北京大学

本科实验报告

课程名称：机器感知实验

姓　　名：金镇雄

学　　院：元培

　　系：智能科学系

专　　业：智能科学与技术

年　　级：19

学　　号：1900094619

指导教师：曲天书

职　　称：副教授

2022 年　　4 月　　25 日

实验四、图像变换

一、 实验目的

- 使用 MATLAB 系统进行图像变换
- 图像变换包括形状变换和颜色变换
  - 图像变换：平移变换、尺度变换、旋转变换等
  - 颜色变换：冰冻效果、融炎效果、漫画效果、羽化效果、缩放模糊效果、LOMO 特效、电影效果等

二、 实验要求

- 界面清晰美观
- 可播放原始图像和变换的图像
- 实验结果分析
- 实验讨论

三、 实验仪器设备

PC 机、图像文件

四、 实验原理

1. 图像变换

[x,y]表示原始图像中像素的位置，[u,v]表示变换后的像素位置。

A. 平移变换

[tx,ty]为 x 轴和 y 轴的平移量

$$[u,v] = [x,y] + [tx,ty] = [x + tx, y + ty]$$

平移变换的变换矩阵为：

$$[u,v,1] = [x,y,1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ tx & ty & 1 \end{bmatrix}$$

B. 尺度变换

s_x, s_y为 x 轴和 y 轴尺度变换的大小

$$[u,v] = [s_x x, s_y y]$$

尺度变换的变换矩阵为：

$$[u,v,1] = [x,y,1]\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

C. 旋转变换

θ为旋转角

$$\begin{cases} u = xcos\theta - ysin\theta \\ v = xsin\theta + ycos\theta \end{cases}$$

旋转变换的变换矩阵为：

$$[\mathrm{u},\mathrm{v},1] = [\mathrm{x},\mathrm{y},1]\begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

D. 错切变换

sh_x, sh_y为错切变换量

$$\begin{cases} u = x + sh_x y \\ v = y + sh_y x \end{cases}$$

错切变换的变换矩阵为：

$$[\mathrm{u},\mathrm{v},1] = [\mathrm{x},\mathrm{y},1]\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. 颜色变换

R,G,B为变换后的像素值， r,g,b为变换前的像素值。

A. 漫画特效

漫画特效的像素替换公式为：

$$R = |g - b + g + r| * r / 256$$
$$G = |b - g + b + r| * r / 256$$
$$B = |b - g + b + r| * g / 256$$

B. 冰冻特效

冰冻特效的像素替换公式为：

$$R = r - g - b$$
$$G = g - b - r$$
$$R = b - r - g$$

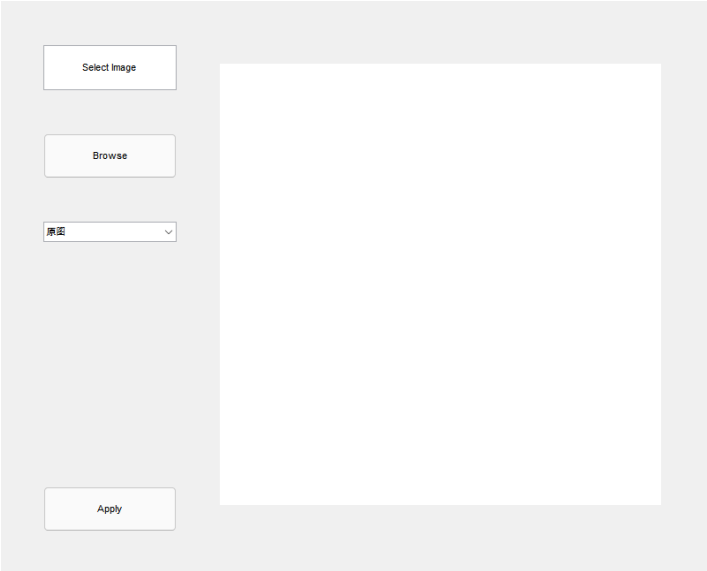
C. 熔炎特效

熔炎特效的像素替换公式为：

$$R = r * 128 / (g + b + 1)$$
$$G = g * 128 / (b + r + 1)$$
$$B = b * 128 / (r + g + 1)$$

五、 实验内容和步骤

1. 设计 GUI 界面



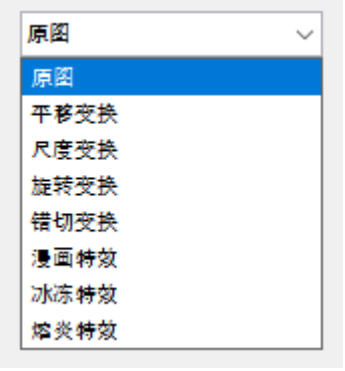
2. 读入原始图像

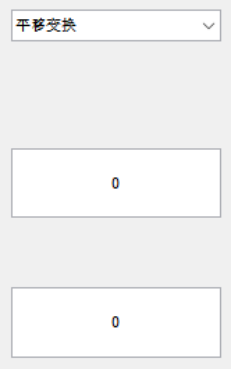
用 imread 函数读入图像，函数的返回值是大小为 height×width×3 的矩阵。并用 size 函数获取图像的宽和长。矩阵的每个元素为 uint8 类型的 RGB 值之一，其范围为 0~255。为了以后进行图像变换后复原原始图像，另外保存图像信息。

```
[file,path] = uigetfile('*.jpg');
str = path + "/" + file;
img = imread(str);
[R, C,~] = size(img);
ori_img = img;
ori_R = R;
ori_C = C;
```

3. 选定变换方法进行变换

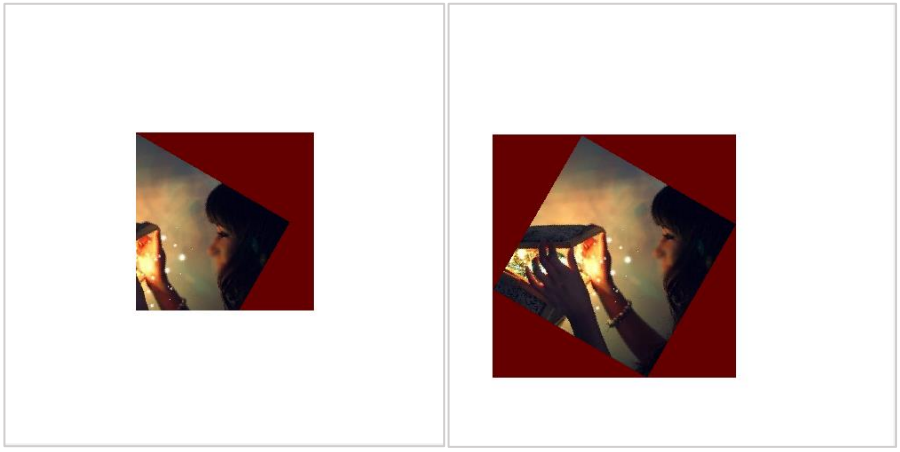
通过 GUI 的 popupmenu 选定变换方法，且选定后显示对应个数的输入框，例如，平移变换需要两个输入框（x 轴、y 轴上的平移量）；旋转变换需要一个输入框（旋转角度）；颜色变换不需要输入框。





根据变换方法对应的矩阵运算，定义变换矩阵，对原始图像的每个像素乘以变换矩阵，实现各种变换。需要注意，若平移量、旋转角等变换参数为 double 类型，则做矩阵乘法后其元素类型会变成 double 类型。而矩阵乘法的结果$[u,v,1]$中 u 和 v 表示变换后图像中像素的位置，应为整数，因此需要进行强制的类型转换，转换成 uint8 类型或 uint16 类型（若变换后图像的宽和长比 255 大，即超出 uint8 类型的范围，则需要用 uint16 类型）。

需要特别注意，旋转变换和错切变换需要注意其变换后的大小，否则变换后的图像会不完整。以旋转变换为例，以下两个图分别为对原始图像进行 30° 的旋转变换时不考虑大小变化和考虑大小变化的变换结果。可见，不考虑图像大小变化时，部分像素不能得到旋转变换而结果不完整。假设原始图像的行数和列数分别是R和C，则旋转α后，其行数变为$abs(Rsin\alpha)+abs(Ccos\alpha)$，列数变为 $abs(Csin\alpha)+abs(Rcos\alpha)$。



颜色变换中也需要注意类型。从原始图像中获取像素值时应先将其值转换成 double 类型，这样才能保证以后做像素变换时不丢失信息，否则变换结果会成为黑白图。另外，变换后需要将 double 类型的像素值转换成 uint8 类型。

```
for i = 1 : R
    for j = 1 : C
        r = double(img(i,j,1));
        g = double(img(i,j,2));
        b = double(img(i,j,3));
        if type == "manhua"
            r_ = uint8(abs(g-b+g+r)*r/256);
            g_ = uint8(abs(b-g+b+r)*r/256);
            b_ = uint8(abs(b-g+b+r)*g/256);
```

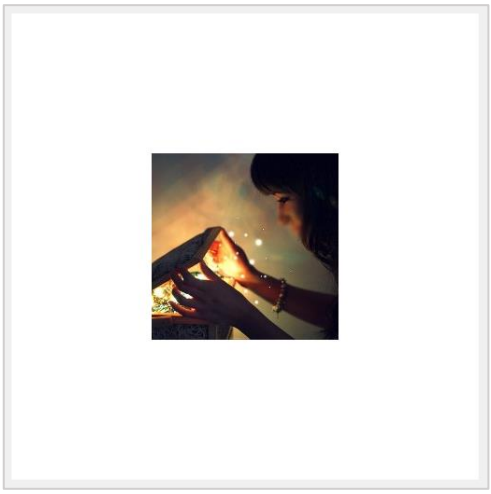
4. 输出变换后的图像

通过 MATLAB 中 imshow 函数输出原始图像以及变换后的图像。变换后图像的大小会发生变化，为了避免变换后的图像发生部分截断，应在图像周围扩充白色。其实现方法可采用图像平移变换，将图像平移到白图的中心处。输出图像时也需要保证图像

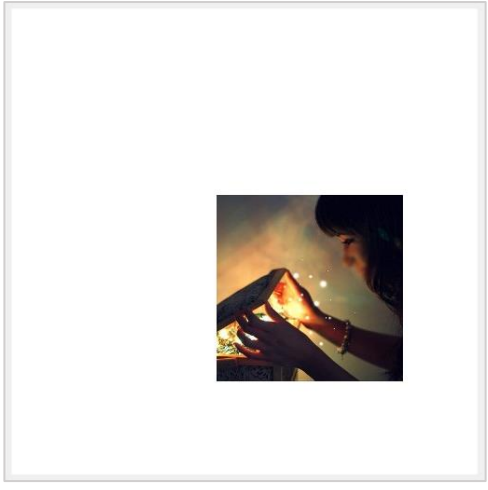
的类型为 uint8 类型，若图像为 double 类型的三维矩阵，则像素值大于 1 的部分都显示成白色。

六、实验结果和分析

1. 原始图像

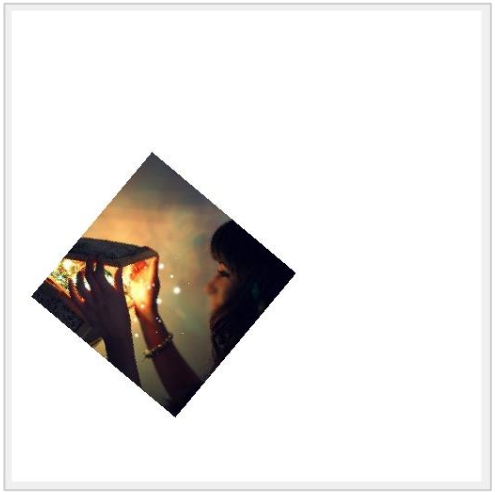


2. 平移变换：(50,70)



3. 尺度变换：(2,0.5)

4. 旋转变换：30°

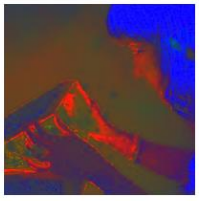


5. 错切变换：(0.2,0.2)

6. 漫画特效

7. 冰冻特效

8. 熔炎特效



七、讨论

在图像处理中，常常出翔图像显示成黑白图的情况，这是数据类型转换时出错导致的。因此，图像变换的过程中需注意像素的数据类型。

1. double 和 uint8 类型

   MATLAB 中 double 是默认的数据类型，允许双精度浮点数据和负数；uint8 是无符号整形数据，取值范围为 0 到2^8，即 0 到 255。

2. im2uint8 函数

   对于 double 类型数据，im2uint8 将其 0 到 1 部分重新映射到 0 到 255 之间，然后对于溢出的部分，就近处理为 0 或 255。

3. im2double 函数

   对于 uint8 类型数据，im2double 将 0 到 255 部分重新映射到 0 到 1 之间，这里 uint8 数据本身有溢出限制，因此不存在小于 0 或大于 1 的部分。

4. imread 函数

   图像文件经过 imread 函数读取后，获取到的图像数据类型为 uint8。

5. imshow 函数

   对于 uint8 数据，imshow 直接按照 0 到 255 将数值映射到灰度级上，0 对应纯黑，255 对应纯白，中间为渐变灰。对于 double 数据，imshow 仅保留其 0 到 1 之间的数值，并映射到灰度级上，0 对应纯黑，1 对应纯白，中间为渐变灰；对于溢出的部分，就近处理到 0 或 1，显示为纯黑或纯白。

假设用 imread 函数读取图像后得到的矩阵 A 为 uint8 类型，而用 zeros 函数得到的矩阵 B 的数据类型为 double，则直接将 A 的像素值付给 B 后用 imshow 函数显示的图像会出现纯黑或纯白的情况。

八、代码

```
function varargout = lab4(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @lab4_OpeningFcn, ...
                   'gui_OutputFcn',  @lab4_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
function lab4_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for lab4
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(handles.param1_text,'Visible','off')
set(handles.param2_text,'Visible','off')
global res;
init_bg()
imshow(res);


function varargout = lab4_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function menu_Callback(hObject, eventdata, handles)

m = get(hObject,'Value');

if m == 2
    set(handles.param1_text,'Visible','on')
    set(handles.param2_text,'Visible','on')
    set(handles.param1_text,'String','0');
    set(handles.param2_text,'String','0');
elseif m == 3
    set(handles.param1_text,'Visible','on')
    set(handles.param2_text,'Visible','on')
    set(handles.param1_text,'String','1');
    set(handles.param2_text,'String','1');
elseif m == 5
    set(handles.param1_text,'Visible','on')
    set(handles.param2_text,'Visible','on')
    set(handles.param1_text,'String','1');
    set(handles.param2_text,'String','1');
elseif m == 4
    set(handles.param1_text,'Visible','on')
    set(handles.param2_text,'Visible','off')
    set(handles.param1_text,'String','0');
else
    set(handles.param1_text,'Visible','off')
    set(handles.param2_text,'Visible','off')
end

global img;
if ~isempty(img)
    init_res()
end

function menu_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function param1_text_Callback(hObject, eventdata, handles)

function param1_text_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function param2_text_Callback(hObject, eventdata, handles)


function param2_text_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function applyButton_Callback(hObject, eventdata, handles)
global res;
global angle;
global shx;
global shy;
m = get(handles.menu,'Value');
x = str2double(get(handles.param1_text,'String'));
y = str2double(get(handles.param2_text,'String'));
init_bg()
if m == 2
    x = round(x);
    y = round(y);
    move_img(x,y)
elseif m == 3
    times_img(x,y)
elseif m == 4
    angle = angle + x;
    rotate_img(angle);
elseif m == 5
    shx = x * (1+shx);
    shy = y * (1+shy);
    sh_img(shx,shy);
elseif m == 6
    effect("manhua");
elseif m == 7
    effect("bingdong");
elseif m == 8
    effect("rongyan");
end
imshow((res))

function file_text_Callback(hObject, eventdata, handles)

function file_text_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function browseButton_Callback(hObject, eventdata, handles)
global img;
global R;
global C;
global ori_img;
global ori_R;
global ori_C;

[file,path] = uigetfile('*.jpg');
str = path + "/" + file;
img = imread(str);
[R, C,~] = size(img);
ori_img = img;
ori_R = R;
ori_C = C;


init_res();
set(handles.file_text,'String',file);


function init_res()
global img;
global R;
global C;
global res;
global p;
global q;
global ori_R;
global ori_C;
global ori_img;
global angle;
global shx;
global shy;
R = ori_R;
C = ori_C;
p = 250-round(R/2);
q = 250-round(C/2);
img = ori_img;
angle = 0;
shx = 1;
shy = 1;
init_bg();
move_img(0,0);
imshow((res));

function init_bg()
global res;
res = uint8(zeros(500, 500, 3));
for i = 1:500
    for j = 1:500
        res(i,j,1:3)=[255,255,255];
    end
end

function move_img(x,y)
```

```
global res;
global img;
global R;
global C;
global p;
global q;
p = p+x;
q = q+y;
tras = [1 0 p; 0 1 q; 0 0 1];
for i = 1 : R
    for j = 1 : C
        temp = [i; j; 1];
        temp = tras * temp;
        a = temp(1, 1);
        b = temp(2, 1);
        if (a <= 500) && (b <= 500) && (a >= 1) && (b >= 1)
            res(a,b,:) = img(i,j,:);
        end

    end
end

function times_img(sx,sy)
global p;
global q;
global img;
global R;
global C;
global res;

tras = [1/sx 0 0; 0 1/sy 0; 0 0 1];
r = round(sx * R);
c = round(sy * C);
tmp = zeros(r,c,3);
p = round(p-(R*(sx-1))/2);
q = round(q-(C*(sy-1))/2);
for i = 1 : r
    for j = 1 : c
        temp = [i; j; 1];
        temp = tras * temp;
        x = uint16(temp(1, 1));
        y = uint16(temp(2, 1));
        a = p+i;
        b = q+j;
         if (x <= R) && (y <= C) && (x >= 1) && (y >= 1)
            tmp(i,j,:) = img(x,y,:);
            if (a <= 500) && (b <= 500) && (a >= 1) && (b >= 1)
                res(a,b,:) = img(x,y,:);
            end
        else
            tmp(i,j,:) = [255, 255, 255];
            if (a <= 500) && (b <= 500) && (a >= 1) && (b >= 1)
                res(a,b,:) = [255, 255, 255];
            end
        end
    end
end
img = tmp;
R = r;
```

```
C = c;


function rotate_img(d)
global res;
global img;
global ori_img;
global ori_R;
global ori_C;
global R;
global C;
global p;
global q;
alpha = d * 3.1415926 / 180.0;
c1 = round(-R*sin(alpha));
c2 = round(C*cos(alpha));
r1 = round(C*sin(alpha));
r2 = round(R*cos(alpha));
if cos(alpha)*sin(alpha) >= 0
    if c1 > c2
        j1 = c2;
        j2 = c1;
    else
        j1 = c1;
        j2 = c2;
    end
    if r1+r2 > 0
        i1 = 0;
        i2 = r1+r2;
    else
        i1 = r1+r2;
        i2 = 0;
    end
else
    if c1+c2 < 0
        j1 = c1+c2;
        j2 = 0;
    else
        j1 = 0;
        j2 = c1+c2;
    end
    if r1 > r2
        i1 = r2;
        i2 = r1;
    else
        i1 = r1;
        i2 = r2;
    end
end
a = abs(i1)+abs(i2);
b = abs(j1)+abs(j2);
tmp = zeros(a,b,3);
tras = [cos(alpha) -sin(alpha) 0; sin(alpha) cos(alpha) 0; 0 0 1];

for i = i1 : i2
    for j = j1 : j2
        temp = [i; j; 1];
        temp = tras * temp;
        x = uint16(temp(1, 1));
```

```
        y = uint16(temp(2, 1));
        if (x <= ori_R) && (y <= ori_C) && (x >= 1) && (y >= 1)
            if (p+i <= 500) && (q+j <= 500) && (p+i >= 1) && (q+j >= 1)
                res(p+i,q+j,:) = ori_img(x,y,:);
            end
        else
            if (p+i <= 500) && (q+j <= 500) && (p+i >= 1) && (q+j >= 1)
                res(p+i,q+j,:) = [256,256,256];
            end
        end
    end
end
img = tmp;
R = a;
C = b;


function sh_img(shx,shy)
global R;
global C;
global res;
global img;
global p;
global q;
sh = [1 shy 0; shx 1 0; 0 0 1]';
for i = -round(R + shx*C) : 1 : 2*R + shx * C
    for j = -round(C + shy * R) : 1 : 2*C + shy * R - 20
        temp = [i; j; 1];
        temp = sh * temp;
        x = uint16(temp(1, 1));
        y = uint16(temp(2, 1));

        if (x <= R) && (y <= C) && (x >= 1) && (y >= 1)
            if (p+i <= 500) && (q+j <= 500) && (p+i >= 1) && (q+j >= 1)
                res(p+i, q+j,:) = img(x, y,:);
            end
        end
    end
end

function effect(type)
global img;
global res;
global R;
global C;
global p;
global q;
for i = 1 : R
    for j = 1 : C
        r = double(img(i,j,1));
        g = double(img(i,j,2));
        b = double(img(i,j,3));
        if type == "manhua"
            r_ = uint8(abs(g-b+g+r)*r/256);
            g_ = uint8(abs(b-g+b+r)*r/256);
            b_ = uint8(abs(b-g+b+r)*g/256);
        elseif type == "bingdong"
            r_ = uint8(abs(r-g-b));
            g_ = uint8(abs(g-b-r));
```

```
            b_ = uint8(abs(b-r-g));
        elseif type == "rongyan"
            r_ = uint8(r*128/(g+b+1));
            g_ = uint8(g*128/(b+r+1));
            b_ = uint8(b*128/(r+g+1));
        end
        if r_ < 0
            r_ = 0;
        elseif r_ > 255
            r_ = 255;
        end
        if g_ < 0
            g_ = 0;
        elseif g_ > 255
            g_ = 255;
        end
        if b_ < 0
            b_ = 0;
        elseif b_ > 255
            b_ = 255;
        end
        img(i,j,:) = [r_ g_ b_];
        res(p+i,q+j,:) = [r_ g_ b_];
    end
end
```