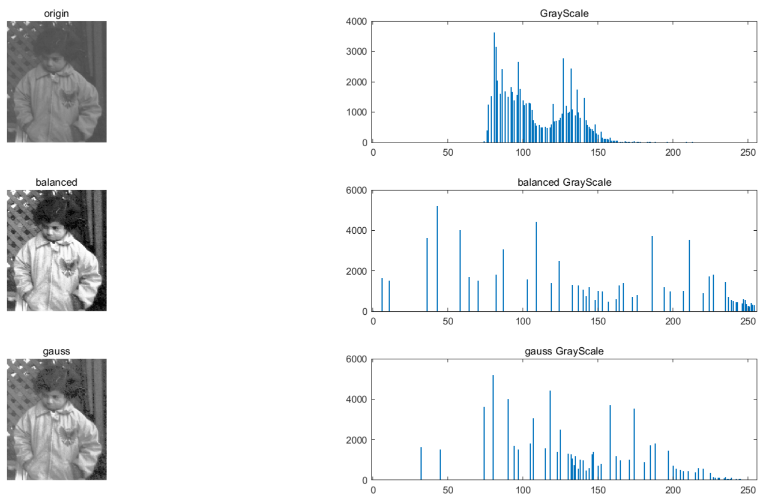


图 9: 直方图均衡