

多媒体实验二

班级：2016211303

姓名：黄若鹏

学号：2016212901

一、实验题目

图像处理

要求：

(1) 选取适当的图片素材和世界地图，运用各种选取方法制作一幅由世界名胜照片揉和在一起的背景。利用图层效果制作一幅有地形质感的世界地图。调整并合并所有层存储为各种图像文件格式并压缩。

(2) 显示一个 bmp 文件的 C 程序，并实现图像亮度、对比度调整、图像平移、放大、旋转和镜像。

提交内容：

图片素材、合成图片、显示 bmp 的程序代码、显示 bmp 的可执行文件、实验报告（电子版+打印版）。

二、实验环境

- (1) 操作系统：macOS
操作软件：Photoshop
- (2) 操作系统：windows10
操作软件：Qt5

三、实验设计

(1) 合成图片

- 1、寻找素材
- 2、导入 Photoshop
- 3、分别将不同图片导入不同图层，进行相应的处理
- 4、合成每一个图层，并导出图片

(2) 操作 bmp 文件程序

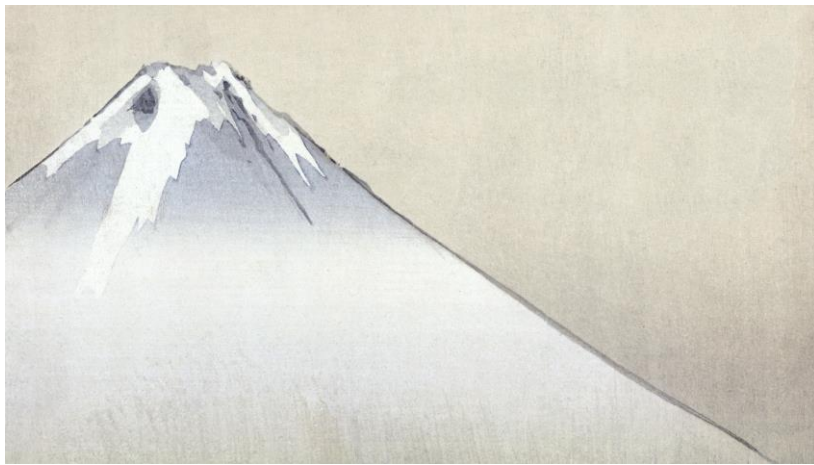
- 1、编写底层功能函数代码
- 2、图形化操作界面代码
- 3、测试程序代码
- 4、完善代码

四、实验过程

(1) 合成图片

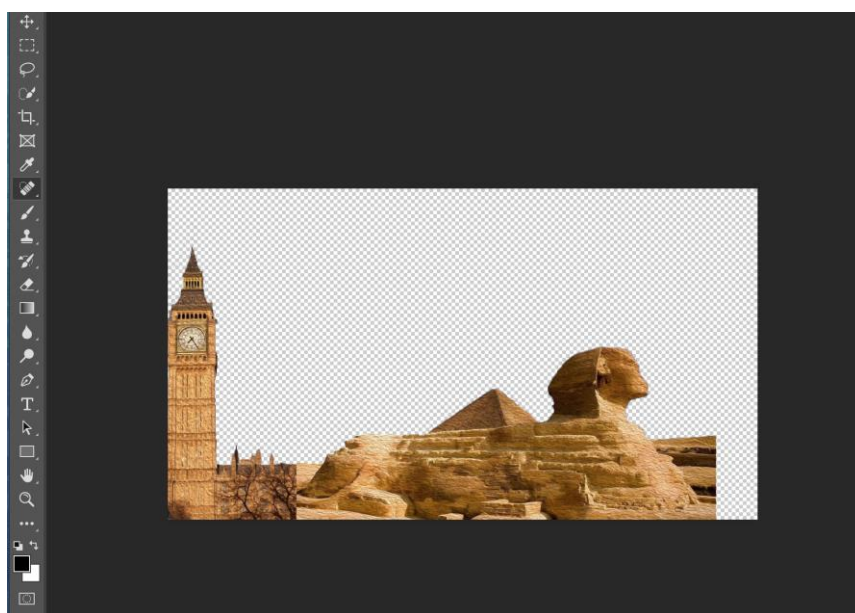
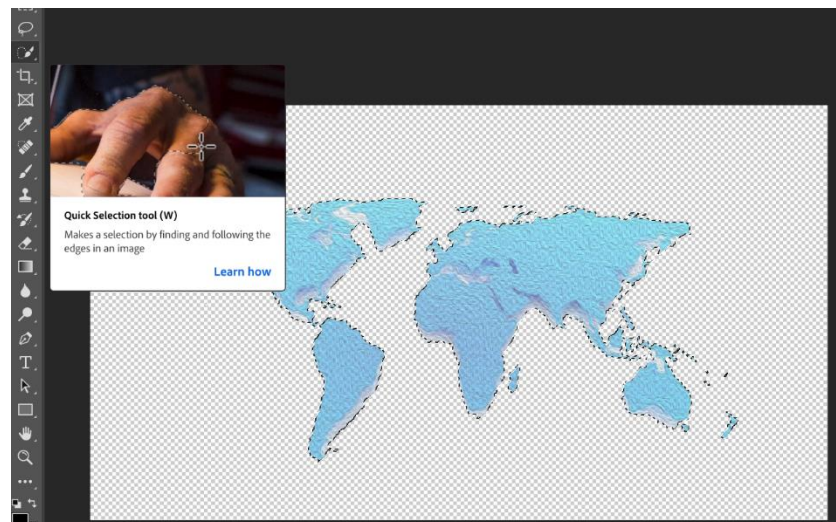
1、寻找素材

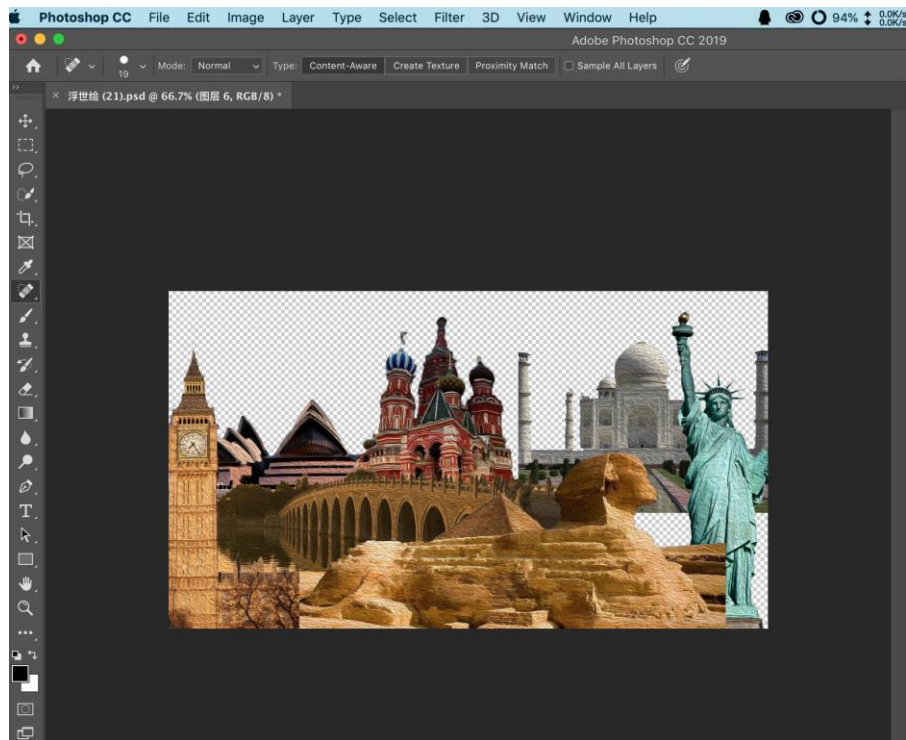
部分素材如下



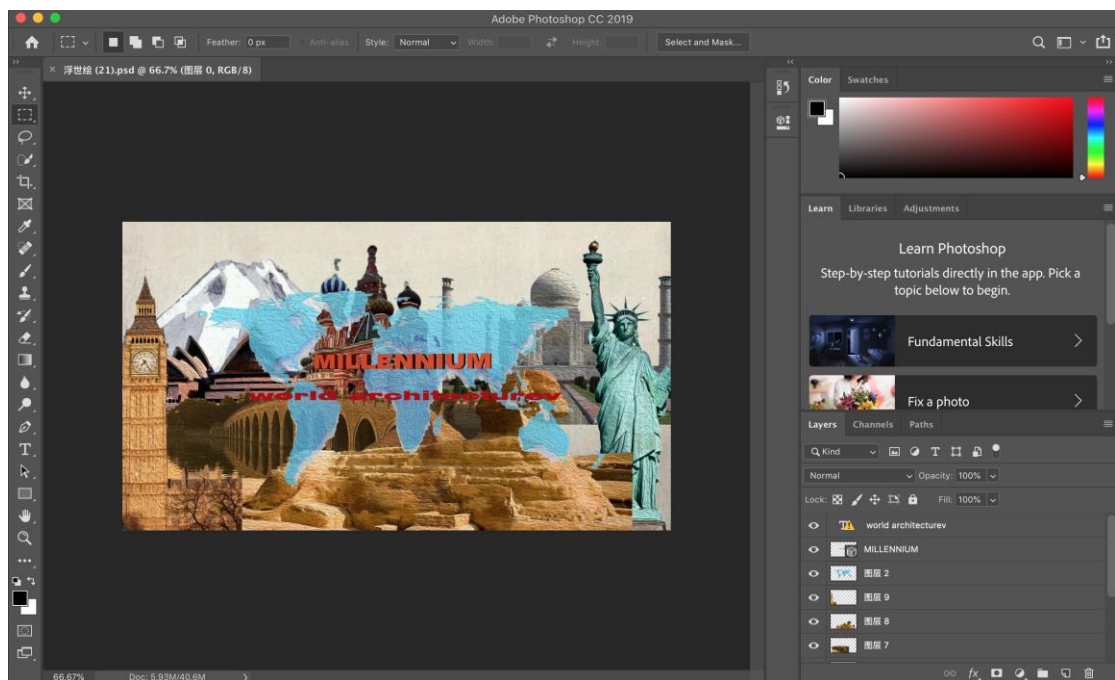


2、分别导入不同的图层进行相应的处理





3、最终合成



4、导出图片



(2) 操作 bmp 文件程序

需要有读取图片文件，存储图片，平移，旋转，镜像，亮度，对比度，放大功能。

下面展示部分重要代码，详细代码在同一文件夹下点击工程文件即可看到

1、编写底层功能函数代码

Cpp 文件

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QFileDialog>
#include <QMessageBox>
#include <QScrollBar>
```

```
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setUpUi(this);
    mainScene = new QGraphicsScene(QRect(0,0,800,600));
    ui->mainGraphic->setScene(mainScene);
```

```

        ui->sld_contrast->setMinimum(-100);
        ui->sld_contrast->setMaximum(100);
        // ui->sld_contrast->setCursor(0);

        ui->sld_light->setMinimum(-100);
        ui->sld_light->setMaximum(100);
        // ui->sld_light->setCursor(0);

        ui->sld_rotate->setMinimum(0);
        ui->sld_rotate->setMaximum(180);

        ui->sld_zoom->setMaximum(1);
        ui->sld_zoom->setMaximum(5);
    }

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_btn_open_clicked()
{
    QString filename;
    filename = QFileDialog::getOpenFileName(this, tr("选择图像"), "",
        tr("Images (*.png *.bmp *.jpg *.tif
*.GIF)")); //选择路径
    if(filename.isEmpty())
    {
        return;
    }
    else
    {
        if(!(currentImg.load(filename)))
        {
            QMessageBox::information(this, tr("error"), tr("打开图像失败!"));
            return;
        }
        currentImg.scaled(300,300,Qt::KeepAspectRatio);

        // ui->mainlabel->setPixmap(QPixmap::fromImage(currentImg)); //显示图像
        mainItem = mainScene->addPixmap(QPixmap::fromImage(currentImg));
    }
}

void MainWindow::on_btn_save_clicked()
{
    QString filename = QFileDialog::getSaveFileName(this, tr("Save Image"), "",
        tr("*.bmp;; *.png;; *.jpg;; *.tif;; *.GIF")); //选择路径
    if(filename.isEmpty())
    {
        return;
    }
}

```

```

        else
        {
            if(! ( saveImg.save(filename) ) ) //保存图像
            {
                QMessageBox::information(this,tr("Error"),tr("Failed to save the
image!"));
                return;
            }
        }
    }

void MainWindow::on_btn_MoveUp_clicked()
{
    mainItem->setPos(mainItem->pos()+ QPointF(0,-5));
}

void MainWindow::on_btn_MoveDown_clicked()
{
    mainItem->setPos(mainItem->pos()+ QPointF(0,5));
}

void MainWindow::on_btn_MoveLeft_clicked()
{
    mainItem->setPos(mainItem->pos()+ QPointF(-5,0));
}

void MainWindow::on_btn_MoveRight_clicked()
{
    mainItem->setPos(mainItem->pos()+ QPointF(+5,0));
}

void MainWindow::on_btn_Mirror_clicked()
{
    QImage temp=currentImg.mirrored(true,false);
    //currentImg = currentImg.mirrored(true,false);
    mainItem->setPixmap(QPixmap::fromImage(temp));
    currentImg=temp;
    saveImg=currentImg;
}

QImage MainWindow::changeLight(QImage &image,int position)
{
    //QImage image= currentImg;
    uchar* line=image.scanLine(0);
    uchar* pixel= line;

    for(int y=0;y<currentImg.height();++y)
    {
        pixel =line;
        for(int x=0; x<currentImg.width();++x)
        {

```

```

        *pixel = qBound(0, *pixel + position, 255);
        *(pixel+1) = qBound(0, *(pixel+1) + position, 255);
        *(pixel+2) = qBound(0, *(pixel+2) + position, 255);
        pixel += 4;
    }
    line += image.bytesPerLine();
}

return image;
}

void MainWindow::on_sld_light_sliderMoved(int position)
{
    QImage temp = currentImg;
    temp = changeLight(temp, position);
    mainItem->setPixmap(QPixmap::fromImage(temp));
    saveImg = temp;
}

QImage MainWindow::changeContrast(QImage &image, int position)
{
    uchar *line = image.scanLine(0);
    uchar *pixel = line;

    for (int y = 0; y < image.height(); ++y)
    {
        pixel = line;
        for (int x = 0; x < image.width(); ++x)
        {
            *pixel = 127 + (double)((double)*pixel - 127.0) / (1.0 - (double)position / 100));
            *(pixel + 1) = 127 + (double)((double)*(pixel + 1) - 127.0) / (1.0 - (double)position / 100));
            *(pixel + 2) = 127 + (double)((double)*(pixel + 2) - 127.0) / (1.0 - (double)position / 100));
            pixel += 4;
        }

        line += image.bytesPerLine();
    }
    return image;
}

void MainWindow::on_sld_contrast_sliderMoved(int position)
{
    QImage temp = currentImg;
    temp = changeContrast(temp, position);
    mainItem->setPixmap(QPixmap::fromImage(temp));
    saveImg = temp;
}

```



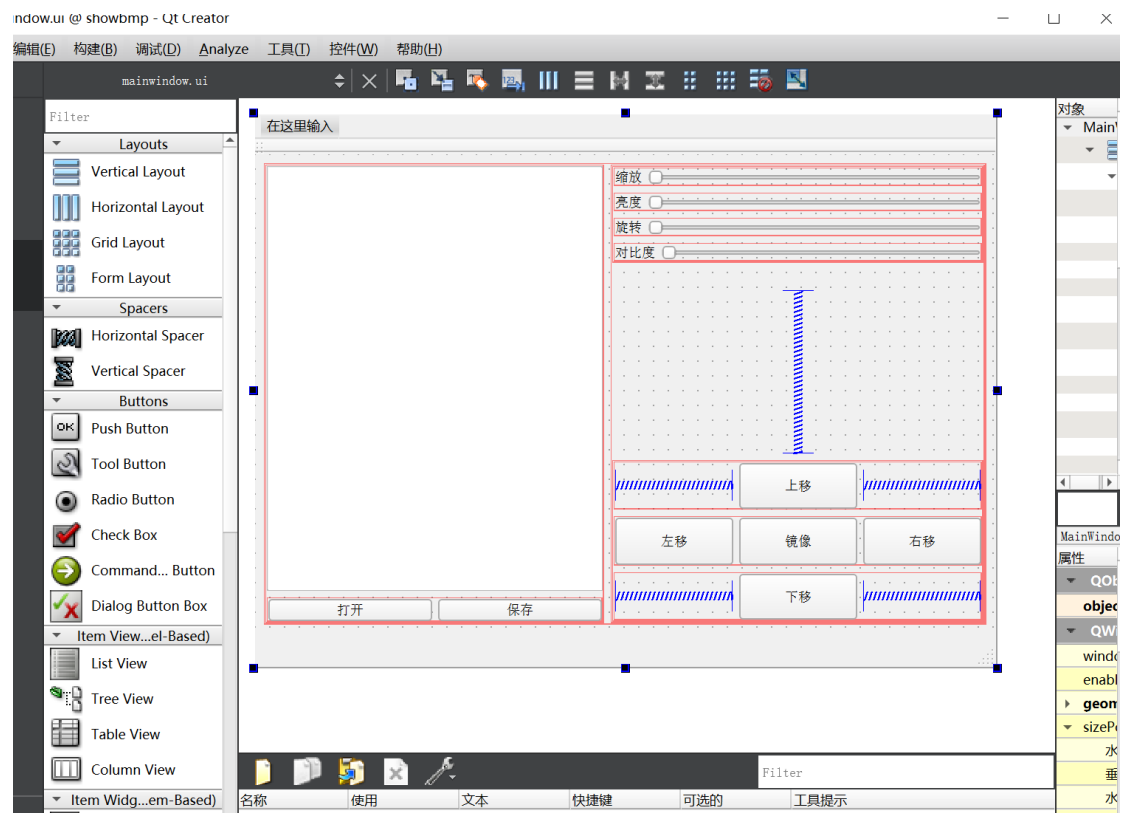
```

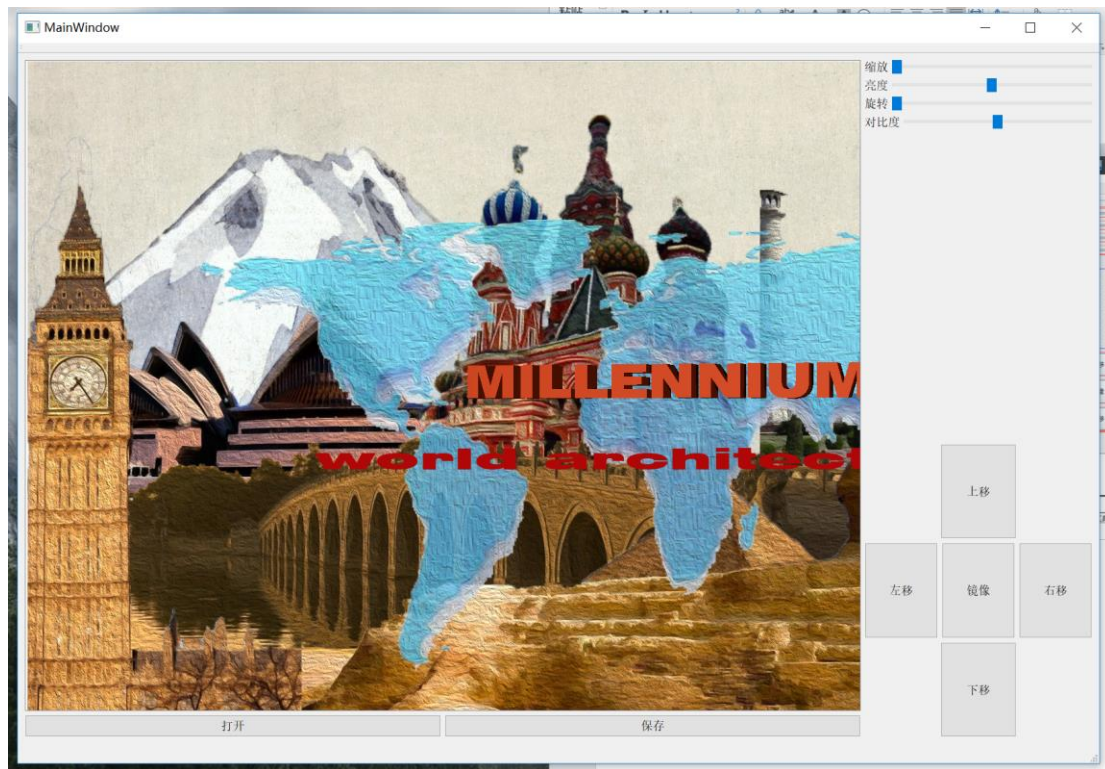
void MainWindow::on_sld_rotate_sliderMoved(int position)
{
    mainItem->setRotation(((double)position/3.14));
    mainItem->setPixmap(QPixmap::fromImage(currentImg));
    saveImg=currentImg;
}

void MainWindow::on_sld_zoom_sliderMoved(int position)
{
    currentImg.scaled(currentImg.size()*float(position),Qt::KeepAspectRatio);
    mainItem->setPixmap(QPixmap::fromImage(currentImg));
    mainItem->setScale(float(position));
    saveImg=currentImg;
}

```

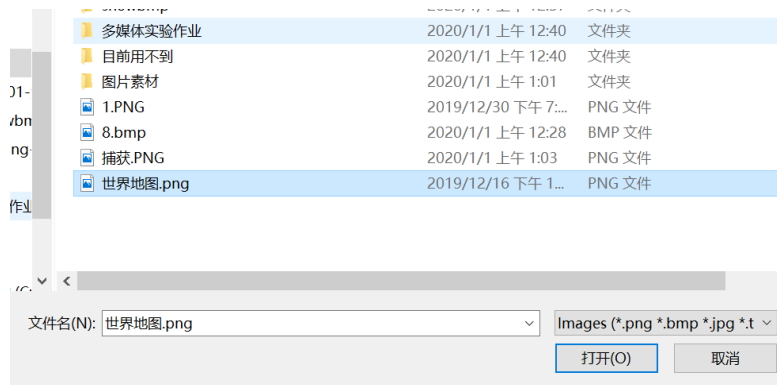
2、图形化操作界面



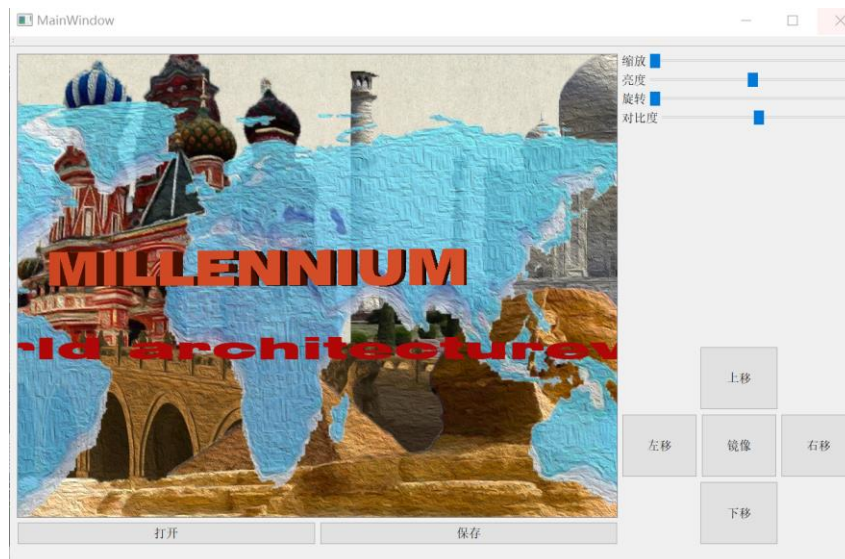
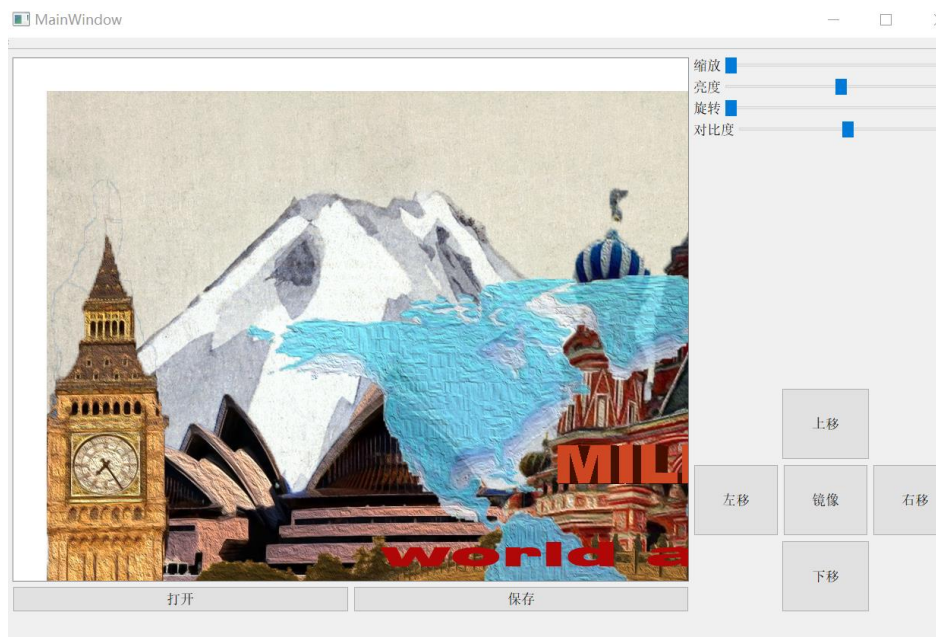


3、测试程序

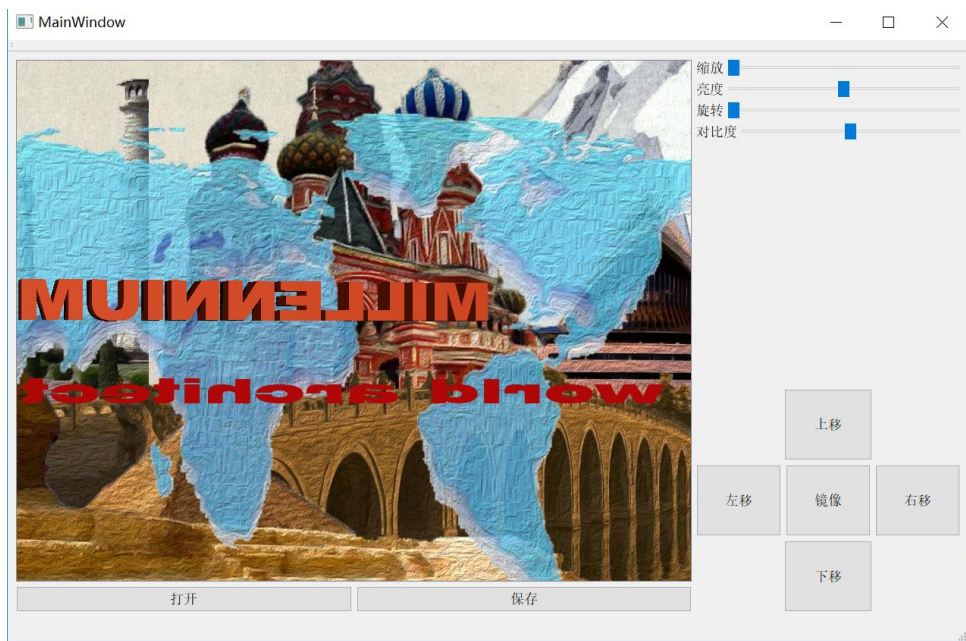
打开图片



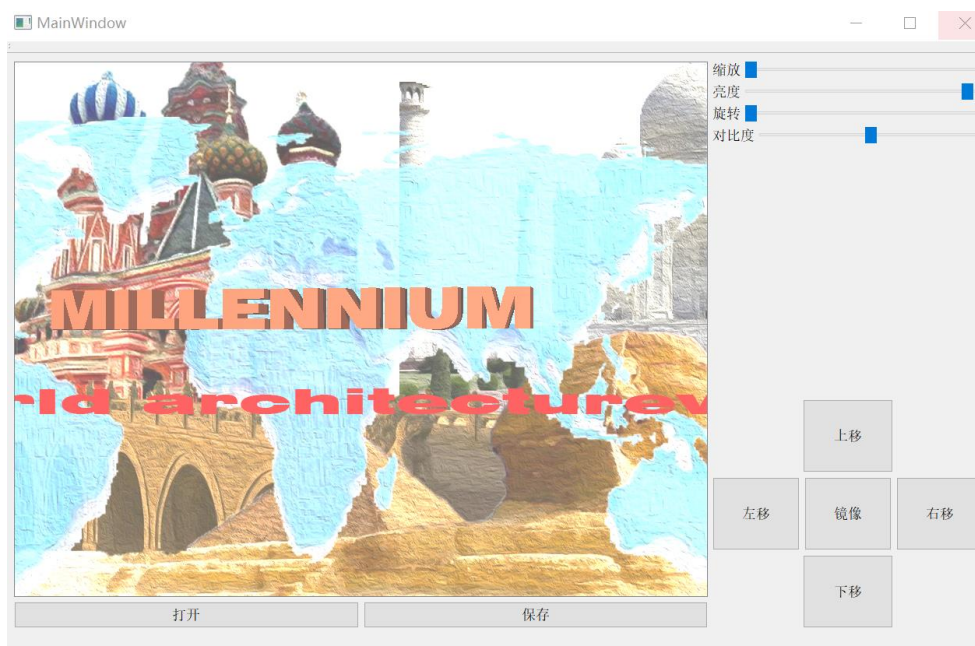
左右上下移动



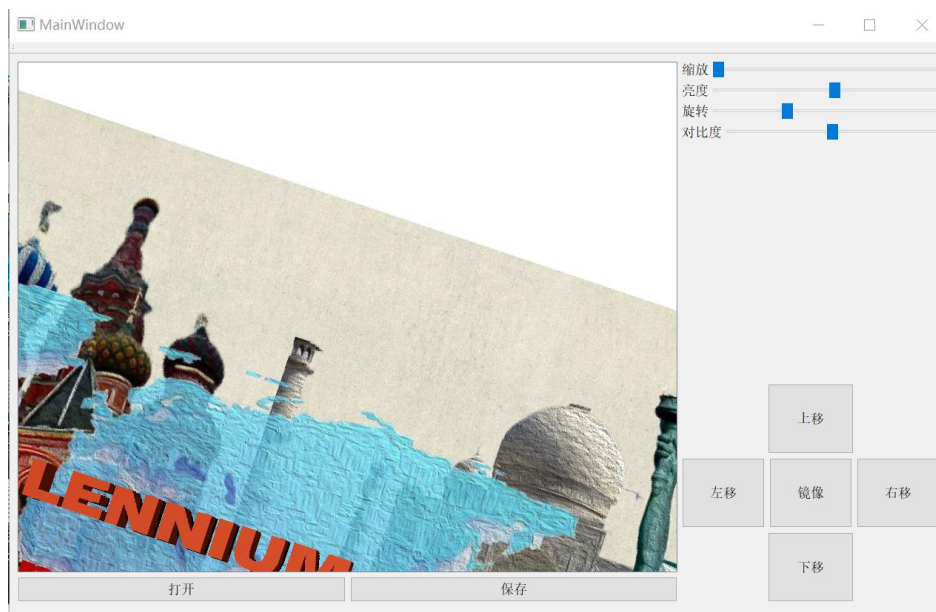
镜像



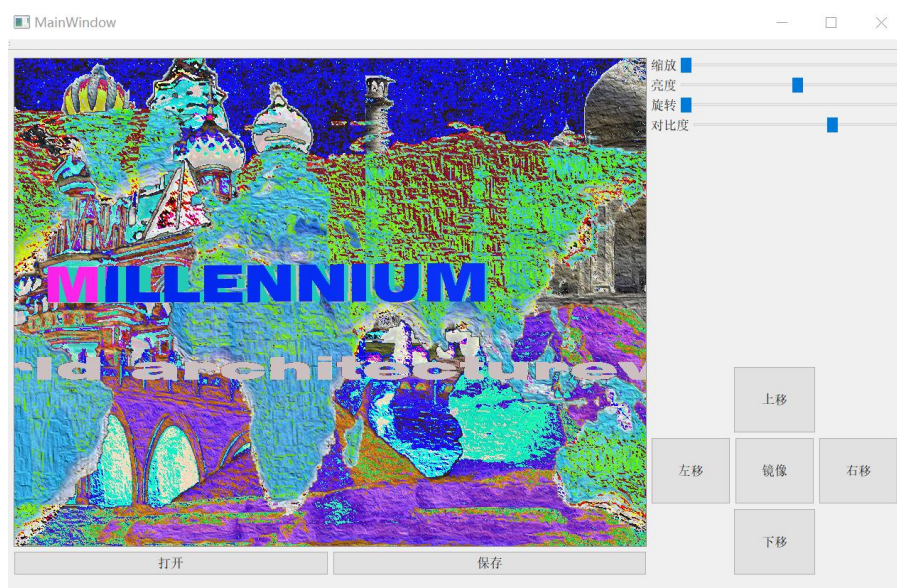
亮度



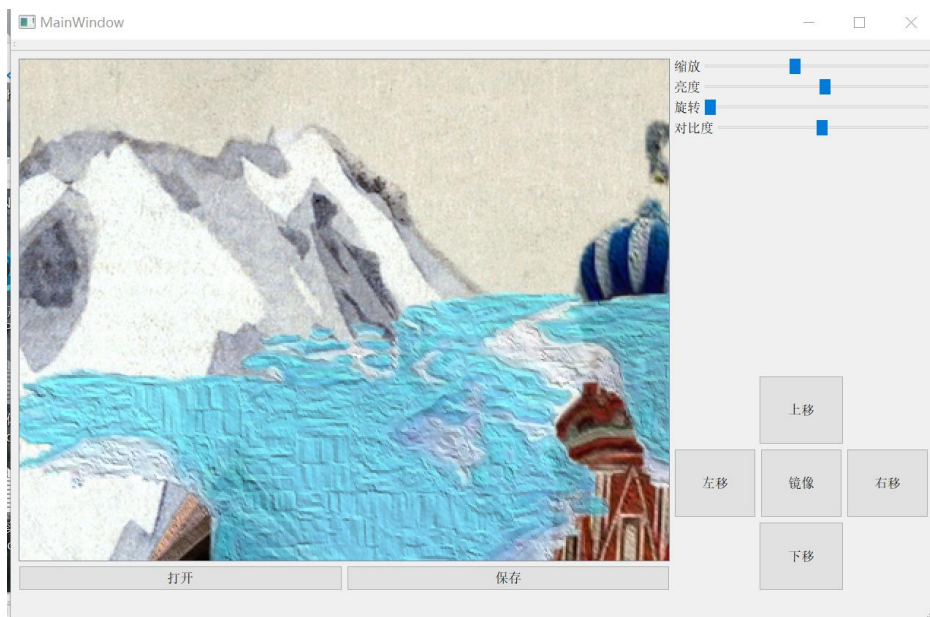
旋转



对比度



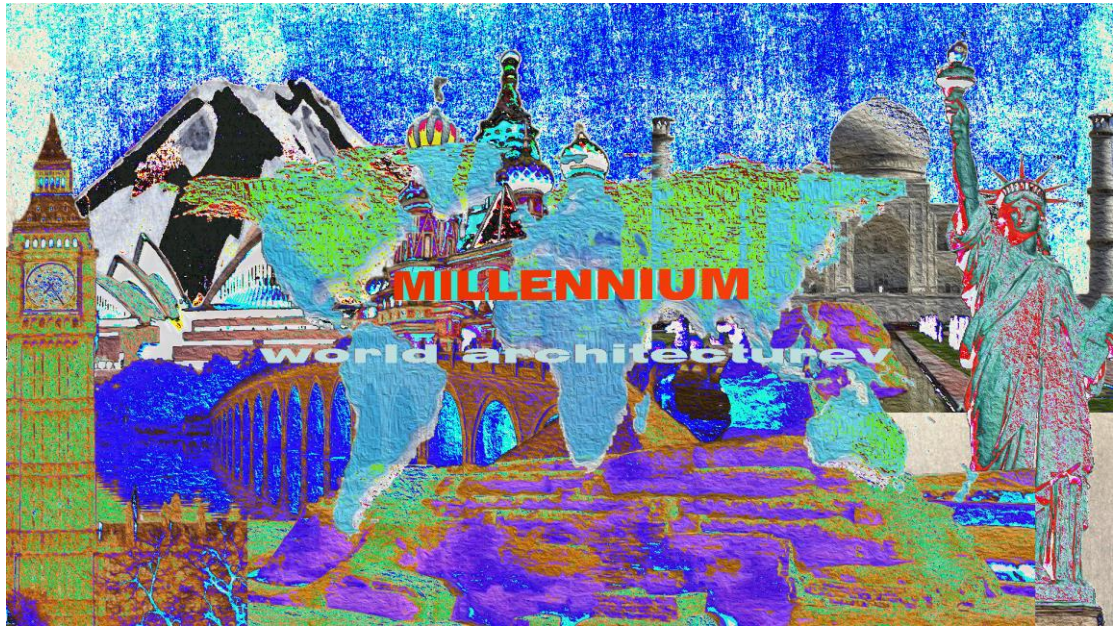
缩放



4、成果展示



之前的图和变换后的图



五、实验总结

对于第一部分的实验，使用 Photoshop 软件，处理图片，并且合成一幅世界地图，体验了一次做图的经历，熟悉了 Photoshop 软件的使用。

对于第二部分的实验，再次使用 Qt 这个软件开发软件，编写一个简单的程序，实现了读取图片，平移，放大，旋转，镜像，对比度以及亮度的改变，加深了对 bmp 格式图片的认知，同时再次熟悉了 Qt 这个开发软件。

总之，还是算有不少收获的。