Panorama Stitching in MATLAB

Mahnoor Anjum
Registration Number 132937
Bachelors in Electrical Engineering (ongoing)
CS-475 Computer Vision, BEE7
Instructor: Dr. Wajahat Hussain

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         Author: Mahnoor Anjum
%         Instructor: Dr. Wajahat Hussain
%         Date Created: 10/21/2018
%         Date Submitted: 10/22/2018
%         Comments: -/
%         Acknowledgements:
%             This assignment is my original work.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% 1. First try making a panorama using two images only. Than you can repeat
the
% process for more image.
% 2. You can load the image in matlab using the imread command. Use matlab
help
% to figure out how to use the commands mentioned. For example in this case
just
% type help imread and press enter.
im1 = imread('view1.png');
im2 = imread('view2.png');
% 3. Find out the size of the image by using the size command.
dim1 = size(im1);
dim2 = size(im2);
% 4. Now you want to find the corresponding points between two images. For
this you
% need to display images in separate windows. Use the figure command to
create
% a window. Next use imshow command to display the image.
% figure, imshow(im1);
% figure, imshow(im2);
% 5. Next select four correspondences between two images. First select four
points
% on figure 1. Use ginput command to get these four points.
figure, imshow(im1);
[x1,y1] = ginput(4);
figure, imshow(im2);
[x2,y2] = ginput(4);
% 6. Remember in matlab the coordinate origin is top left corner. X is the
column
% number, Y is the row number.
% 7. You want to solve this equation A h = 0 . Matrix A is 8x9 and h is 9x1.
Convert
% the four point correspondence into A using the equation you studied in the
class.
```

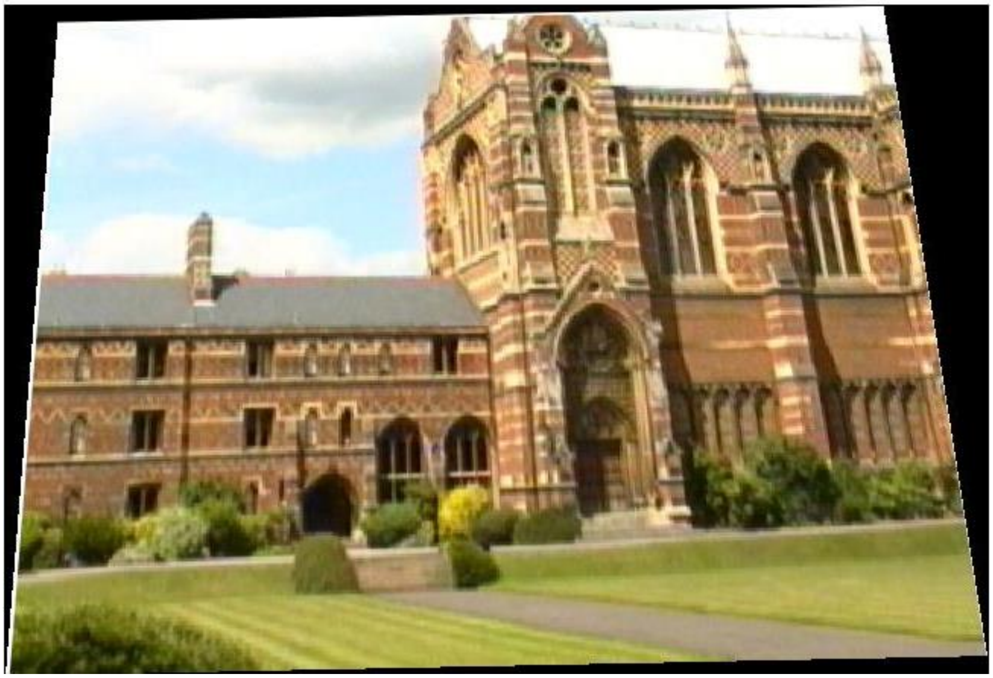
```
npoints = 4;
A = zeros(2*npoints, 9);

xi = [x1(1),y1(1) 1;
       x1(2),y1(2) 1;
       x1(3),y1(3) 1;
       x1(4),y1(4) 1];
xi_ =   [x2(1),y2(1) 1;
         x2(2),y2(2) 1;
         x2(3),y2(3) 1;
         x2(4),y2(4) 1];
npoints = 4;
A = zeros(2*npoints, 9);


k = 1
for i = 1:2:8
    xi_s = xi_(k,:);
    x = xi_s(1);
    y = xi_s(2);
    w = xi_s(3);
    A(i,4:6)= -w*xi(k,:);
    A(i,7:9)= y*xi(k,:);
    A(i+1,1:3)= w*xi(k,:);
    A(i+1,7:9)= -x*xi(k,:);
    k = k+1;
end;


if npoints==4
    H = null(A);
else
    [U,S,V] = svd(A);
    H = V(:,9);
end;

H_shaped = reshape(H,3,3);

x122= H_shaped*(xi');
tform = projective2d(H_shaped);
transformed = imwarp(im1, tform);
%figure,imshow(transformed);

% 9. Now comes the interesting step. You want to merge the two images. You
want to
% figure out automatically which pixels are novel in image 2 not present in
image 1.
% You can do this by applying the inverse homography to image 2 pixel
coordinate.
% Those pixels that fall outside the boundary of image 1 are new pixels.
H_inv = pinv(H_shaped);
x_221= H_inv*(xi');
tform = projective2d(H_inv);
transformed = imwarp(im1, tform);
%figure,imshow(transformed);
```

```
% 10.In order to merge the two images, start by making a big empty matrix.
What will
% be the size of this empty matrix. You can figure this by applying inverse
% homography to four corners of image 2.
H_shaped = H_shaped';
H_inv = pinv(H_shaped);


im2_new= zeros(size(im2));
for j=1:size(im2,1)
    for i= 1:size(im2,2)
    im2t= [i;j;1];
    im2_mp= H_inv*im2t;
    im2_mp = im2_mp./im2_mp(3);
    im2_new(j,i,:)= round(im2_mp);
    end
end
%image 1 corners (referenced through coordinates)
im1_c1=[1;1;1]; %top-left
im1_c2=[size(im1,1);1;1]; %top-right
im1_c3=[1;size(im1,2);1]; %bottom-left
im1_c4=[size(im1,1);size(im1,2);1]; %bottom-right
%image2 corners (referenced through spatial position/index of coordinate
matrix)
im2_cords=im2_new(1,1,:); %top-left
im2_cords= im2_cords(:)
im2_cords2=im2_new(1,size(im2,2),:); %top-right
im2_cords2= im2_cords2(:)
im2_cords3=im2_new(size(im2,1),1,:); %bottom-left
im2_cords3= im2_cords3(:)
im2_cords4=im2_new(size(im2,1),size(im2,2),:); %bottom-right
im2_cords4= im2_cords4(:)
%Finding min and max of x and y axis
xmax = max([im2_cords(1) im2_cords2(1) im2_cords3(1) im2_cords4(1)
size(im1,2)])
xmin = min([im2_cords(1) im2_cords2(1) im2_cords3(1) im2_cords4(1) 1])
ymax = max([im2_cords(2) im2_cords2(2) im2_cords3(2) im2_cords4(2)
size(im1,1)])
ymin = min([im2_cords(2) im2_cords2(2) im2_cords3(2) im2_cords4(2) 1])
%Initialization of M
M= 255*ones(ymax-ymin+1,xmax-xmin+1,3);
%size(M)
if xmin<0
tx = abs(xmin)+1;
end
if ymin<0
ty = abs(ymin)+1;
end
%Transferring Image 1 pixels
for r=1:size(im1,1)
for c= 1:size(im1,2)
pixel_value= im1(r,c,:);
pixel_value= pixel_value(:);
M(r+ty, c+tx, :)= pixel_value;
end
```

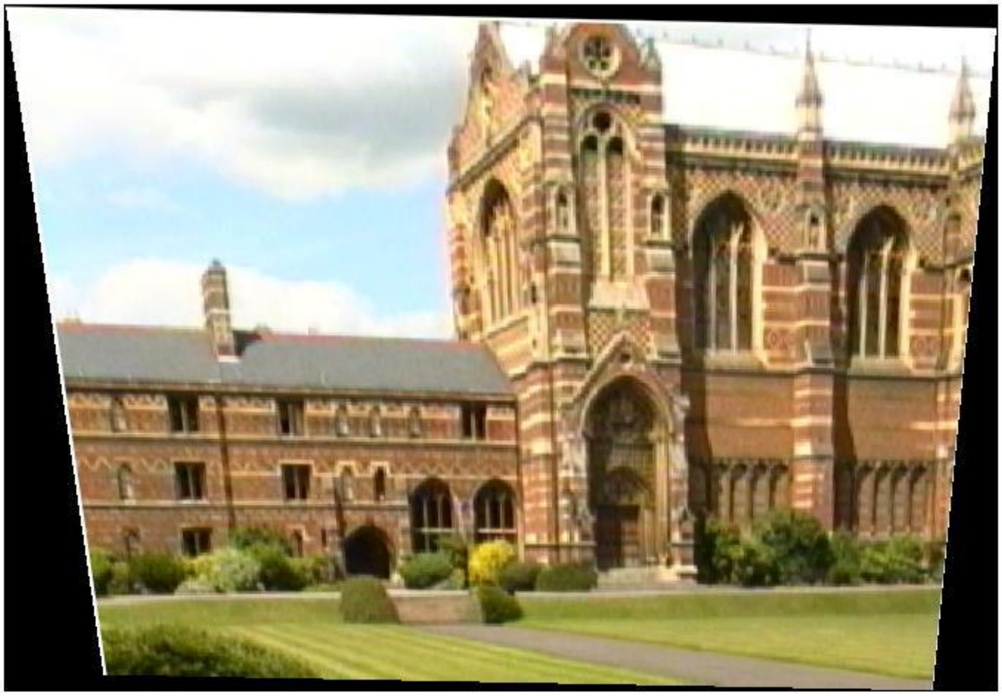
```
end
%Transferring Image 2 pixels
for r=1:size(im2,1)
for c= 1:size(im2,2)
pixel_value= im2(r,c,:);
pixel_value= pixel_value(:);
index= im2_new(r,c,:);
if (index(1)<=1 || index(1)>=size(im1, 2)) || (index(2)<=1 ||
index(2)>=size(im1, 1))
M(index(2)+ty,index(1)+tx, :)= pixel_value;
end
end
end

M_fin=uint8(M);
figure;
imshow(M_fin)

% 11.Transfer all the pixels from image 2, not present in image 1, to the big
empty
% matrix that you made in the last step.
% 12.Finally use interp2 command to interpolate missing values.
% 13.Remember to use the homogeneous version of all pixel coordinates.
% 14.Oral viva will be conducted to judge the originality of assignment.
```

OUTPUT

Mapping one to two



Mapping two to one



Panorama Stitching