Task3

`**Name:** Hongyi liu` `**Degree:** ME` `**ID:** 20788389`

Problem 1: Homogeneous Coordinate (Lines and Points) (10 points)

The intersection of two lines l1 and l2, with l1 passing through the points (0,4) and (3,0), and l2 passing through the points (8,-2) and (10,8).

(a) Compute the intersection without using homogeneous coordinates

(a) $l1: y = a1x + b1$; $l2: y = a2x + b2$;

$x_{11} = (0,4)$; $x_{12} = (3,0)$; $x_{21} = (8,-2)$; $x_{22} = (10,8)$;

$a1 = -\frac{4}{3}$; $b1 = 4$; $a2 = 5$; $b2 = -42$;

$$\begin{cases} l1: y = -\frac{4}{3}x + 4 \\ l2: y = 5x - 42 \end{cases} \qquad \begin{cases} x = \frac{138}{19} \\ y = -\frac{108}{19} \end{cases}$$

$P\left(\frac{138}{19}, -\frac{108}{19}\right)$

(b) Compute the intersection using homogeneous coordinates

(b) $l1 = (0,4,1) \times (3,0,1) = (4,3,-12)$

$l2 = (8,-2,1) \times (10,8,1) = (-5,1,42)$

$p = l1 \times l2 = (4,3,-12) \times (-5,1,42) = (138,-108,19) = \left(\frac{138}{19}, -\frac{108}{19}, 1\right)$

$p = \left(\frac{138}{19}, -\frac{108}{19}, 1\right)$

Problem 2: Homogeneous Coordinate (Conic) (10 points)

Given a conic

$$x^2 + y^2 + 4x + 2y - 29 = 0$$

(a) Compute the intersection of a tangential line to this conic at (1,4) with the y-axis

(a) $x^2+y^2+4x+2y-29=0$; $a=1$; $b=0$; $c=1$; $d=4$; $e=2$; $f=-29$;

$$C = \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 1 & -29 \end{bmatrix}$$

$x_1 = (1,4,1)$

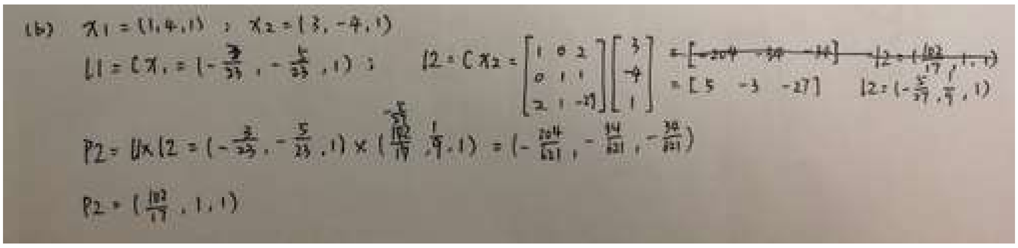
$$l_1 = Cx_1 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 1 & -29 \end{bmatrix}\begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} = [3 \ \ 5 \ \ -23] \qquad l1 = \left(-\frac{3}{23}, -\frac{5}{23}, 1\right)$$

$y\text{-axis} = (1,0,0)$

$P1 = l1 \times y\text{-axis} = \left(-\frac{3}{23}, -\frac{5}{23}, 1\right) \times (1,0,0) = \left(0, 1, \frac{5}{23}\right)$

$P1 = \left(0, \frac{23}{5}, 1\right)$

(b) Compute the coorindates of the intersection of the tangents to this conic at points (1,4) and (3, -4)


(b) $x_1 = (1,4,1)$; $x_2 = (3,-4,1)$

$l_1 = Cx_1 = (-\frac{3}{23}, -\frac{5}{23}, 1)$; $l_2 = Cx_2 = \begin{bmatrix}1&0&2\\0&1&1\\2&1&-27\end{bmatrix}\begin{bmatrix}3\\-4\\1\end{bmatrix} = [5 \ \ -3 \ \ -27]$ $l_2 = (-\frac{5}{27}, \frac{1}{9}, 1)$

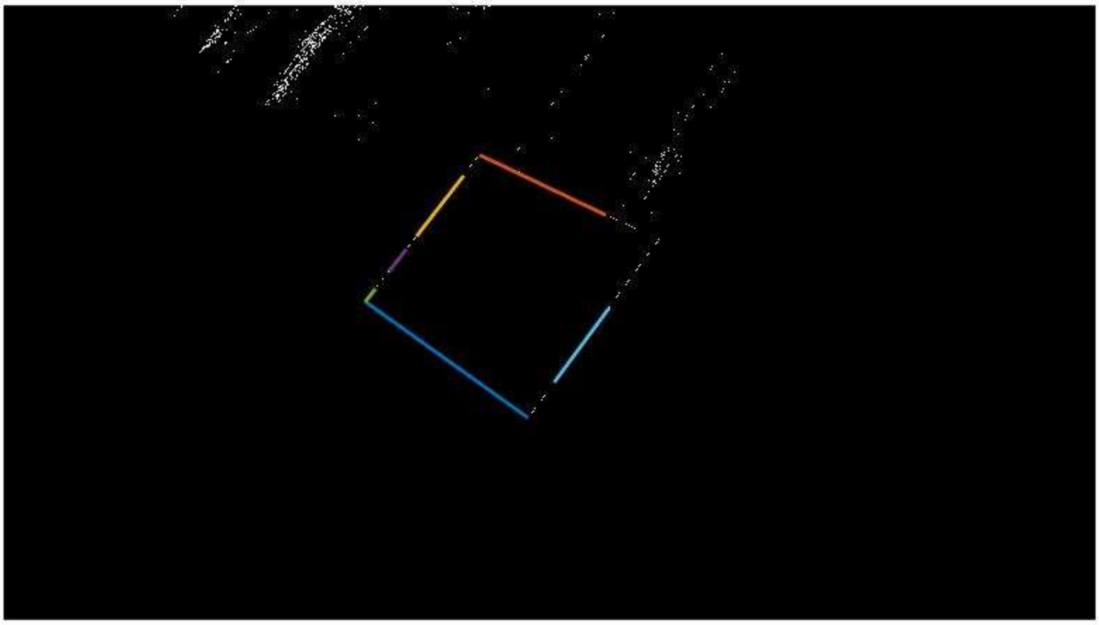
$P_2 = l_1 \times l_2 = (-\frac{3}{23}, -\frac{5}{23}, 1) \times (-\frac{5}{27}, \frac{1}{9}, 1) = (-\frac{204}{621}, -\frac{34}{621}, -\frac{30}{621})$

$P_2 = (\frac{102}{17}, 1, 1)$

Problem 3: Vanishing Line and Point (10 points)

(a) Compute two vanishing point coordinates that are formed from vertical edges and horizontal edges of the paper, respectively.

```
%% Problem 3

clear;clc;close all;
I=imread('20190206_155248.jpg');
I=rgb2gray(I);
level = graythresh(I)
BW = imbinarize(I,level);
F5=bwperim(BW,8);
%imshow(F5);

[H, theta, rho]= hough(F5,'RhoResolution', 0.5,'ThetaResolution',0.5);
peak=houghpeaks(H,4); %peak
lines=houghlines(F5,theta,rho,peak); %lines
subplot(111),imshow(F5,[]),hold on
for k=1:length(lines)
    xy=[lines(k).point1;lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2);
end
%%
a1=[lines(1).point1,1];b1=[lines(1).point2,1];
a2=[lines(2).point1,1];b2=[lines(2).point2,1];
a3=[lines(3).point1,1];b3=[lines(3).point2,1];
a4=[lines(6).point1,1];b4=[lines(6).point2,1];
l1=cross(a1,b1);l1=l1(1,:)/l1(1,3);
l2=cross(a2,b2);l2=l2(1,:)/l2(1,3);
l3=cross(a3,b3);l3=l3(1,:)/l3(1,3);
l4=cross(a4,b4);l4=l4(1,:)/l4(1,3);
v1=cross(l1,l2);v1=v1(1,:)/v1(1,3);
v2=cross(l3,l4);v2=v2(1,:)/v2(1,3);
v1 %vanishing point 1
v2 %vanishing point 2
```



```
v1 =

   1.0e+03 *

  -2.0803   -1.3218    0.0010


v2 =

   1.0e+04 *

   2.2122   -2.5150    0.0001
```

(b) Compute a vanishing line that connect the above two vanishing points.

You need to clearly explain how you get the answer with a code or hand calculation.

```
v=cross(v1,v2);v=v(1,:)/v(1,3) %vanishing line
```

```
v =

    0.0003    0.0003    1.0000
```

(c) Please explain how to take a picture of the paper in such a way that the vanishing line becomes a line at infinity.

Do not take the picture right over the paper. So the vertical edges and horizontal edges won't be parallel in the picture and they will intersect at the vanishing points. The vanishing points compose the vanishing line.

Problem 4: Affine Rectification (10 points)

I take a photo of a rectangular picture frame and the pixel coordinates of the corners of the frame are at P1 = (3,1), P2 = (7, 4), P3 = (8, 2), P4 = (7 -1). Write a code to transform this frame (rectangle) to have only affine distortion (the rectangle determined by new four points has only affine distortion). Please justify why a transformed rectangle has only affine distortion.

The transformed rectangle has 2 pairs of parallel lines which means it has only affine distortion.

The affine transformed rectangle is obtained by multiplying projective transformed rectangle by H.

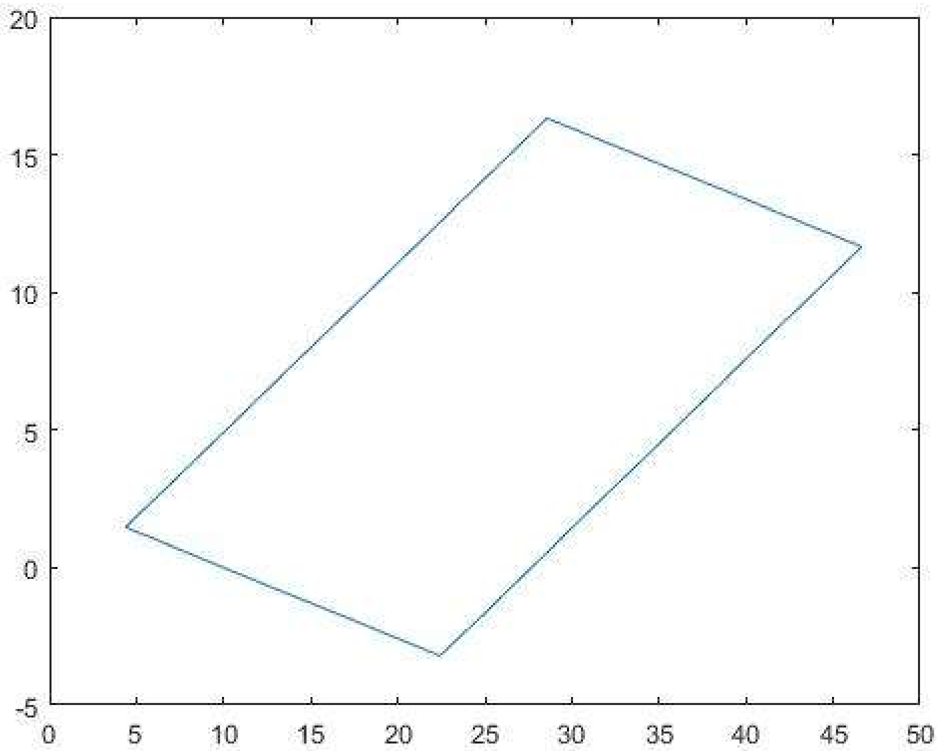
H=[1,0,0;

0,1,0;

l_1,l_2,l_3]

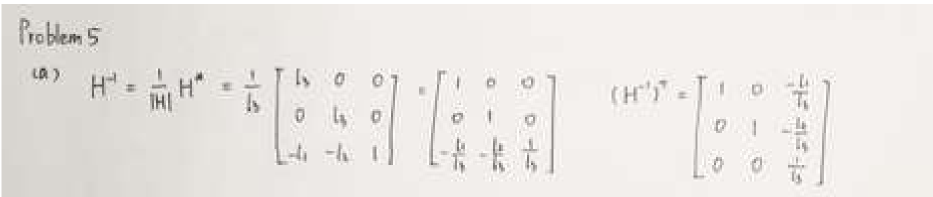
vanishing line l=(l_1,l_2,l_3])

```
%% Problem 4

clear;clc;close all;

x1=[3,1,1];
x2=[7,4,1];
x3=[8,2,1];
x4=[7,-1,1];
l1=cross(x1,x2);
l2=cross(x3,x4);
v1=cross(l1,l2);
l3=cross(x2,x3);
l4=cross(x4,x1);
v2=cross(l3,l4);
v=cross(v1,v2);
v=v/v(3);
H = [1 0 0;0 1 0;v(1) v(2) v(3)];
x1=H*x1';
x2=H*x2';
x3=H*x3';
x4=H*x4';
x=[x1(1)/x1(3),x2(1)/x2(3),x3(1)/x3(3),x4(1)/x4(3)];x=[x,x(1)];
y=[x1(2)/x1(3),x2(2)/x2(3),x3(2)/x3(3),x4(2)/x4(3)];y=[y,y(1)];
plot(x,y);
```



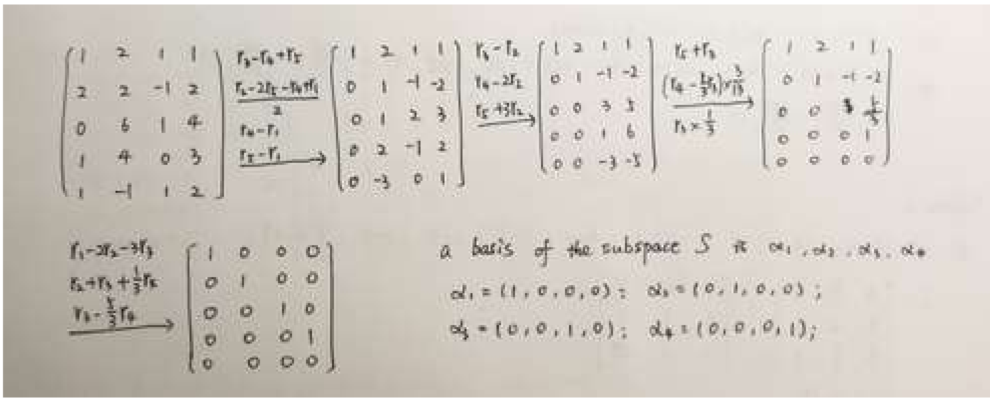
Problem 5: Linear Algebra (20 points)

(a) Find a transpose of inverse H



(b) Find a basis of the subspace S. What is the definition of a basis?

A basis B is a linearly independent subset of a vector space V over a field F that spans V. For every finite subset$\{b_1,\ldots,b_n\}$ of B and every $a_1,\ldots,a_n$ in F, if $a_1b_1+\ldots+a_nb_n=0$, then necessarily $a_1=\ldots=a_n=0$. For every vector v in V, it is possible to choose $v_1,\ldots,v_n$ in F and $b_1,\ldots,b_n$ in B such that $v=v_1b_1+\ldots,v_nb_n$.

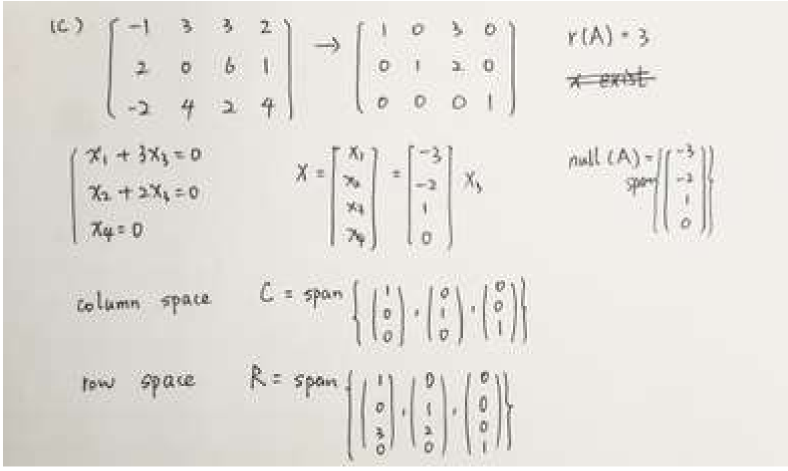


(c) What is the definition of a row, column, and null space? Please find a column, row, and null space of a matrix A, respectively.

The row space of A is the subspace of F^n spanned by the row vectors of A.

The column space of a matrix A is the subspace of F^n spanned by the column vectors of A.

The null space of a matrix A is the set of all vectors x which satisfy Ax=0.



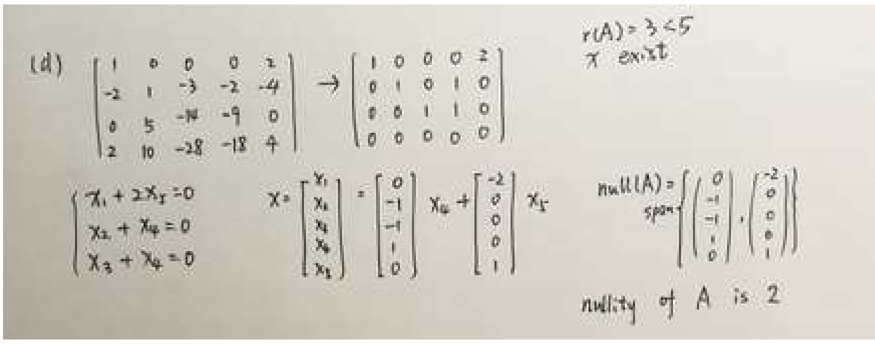
(c) $\begin{bmatrix} -1 & 3 & 3 & 2 \\ 2 & 0 & 6 & 1 \\ -2 & 4 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $r(A)=3$ ~~x exist~~

$\begin{cases} x_1 + 3x_3 = 0 \\ x_2 + 2x_3 = 0 \\ x_4 = 0 \end{cases}$ $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 1 \\ 0 \end{bmatrix} x_3$ $null(A) = span\left\{\begin{pmatrix} -3 \\ -2 \\ 1 \\ 0 \end{pmatrix}\right\}$

column space $C = span\left\{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\right\}$

row space $R = span\left\{\begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right\}$

(d) Find a non-singular (non-trivial) vector of x that satisfy Ax = 0. Does x exist? What is the null space of A and the nullity of A? Please explain your answer.

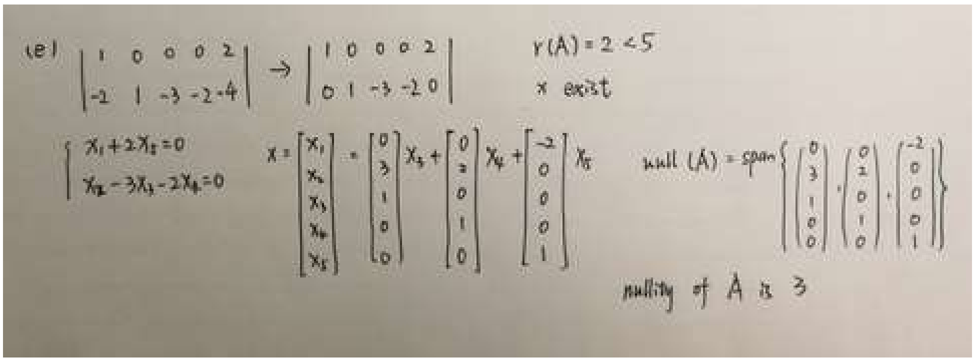


(d) $\begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ -2 & 1 & -3 & -2 & -4 \\ 0 & 5 & -14 & -9 & 0 \\ 2 & 10 & -28 & -18 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $r(A)=3<5$ x exist

$\begin{cases} x_1 + 2x_5 = 0 \\ x_2 + x_4 = 0 \\ x_3 + x_4 = 0 \end{cases}$ $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 0 \end{bmatrix} x_4 + \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} x_5$ $null(A) = span\left\{\begin{pmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right\}$

nullity of A is 2

(e) Find a non-singular (non-trivial) vector of x that satisfy Ax = 0. Does x exist? What is the null space of A and the nullity of A? Please explain your answer.



(e) $\begin{vmatrix} 1 & 0 & 0 & 0 & 2 \\ -2 & 1 & -3 & -2 & -4 \end{vmatrix} \rightarrow \begin{vmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & -3 & -2 & 0 \end{vmatrix}$ $r(A)=2<5$ x exist

$\begin{cases} x_1 + 2x_5 = 0 \\ x_2 - 3x_3 - 2x_4 = 0 \end{cases}$ $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} x_3 + \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} x_4 + \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} x_5$ $null(A) = span\left\{\begin{pmatrix} 0 \\ 3 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right\}$

nullity of A is 3

Problem 6: Image Overlay (20 points)

```
%% Problem 6(a)

clear;close all;clc;

imgBoardFile = 'IMG_0067.JPG';
```

```
imgPicFile = 'bnw43.jpg';

info = imfinfo(imgPicFile);
sizePic = [info.Width info.Height];

imgPic=imread(imgPicFile);
imgPic=rgb2gray(imgPic);
[h1 w1]=size(imgPic);

imgBoard=imread(imgBoardFile);
[h2 w2]=size(imgBoard);

figure;imshow(imgBoard);

p1=[1,1;w1,1;1,h1;w1,h1];
p2=ginput(4); %Please click on the 4 corners in the following order:Upper left, upper
right, lower left, lower right

corner(1:2,:)=p2(1:2,:);
corner(3,:)=p2(4,:);
corner(4,:)=p2(3,:);

H = calc_homography6(p1,p2);

imgPic = imread(imgPicFile);
[imgPicTran, RB] = imwarp(imgPic, projective2d(H));
BWPic = roipoly(imgPicTran, corner(:,1)-RB.XWorldLimits(1), corner(:,2)-
RB.YWorldLimits(1));

BWBoard = ~roipoly(imgBoard, corner(:,1), corner(:,2));
RA = imref2d(size(BWBoard));

imgBoardMask = bsxfun(@times, imgBoard, cast(BWBoard, 'like', imgBoard));
imgPicTranMask = bsxfun(@times, imgPicTran, cast(BWPic, 'like', imgPicTran));

imgFinal(:,:,1) = imfuse(imgBoardMask(:,:,1),RA, imgPicTranMask(:,:,1),RB,'diff');
imgFinal(:,:,2) = imfuse(imgBoardMask(:,:,2),RA, imgPicTranMask(:,:,2),RB,'diff');
imgFinal(:,:,3) = imfuse(imgBoardMask(:,:,3),RA, imgPicTranMask(:,:,3),RB,'diff');

imshow(imgFinal); imwrite(imgFinal, 'resulta.jpg');

%%

function T = calc_homography6(points1, points2)

xaxb = points2(:,1) .* points1(:,1);
xayb = points2(:,1) .* points1(:,2);
yaxb = points2(:,2) .* points1(:,1);
yayb = points2(:,2) .* points1(:,2);

A = zeros(size(points1, 1)*2, 9);
A(1:2:end,3) = 1; A(2:2:end,6) = 1;
A(1:2:end,1:2) = points1;
```

```
A(2:2:end,4:5) = points1;
A(1:2:end,7) = -xaxb;
A(1:2:end,8) = -xayb;
A(2:2:end,7) = -yaxb;
A(2:2:end,8) = -yayb;
A(1:2:end,9) = -points2(:,1);
A(2:2:end,9) = -points2(:,2);
[junk1,junk2,V] = svd(A);
h = V(:,end) ./ V(end,end);
T= reshape(h,3,3);
end
```

```
%% Problem 6(b)

clear;close all;clc;

imgBoardFile = 'IMG_0069.JPG';
imgPicFile = 'bnw43.jpg';

info = imfinfo(imgPicFile);
sizePic = [info.Width info.Height];

imgPic=imread(imgPicFile);
imgPic=rgb2gray(imgPic);
[h1 w1]=size(imgPic);

imgBoard=imread(imgBoardFile);
```

```
[h2 w2]=size(imgBoard);

figure;imshow(imgBoard);

p1=[1,1;w1,1;1,h1;w1,h1];
p2=ginput(4); %Please click on the 4 corners in the following order:Upper left, upper
right, lower left, lower right

corner(1:2,:)=p2(1:2,:);
corner(3,:)=p2(4,:);
corner(4,:)=p2(3,:);

H = calc_homography6(p1,p2);

imgPic = imread(imgPicFile);
[imgPicTran, RB] = imwarp(imgPic, projective2d(H));
BWPic = roipoly(imgPicTran, corner(:,1)-RB.XWorldLimits(1), corner(:,2)-
RB.YWorldLimits(1));

BWBoard = ~roipoly(imgBoard, corner(:,1), corner(:,2));
RA = imref2d(size(BWBoard));

imgBoardMask = bsxfun(@times, imgBoard, cast(BWBoard, 'like', imgBoard));
imgPicTranMask = bsxfun(@times, imgPicTran, cast(BWPic, 'like', imgPicTran));

imgFinal(:,:,1) = imfuse(imgBoardMask(:,:,1),RA, imgPicTranMask(:,:,1),RB,'diff');
imgFinal(:,:,2) = imfuse(imgBoardMask(:,:,2),RA, imgPicTranMask(:,:,2),RB,'diff');
imgFinal(:,:,3) = imfuse(imgBoardMask(:,:,3),RA, imgPicTranMask(:,:,3),RB,'diff');

imshow(imgFinal); imwrite(imgFinal, 'resultb.jpg');

%%

function T = calc_homography6(points1, points2)

xaxb = points2(:,1) .* points1(:,1);
xayb = points2(:,1) .* points1(:,2);
yaxb = points2(:,2) .* points1(:,1);
yayb = points2(:,2) .* points1(:,2);

A = zeros(size(points1, 1)*2, 9);
A(1:2:end,3) = 1; A(2:2:end,6) = 1;
A(1:2:end,1:2) = points1;
A(2:2:end,4:5) = points1;
A(1:2:end,7) = -xaxb;
A(1:2:end,8) = -xayb;
A(2:2:end,7) = -yaxb;
A(2:2:end,8) = -yayb;
A(1:2:end,9) = -points2(:,1);
A(2:2:end,9) = -points2(:,2);
[junk1,junk2,V] = svd(A);
h = V(:,end) ./ V(end,end);
T= reshape(h,3,3);
```

```
end
```

Problem 7: Build your 3D Planar Measurement Software (30 points)

The width and length of the table measured physically is 59 cm and 106 cm. The result calculated by the codes is close to the true value.

```
dw =

   59.5971


dh =

  108.5695

fx >>
```

```
%% Problem 7
clear;close all;clc;

imgBoardFile = 't3p7.jpg';

h1=27.8;
w1=21.6;

imgBoard=imread(imgBoardFile);

figure;imshow(imgBoard);

p1=[1,1;w1,1;w1,h1;1,h1];
p2=ginput(4); %Please click on the 4 corners in the following order:Upper left, upper right, lower right, lower left

H = calc_homography7(p1,p2);

figure;imshow(imgBoard);
p3=ginput(2);  % select the short side of the table
q3=ginput(2);  % select the long side of the table

p2_new(:,1:2)=p2;
p2_new(:,3)=1;
p2_new=p2_new';
p3_new(:,1:2)=p3; q3_new(:,1:2)=q3;
p3_new(:,3)=1;    q3_new(:,3)=1;
p3_new=p3_new';   q3_new=q3_new';

h=inv(H);
p4=h*p2_new;      q4=h*p2_new;
p5=h*p3_new;      q5=h*q3_new;

p4x(1)=p4(1,1)/p4(3,1);
p4x(2)=p4(1,2)/p4(3,2);
p4x(3)=p4(1,3)/p4(3,3);
p4x(4)=p4(1,4)/p4(3,4);
p4y(1)=p4(2,1)/p4(3,1);
p4y(2)=p4(2,2)/p4(3,2);
p4y(3)=p4(2,3)/p4(3,3);
p4y(4)=p4(2,4)/p4(3,4);
```

```
p5x(1)=p5(1,1)/p5(3,1);  q5x(1)=q5(1,1)/q5(3,1);
p5x(2)=p5(1,2)/p5(3,2);  q5x(2)=q5(1,2)/q5(3,2);
p5y(1)=p5(2,1)/p5(3,1);  q5y(1)=q5(2,1)/q5(3,1);
p5y(2)=p5(2,2)/p5(3,2);  q5y(2)=q5(2,2)/q5(3,2);

d1=sqrt((p4x(1)-p4x(2))^2+(p4y(1)-p4y(2))^2);
d2=sqrt((p5x(1)-p5x(2))^2+(p5y(1)-p5y(2))^2);
dw=d2/d1*21.6;
dw

d3=sqrt((q5x(1)-q5x(2))^2+(q5y(1)-q5y(2))^2);
dh=d3/d1*21.6;
dh

%%
function T = calc_homography7(points1, points2)

xaxb = points2(:,1) .* points1(:,1);
xayb = points2(:,1) .* points1(:,2);
yaxb = points2(:,2) .* points1(:,1);
yayb = points2(:,2) .* points1(:,2);

A = zeros(size(points1, 1)*2, 9);
A(1:2:end,3) = 1; A(2:2:end,6) = 1;
A(1:2:end,1:2) = points1;
A(2:2:end,4:5) = points1;
A(1:2:end,7) = -xaxb;
A(1:2:end,8) = -xayb;
A(2:2:end,7) = -yaxb;
A(2:2:end,8) = -yayb;
A(1:2:end,9) = -points2(:,1);
A(2:2:end,9) = -points2(:,2);
[junk1,junk2,V] = svd(A);
h = V(:,end) ./ V(end,end);
T= reshape(h,3,3)';
end
```