Hao-en Sung
University of California, San Diego

October 2, 2016

1 Assignment 1

1.1 Image Manipulation [5pts]

On the course website, you will find 4 sets of images (border1.png & center1.png, border2.png & center2.png, border3.jpg & center3.jpg, and border4.jpg & center4.jpg). The border images have had their center removed, and centers represent the respective removed portions. Write a matlab function that loads two images (border and center) and reconstructs the original image from them, then displays them. Do not use any for or while loops, or any hard coded sizes or borders. You should use the same function to reconstruct all images. Add all the output images and all of your code to the report described in the next section (4 images total).The images are here.

Sol. Calculate the position of top-right corner in cropped border picture, and simply add the center picture into it. However, I encountered some difficulties in dealing with odd picture sizes, since there is no accurate definition of "center". I finally tried it out by enumerating the odd-and-even combinations between the length of border images and center images. Codes and results are shown below.

```
function [] = manipulate(bname, cname, oname)

bfile = imread(bname);
cfile = imread(cname);

[bh, bw, ~] = size(bfile);
[ch, cw, ~] = size(cfile);

ph = idivide(int32(bh-ch+1), int32(2), 'floor');
pw = idivide(int32(bw-cw+1), int32(2), 'floor');

bfile(ph:ph+ch-1, pw:pw+cw-1, :) = cfile;
imwrite(bfile, oname);

end
```

□

1.2 Image Rotation [5pts]

Consider an Image I. Write a function to rotate I by θ radians anticlockwise, $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Do not use IMROTATE.M or any other library function. Tile the output similar to figure 1 *(omitted)*. You should run your function on the starbucks-ring.jpg image provided on the website and provide results only for that.

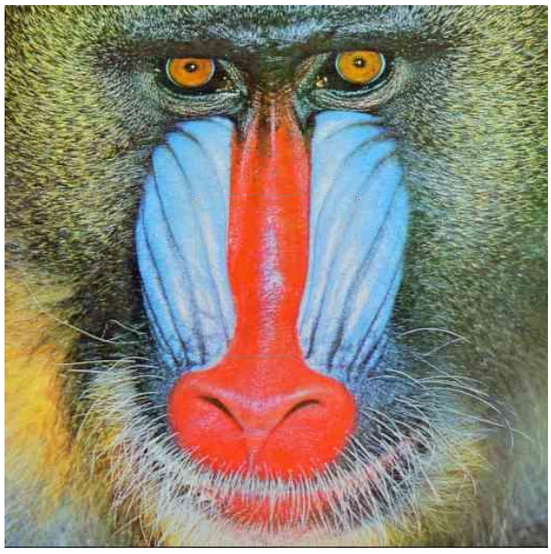
Sol. The solving framework is simple: translation, rotation, and translation. Nonetheless, it took me some time to make sure the translation part, since coordinate axis is different from matrix axis. I finally figured it out by drawing some simple examples.

```
function [ret] = rotate(fname, theta)

ffile = imread(fname);
[h, w, d] = size(ffile);

ret = zeros(w, h, d, 'uint8');

hh = (1+h) / 2;
hw = (1+w) / 2;
rotMat = [cos(theta), sin(theta); -sin(theta), cos(theta)];

for i = 1:h
    for j = 1:w
        resMax = [i-hh, j-hw] * rotMat + [hh, hw];
        nh = int32(resMax(1,1));
        nw = int32(resMax(1,2));
        ret(nh,nw,:) = ffile(i,j,:);
    end
end

end
```

□

(a) output1.png

(b) output2.png



(c) output3.jpg

(d) output4.jpg

Figure 1: Reconstructed Figures

Figure 2: Rotated Starbucks-ring

1.3 Appendix

A small script to run the above two functions is added here for reference.

```
%% Path setting
DAT = '../dat/';
RES = '../res/';


%% Problem 1: Image Manipulation [5 pts]

bnameLst = cellstr(['border1.png';'border2.png';'border3.jpg';'border4.jpg']);
cnameLst = cellstr(['center1.png';'center2.png';'center3.jpg';'center4.jpg']);

for i = 1:length(bnameLst)
    bname = strcat(DAT, bnameLst{i});
    cname = strcat(DAT, cnameLst{i});
    oname = strcat(RES, 'output', bname(end-4:end));
    manipulate(bname, cname, oname);
end



%% Problem 2:  Image Rotation [5 pts]

% parameters
angleLst = [pi/2, pi, pi*3/2];
fname = strcat(DAT, 'starbucks-ring.jpg');
oname = strcat(RES, 'rotation.jpg');

res = figure('visible','off');
ffile = imread(fname);
subplot(3,3,[1:6]);
imshow(ffile);

for i = 1:length(angleLst)
    img = rotate(fname, angleLst(i));
    subplot(3,3,6+i);
    imshow(img);
end

saveas(res, oname);
```

Submitted by Hao-en Sung on October 2, 2016.