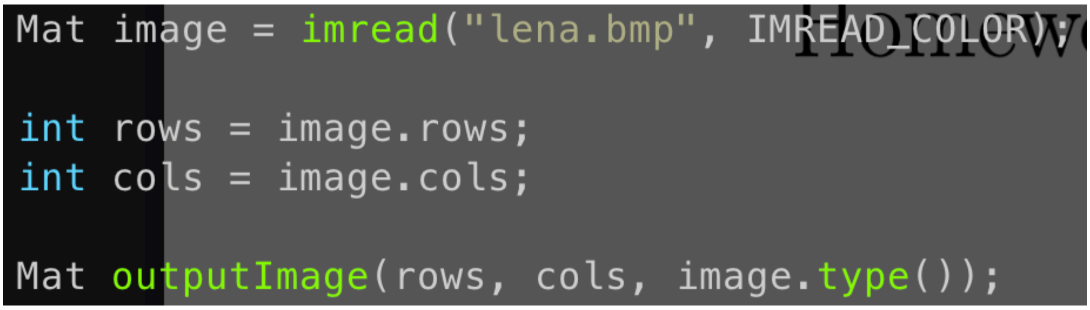
Homework Report for Computer Vision

Yu Xiang, Luo

September 12, 2023



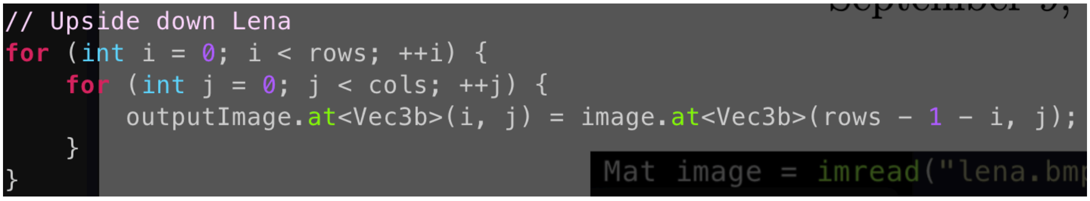
```
Mat image = imread("lena.bmp", IMREAD_COLOR);

int rows = image.rows;
int cols = image.cols;

Mat outputImage(rows, cols, image.type());
```

First, we have to read the bmp file and create the output image.

Complete Code

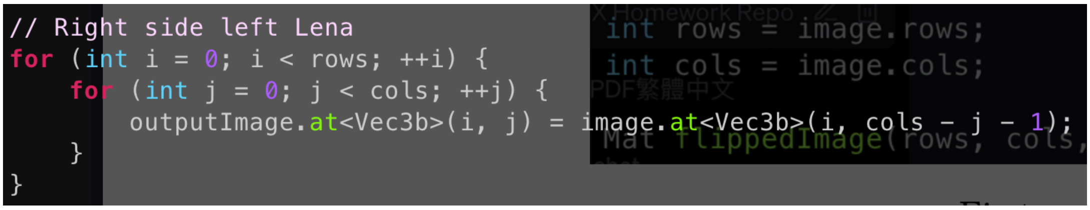
Part 1

(a) Simply reverse the X-axis index can solve this problem.



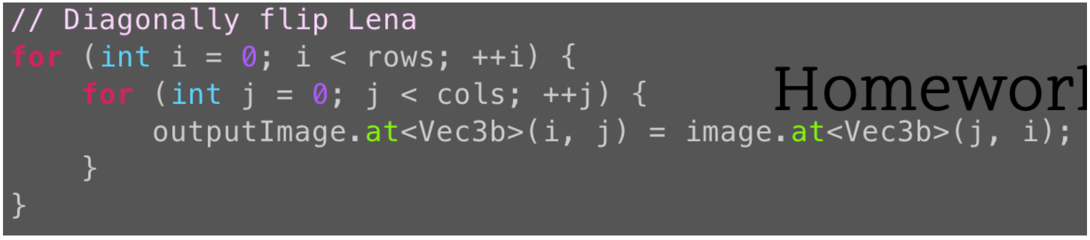
```
// Upside down Lena
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        outputImage.at<Vec3b>(i, j) = image.at<Vec3b>(rows - 1 - i, j);
    }
}
```

(b) Simply reverse the Y-axis index can solve this problem.



```
// Right side left Lena
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        outputImage.at<Vec3b>(i, j) = image.at<Vec3b>(i, cols - j - 1);
    }
}
```

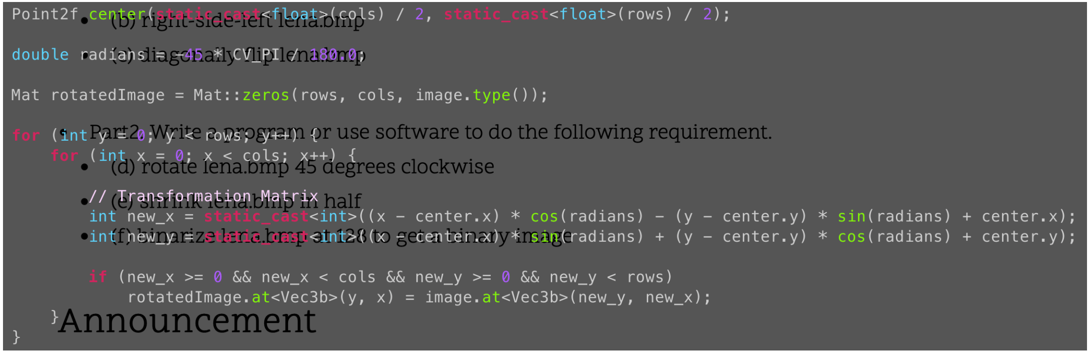
(c) Make all (x, y) to (y, x).



```
// Diagonally flip Lena
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        outputImage.at<Vec3b>(i, j) = image.at<Vec3b>(j, i);
    }
}
```

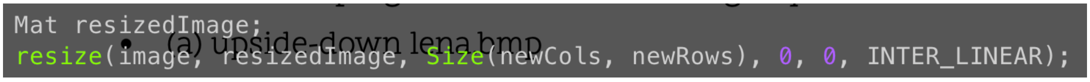
Part 2

(d) In this case, we need to find the center, calculate the radians of the degree of 45, and operate the image using the transformation matrix.



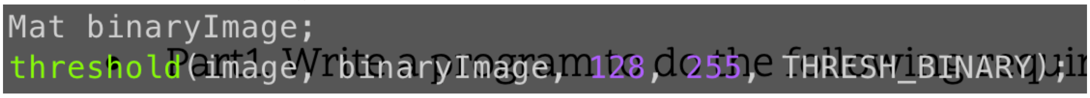
```
Point2f center(static_cast<float>(cols) / 2, static_cast<float>(rows) / 2);

double radians = -45 * CV_PI / 180.0;

Mat rotatedImage = Mat::zeros(rows, cols, image.type());

for (int y = 0; y < rows; y++) {
    for (int x = 0; x < cols; x++) {

        // Transformation Matrix
        int new_x = static_cast<int>((x - center.x) * cos(radians) - (y - center.y) * sin(radians) + center.x);
        int new_y = static_cast<int>((x - center.x) * sin(radians) + (y - center.y) * cos(radians) + center.y);

        if (new_x >= 0 && new_x < cols && new_y >= 0 && new_y < rows)
            rotatedImage.at<Vec3b>(y, x) = image.at<Vec3b>(new_y, new_x);
    }
}
```

(e) Just call the function in opencv can solve this easily.



```
Mat resizedImage;
resize(image, resizedImage, Size(newCols, newRows), 0, 0, INTER_LINEAR);
```

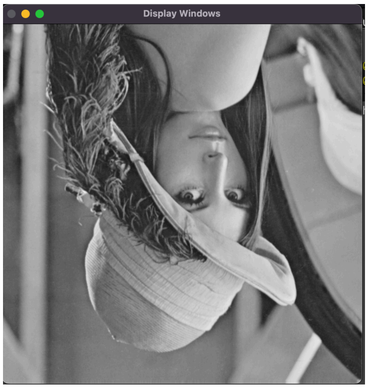
(f) Call the function in opencv.



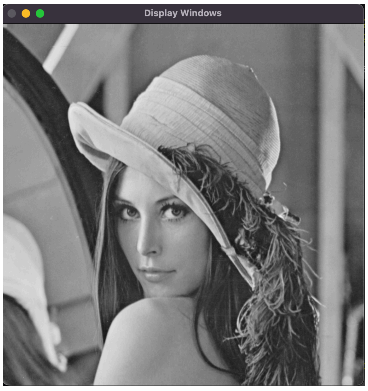
```
Mat binaryImage;
threshold(image, binaryImage, 128, 255, THRESH_BINARY);
```

All the output Lena

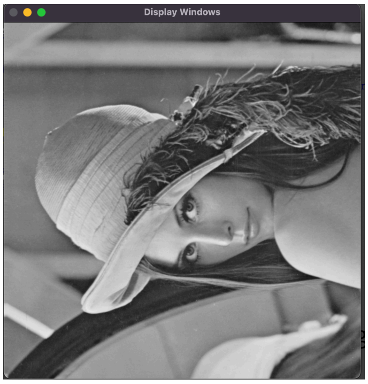
(a) Upside down Lena



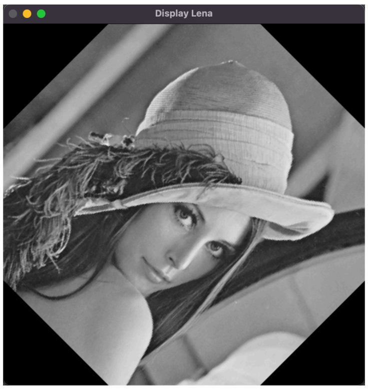
(b) Right side left Lena



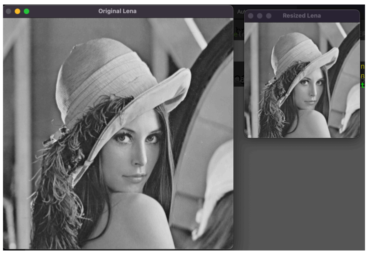
(c) Diagonally flip Lena



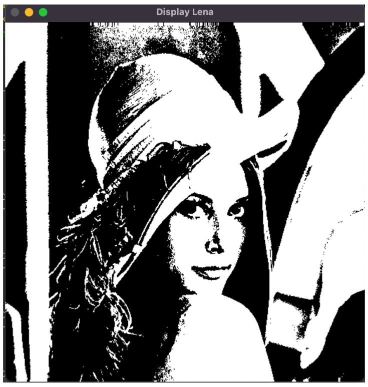
(d) Rotate Lena



(e) Shrink Lena



(f) Binarized Lena


Display Lena