

实验：透视变换 (Perspective Transformation)

实验概要

透视变换与我们迄今为止看到的所有变换不同，它需要 3×3 变换矩阵，我们暂时不讨论其背后的数学原理。我们将在输入图像中需要四个点，并在输出图像中需要相同点的坐标。请注意，这些点不应共线。接下来，类似于仿射变换步骤，我们将进行透视变换。

我们将使用以下代码创建一个 NumPy 数组来存储点：

```
ptsInput = np.float32([[in1x, in1y], [in2x, in2y], [in3x, in3y], [in4x, in4y]])
```

另外，我们可以使用以下代码：

```
ptsOutput = np.float32([[out1x, out1y], [out2x, out2y], [out3x, out3y], [out4x, out4y]])
```

接下来，我们将把这两个 NumPy 数组传递给 `cv2.getPerspectiveTransform` 函数：

```
M = cv2.getPerspectiveTransform(ptsInput, ptsOutput)
```

然后可以使用以下代码来应用转换矩阵：

```
outputImage = cv2.warpPerspective(inputImage, M, (outputImageWidth, outputImageHeight))
```

透视变换和仿射变换之间的主要区别在于，在透视变换中，直线在变换后仍将保持直线，这与仿射变换不同。在仿射变换中，由于多次变换，直线可以变换为曲线。

实验目标

在本实验中，我们将进行透视转换，以获得书的封面。我们将使用 OpenCV 的 `cv2.getPerspectiveTransform` 和 `cv2.warpPerspective` 函数。我们将使用以下图书的图像：



1. 导入依赖库

In [1]:

```
# 导入模块
import cv2                                # 导入OpenCV
import numpy as np                        # 导入NumPy
import matplotlib.pyplot as plt          # 导入matplotlib

# 魔法指令，使图像直接在Notebook中显示
# 注意，魔法指令前端%提示符表示该命令为shell执行命令，
# 执行环境并不在Notebook里面；
# 因此，相关注释不能直接跟在代码后面，需要在另行写入
%matplotlib inline
```

2. 加载图像

指定需要执行透视变换操作的目标图像路径。

您也可以上传自己的图像，需要注意的是确保加载图像路径有效，95% 以上的程序报错,除了缺少安装依赖库以外，大部分就跟数据路径不正确有关。这里使用 `cv2.imread()` 加载图像的路径，往往使用的都是相对路径，应该确保指定了正确的图片文件所在的路径。

In [2]:

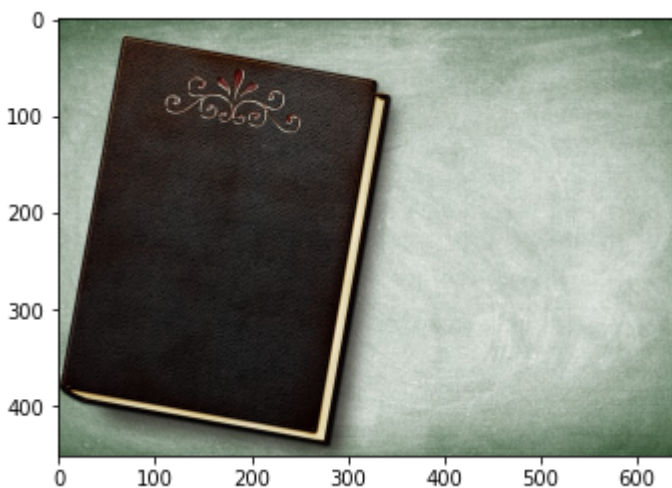
```
# 设置输入输出路径
import os
base_path = os.environ.get("BASE_PATH", '../data/')
data_path = os.path.join(base_path + "lab2/")
result_path = "result/"
os.makedirs(result_path, exist_ok=True)

img = cv2.imread("./data/book.jpg") # 读取图像文件
```

输出如下，X 和 Y 轴分别指图像的宽度和高度：

In [3]:

```
plt.imshow(img[:, :, ::-1]) # 将图像从BGR转换为RGB
plt.show()                  # 显示图像
```



3. 定位目标图像上的四个点

在图像中指定四个点，使其位于封面的四个角。

由于我们只需要封面，因此输出点将是最终 300×300 图像四个角点上。请注意，输入和输出点的点顺序应保持一致：

In [4]:

```
# 从原图像提取四个定点
inputPts = np.float32([[4, 381],
                       [266, 429],
                       [329, 68],
                       [68, 20]])

# 将提取的四个定点，映射到新的图像的四个角落点上
outputPts = np.float32([[0, 300],
                        [300, 300],
                        [300, 0],
                        [0, 0]])
```

4. 提取透视变换矩阵

In [5]:

```
# 通过透视变换生成函数，生成转换矩阵M
M = cv2.getPerspectiveTransform(inputPts, outputPts)
```

5. 应用透视变换矩阵

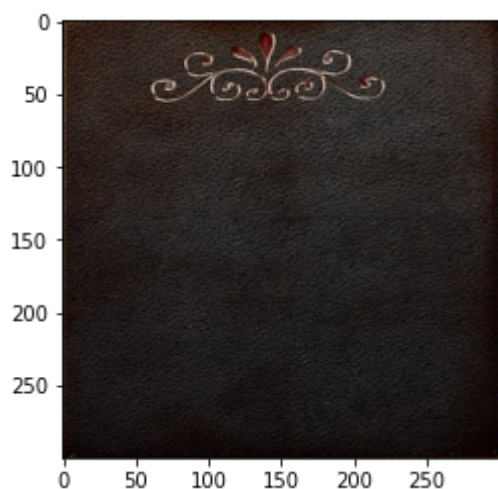
In [6]:

```
# 通过透视变换函数，将生成的转换矩阵M应用到原始图像img上，输出dst
dst = cv2.warpPerspective(img, M, (300, 300))
```

使用 Matplotlib 显示生成的图像。X 轴和 Y 轴分别为图像的宽度和高度。

In [7]:

```
# 将图像从BGR转换为RGB
plt.imshow(dst[:, :, ::-1])
# 显示图像
plt.show()
```



实验小结

在这个实验中，我们看到了如何使用几何变换中的透视变换，来提取给定图像中提供的书籍封面。这种技术应用场景在我们的生活中很常见，譬如：当我们试图扫描文档，并希望在扫描后获得文档的正确方向。