

Завдання на курсову роботу з дисципліни "Комп'ютерна графіка"

Номер варіанту	Розмір зображення		Двовимірна геометрична фігура – прямокутник			Зображення літер
	ширина N , дискр. точок	висота M , дискр. точок	ширина H , дискр. точок	висота L , дискр. точок	кут повороту α , градусів відносно Ox проти год. стрілки	
1	250	250	70	80	0	АБВ
2	300	300	80	100	10	БВГ
3	350	400	90	120	20	ВГД
4	400	500	100	80	30	ГДЕ
5	450	250	120	100	40	ДЕЄ
6	500	300	70	120	45	ЕЄІ
7	250	400	80	80	60	Є І І
8	300	500	90	100	0	І І Й
9	350	250	100	120	10	Й Й К
10	400	300	120	80	20	ЙКЛ
11	450	400	70	100	30	КЛМ
12	500	500	80	120	40	ЛМН
13	250	250	90	80	45	МНО
14	300	300	100	100	60	НОП
15	350	400	120	120	0	ОПР
16	400	500	70	80	10	ПРС
17	450	250	80	100	20	РСТ
18	500	300	90	120	30	СТУ
19	250	400	100	80	40	ФХШ
20	300	500	120	100	45	ФШЧ
21	350	250	70	120	60	ЧШТ
22	400	300	80	80	0	ЩФУ
23	450	400	90	100	10	ЮЯЪ
24	500	500	100	120	20	ЛМТ
25	250	250	120	80	30	МНФ
26	300	300	70	100	40	РСФ
27	350	400	80	120	45	АПР
28	400	500	90	80	60	ВРТ
29	450	350	100	100	45	РСЯ
30	500	500	120	120	50	БЮФ

Номер варіанту	Параметри геометричного перетворення								
	Зсув, дискр. точок			Масштабування			Кут повороту, градусів проти год. стрілки відносно вказаної осі		
	T_x	T_y	T_z	S_x	S_y	S_z	$Teta_x$	$Teta_y$	$Teta_z$
1	0	10	0	0,5	0,5	0,8	10	20	10
2	20	20	10	0,6	0,7	1,2	20	30	20
3	30	30	20	0,7	0,9	1,5	30	45	30
4	40	10	0	0,8	1,0	0,8	45	0	40
5	0	20	10	0,9	1,2	1,2	40	15	45
6	20	30	20	1,1	1,3	1,3	35	5	50
7	30	10	0	1,2	1,5	1,5	25	40	0
8	40	20	10	1,3	0,5	0,8	0	30	75
9	0	30	20	1,4	0,7	1,2	10	20	10
10	20	10	0	1,5	0,9	1,5	20	30	20
11	30	20	10	0,5	1,0	0,8	30	45	30
12	40	30	20	0,6	1,2	1,2	45	0	40
13	0	10	0	0,7	1,3	1,3	40	15	45
14	20	20	10	0,8	1,5	1,5	35	5	50
15	30	30	20	0,9	0,5	0,8	25	40	0
16	40	10	0	1,1	0,7	1,2	0	30	75
17	0	20	10	1,2	0,9	1,5	10	20	10
18	20	30	20	1,3	1,0	0,8	20	30	20
19	30	10	0	1,4	1,2	1,2	30	45	30
20	40	20	10	1,5	1,3	1,3	45	0	40
21	0	30	20	0,5	1,5	1,5	40	15	45
22	20	10	0	0,6	0,5	0,8	35	5	50
23	30	20	10	0,7	0,7	1,2	25	40	0
24	40	30	20	0,8	0,9	1,5	0	30	75
25	0	10	0	0,9	1,0	0,8	10	20	10
26	20	20	10	1,1	1,2	1,2	20	30	20
27	30	30	20	1,2	1,3	1,3	30	45	30
28	40	10	0	1,3	1,5	1,5	45	0	40
29	25	20	10	1,4	1,0	0,8	40	15	45
30	35	30	20	1,5	1,2	1,2	35	5	50

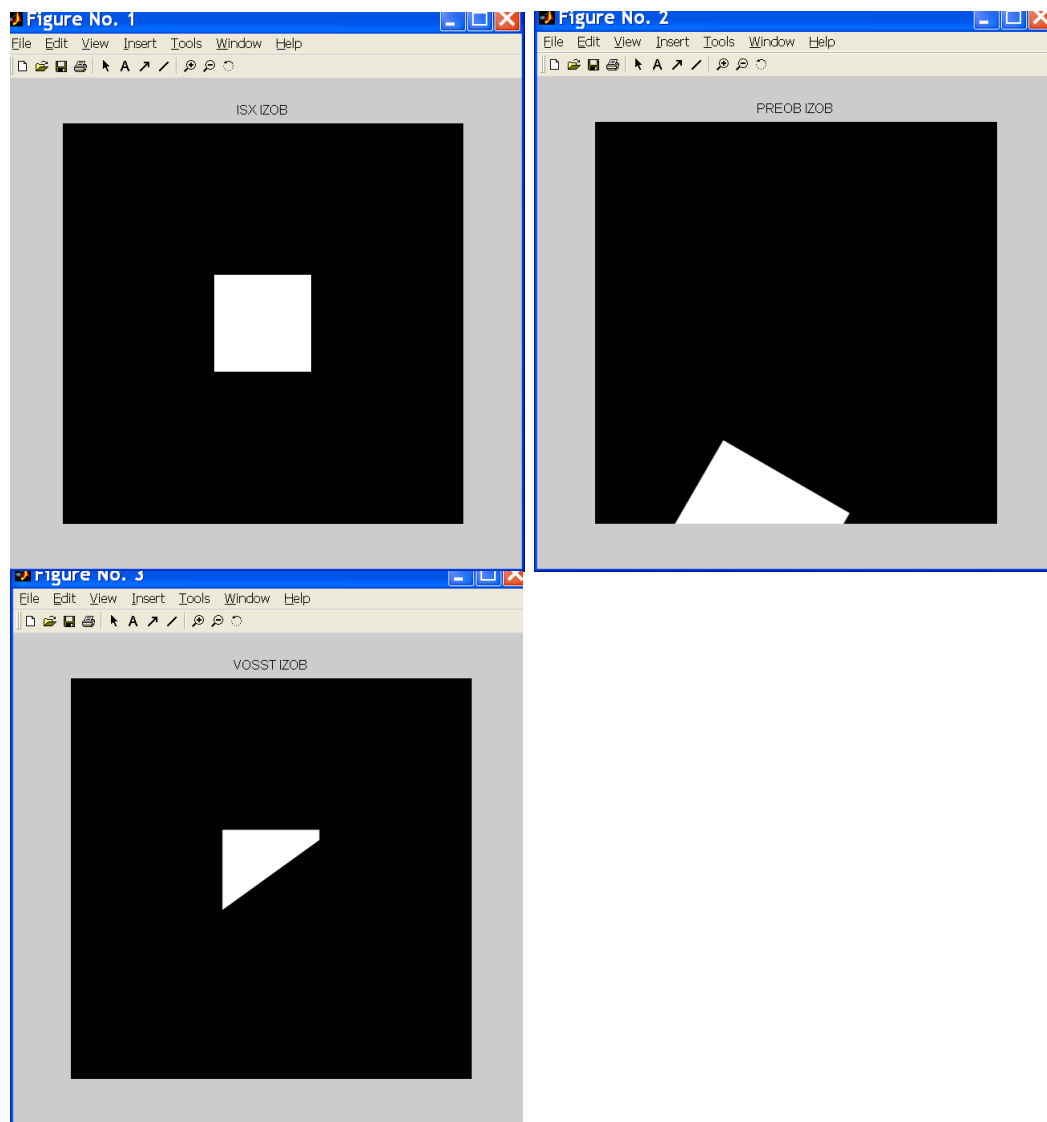
Розділи пояснювальної записки:

- титульний лист
- зміст
- завдання на курсову роботу
- вступ
- 1. Огляд методів геометричних перетворень растрової графіки
- 2. Розробка методу геометричних перетворень зображень у двовимірному просторі
- 3. Розробка методу геометричних перетворень зображень тривимірних об'єктів
- висновки
- список літератури

Послідовність виконання курсової роботи:

1. Для двовимірного зображення геометричної фігури згідно варіанту завдання дослідити задане геометричне перетворення.
 - 1.1. Сформувані двовимірне зображення заданої геометричної фігури
 - 1.2. Виділити 3 контрольні точки геометричної фігури та визначити їх координати:
 - а) шляхом розрахунку
 - б) шляхом вимірювань на сформованому зображенні з п.1.1. cрselect
 - 1.3. Обчислити координати контрольних точок після застосування до сформованого зображення заданого геометричного перетворення
 - 1.4. Визначити матрицю та операцію геометричного перетворення в MATLAB та отримати зображення-результат перетворення
 - 1.5. Визначити на зображенні-результаті перетворення координати контрольних точок, порівняти їх з результатами п.1.3.
 - 1.6. Виконати в MATLAB зворотнє геометричне перетворення шляхом:
 - а) застосування матриці оберненого перетворення відносно результатів п.1.4.
 - б) застосування геометричного перетворення на основі координат початкових та результуючих контрольних точок з п.1.1.5 та п. 1.2.
 - 1.7. Визначити на зображенні, відновленому шляхом зворотнього геометричного перетворення, координати контрольних точок, порівняти їх з результатами п.1.2.

АБВ АБВ



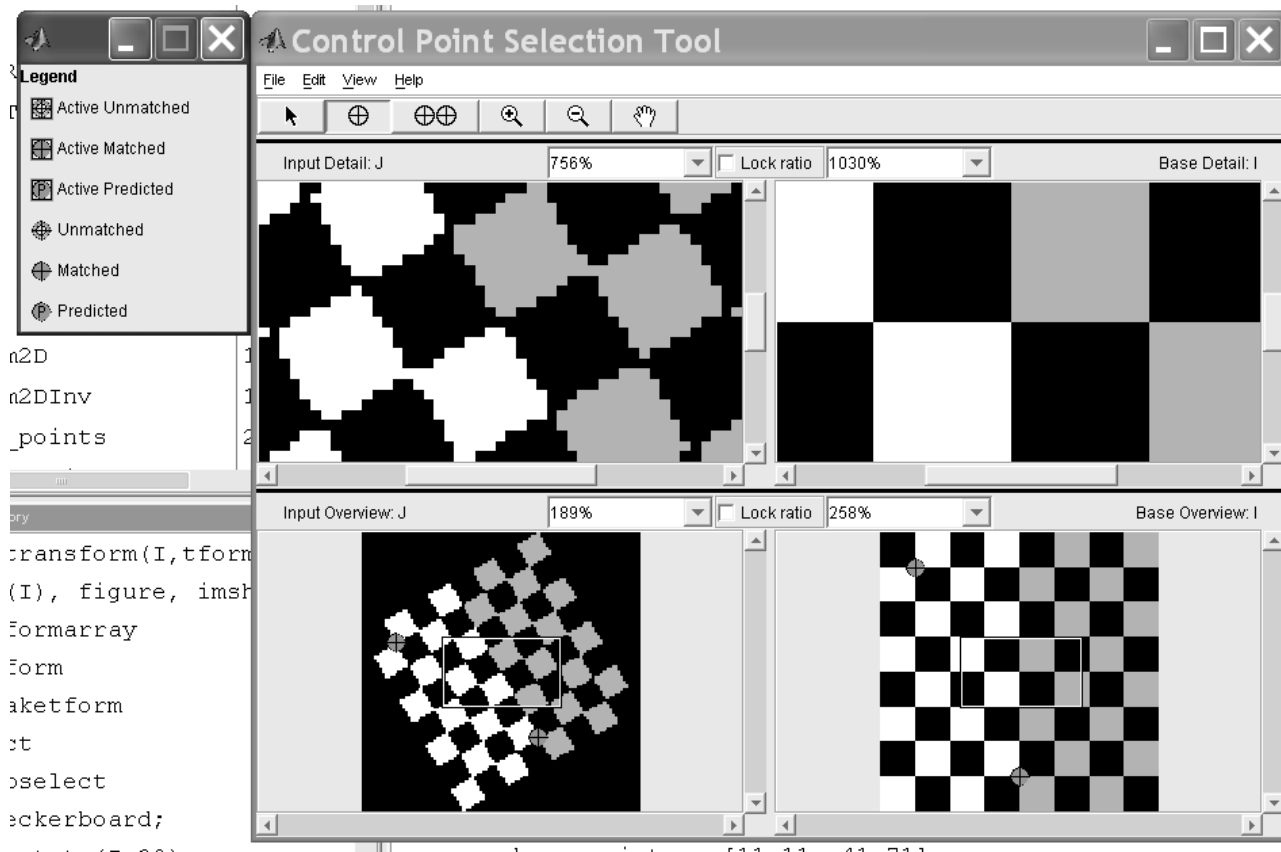
```

Test2D.m
Im2D=zeros(500,500);
Im2D(round(500/2-120/2):round(500/2+120/2),round(500/2-
120/2):round(500/2+120/2))=1;
Im2D=imrotate(Im2D,0); % в градусах против час стр
figure; imshow(Im2D); title('ISX IZOB');
R2D=[cos(30*pi/180) sin(30*pi/180) 0
     -sin(30*pi/180) cos(30*pi/180) 0
      0 0 1]; % в рад по час стр
S2D=[1.5 0 0
     0 1.2 0
     0 0 1]; % растяжение раз
T2D=[1 0 0
     0 1 0
     35 30 1]; % перенос вправо и вниз
M2D=T2D*S2D*R2D;
TForm2D=maketform('affine',M2D);
Im2DTransform=imtransform(Im2D,TForm2D,'XData',[1 500], 'YData',
[1 500]);
figure; imshow(Im2DTransform); title('PREOB IZOB');
TForm2DInv=maketform('affine',M2D^-1);
Im2DRestore1=imtransform(Im2DTransform,TForm2DInv,'XData',[1
500], 'YData',[1 500]);
figure; imshow(Im2DRestore1); title('VOSST IZOB');

```

Start tool with workspace images and points:

```
I = checkerboard;
J = imrotate(I,30);
base_points = [11 11; 41 71];
input_points = [14 44; 70 81];
cpselect(J,I,input_points,base_points);
```



IMPIXEL

Определение значения пикселей*Синтаксис:*

```

P=impixel()
P=impixel(I)
P=impixel(X, map)
P=impixel(RGB)
P=impixel(I, c, r)
P=impixel(X, map, c, r)
P=impixel(RGB, c, r)
[c, r, P]=impixel(...)
P=impixel(XData, YData, I, xi, yi)
P=impixel(XData, YData, X, map, xi, yi)
P=impixel(XData, YData, RGB, xi, yi)
[xi yi P]=impixel(XData, YData, ...)

```

Описание:

Функция `P=impixel()` возвращает значения красной (R), зеленой (G) и синей (B) составляющих цвета для определенных пикселей изображения, находящегося в текущем окне. У полутоновых изображений значения составляющих совпадают. Пиксели необходимо определять интерактивно. Для выбора пикселя следует установить на него курсор и нажать на левую клавишу мыши. Таким образом можно выбрать несколько пикселей. На изображении они помечаются крестиками. Предыдущий выбранный пиксел можно удалить, если нажать на клавиши Backspace или Delete. Нажатие на правую клавишу мыши или двойной щелчок левой клавишей выбирает последний пиксел и завершает процесс выбора пикселей. Также завершить процесс выбора пикселей без указания последнего пикселя можно нажатием на клавишу Enter. Когда процесс выбора пикселей завершен, функция `impixel` возвращает в `P` матрицу $N \times 3$, в столбцах которой содержатся значения R-, G-, B-составляющих N выбранных пикселей.

Функции `P=impixel(I)`, `P=impixel(X, map)`, `P=impixel(RGB)` выводят на экран соответственно полутоновое изображение `I`, палитровое `X` с палитрой `map` и полноцветное `RGB` и переходят в режим выбора пикселей. Далее работа осуществляется полностью аналогично функции `impixel` без параметров.

При вызове функции `impixel` можно явно (не интерактивно) задать координаты пикселей, значения составляющих которых требуется получить. Причем координаты могут быть заданы как в пиксельной, так и в пространственной системе координат.

Для задания координат в пиксельной системе координат для различных типов изображений следует воспользоваться одной из функций `P=impixel(I, c, r)`, `P=impixel(X, map, c, r)`, `P=impixel(RGB, c, r)`, где `c` и `r` - векторы значений строк и столбцов требуемых пикселей. При этом в i -той строке матрицы `P` будут находиться R-, G-, B-составляющие пикселя из $c(i)$ столбца и $r(i)$ строки.

Если для функций, рассмотренных выше, дополнительно определить два выходных параметра `c` и `r`, то функции `[c, r, P]=impixel(...)` возвратят в них пиксельные координаты выбранных пикселей.

Для задания координат в пространственной системе координат для различных типов изображений следует воспользоваться одной из функций `P=impixel(XData, YData, I, xi, yi)`, `P=impixel(XData, YData, X, map, xi, yi)`, `P=impixel(XData, YData, RGB, xi, yi)`, где xi и yi - координаты, а двухэлементные векторы `XData` и `YData` определяют диапазон изменений значений по осям пространственной системы координат. Параметры `XData` и `YData` при вызове функций можно опустить. В этом случае будет использоваться пространственная система координат со значениями по осям по умолчанию.

Если для трех функций, работающих с пространственной системой координат, дополнительно определить два выходных параметра xi и yi , то функции `[xi yi P]=impixel(XData, YData, ...)` возвратят в них значения координат выбранных пикселей.

```

Test2DNew.m
% исходные данные
clear all;
M=400; N=500; H=120; L=100; Alfa=30;
Tetaz=-30; Tetay=0; Tetax=0;
Tz=10; Ty=30; Tx=35;
Sz=1; Sy=1.2; Sx=1.5;

% формирование исходного изображения
Im2D=zeros(M,N);
Im2D(round(M/2-L/2):round(M/2+L/2),round(N/2-H/2):round(N/2+H/2))=1;
Im2D=imrotate(Im2D,Alfa,'crop'); % в градусах против час стр
figure; imshow(Im2D); title('ISX IZOB');

% матрицы прямого геометрического преобразования
R2D=[cos(Tetaz*pi/180) sin(Tetaz*pi/180) 0
     -sin(Tetaz*pi/180) cos(Tetaz*pi/180) 0
     0 0 1]; % в рад по час стр
S2D=[Sx 0 0
     0 Sy 0
     0 0 1]; % растяжение раз
T2D=[1 0 0
     0 1 0
     Tx Ty 1]; % перенос вправо и вниз
TFormR2D=maketform('affine',R2D);
TFormS2D=maketform('affine',S2D);
TFormT2D=maketform('affine',T2D);

% прямое геометрическое преобразование исх изображения
Im2DTransform1=imtransform(Im2D,TFormR2D,...
    'UData',[-round(N/2)+1 round(N/2)],'VData',[-round(M/2)+1
round(M/2)],...
    'XData',[-round(N/2)+1 round(N/2)],'YData',[-round(M/2)+1
round(M/2)]);
Im2DTransform2=imtransform(Im2DTransform1,TFormS2D,...
    'UData',[-round(N/2)+1 round(N/2)],'VData',[-round(M/2)+1
round(M/2)],...
    'XData',[-round(N/2)+1 round(N/2)],'YData',[-round(M/2)+1
round(M/2)]);
Im2DTransform3=imtransform(Im2DTransform2,TFormT2D,...
    'UData',[-round(N/2)+1 round(N/2)],'VData',[-round(M/2)+1
round(M/2)],...
    'XData',[-round(N/2)+1 round(N/2)],'YData',[-round(M/2)+1
round(M/2)]);

% вывод преобразованного изображения
figure; subplot(2,2,1); imshow(Im2D); title('Isx Izob');
subplot(2,2,2); imshow(Im2DTransform1); title('Povorot');
subplot(2,2,3); imshow(Im2DTransform2); title('Mashtab');
subplot(2,2,4); imshow(Im2DTransform3); title('Peremesch');

% матрицы обратного геометрического преобразования

```



```

R2DInv=[cos (Tetaz*pi/180) -sin (Tetaz*pi/180) 0
        sin (Tetaz*pi/180)  cos (Tetaz*pi/180) 0
        0                      0                  1]; % в рад по час
стр
S2DInv=[1/Sx 0      0
        0     1/Sy 0
        0     0     1]; % растяжение раз
T2DInv=[1   0   0
        0   1   0
        -Tx -Ty 1]; % перенос вправо и вниз
TFormT2DInv=maketform('affine',T2DInv);
TFormS2DInv=maketform('affine',S2DInv);
TFormR2DInv=maketform('affine',R2DInv);

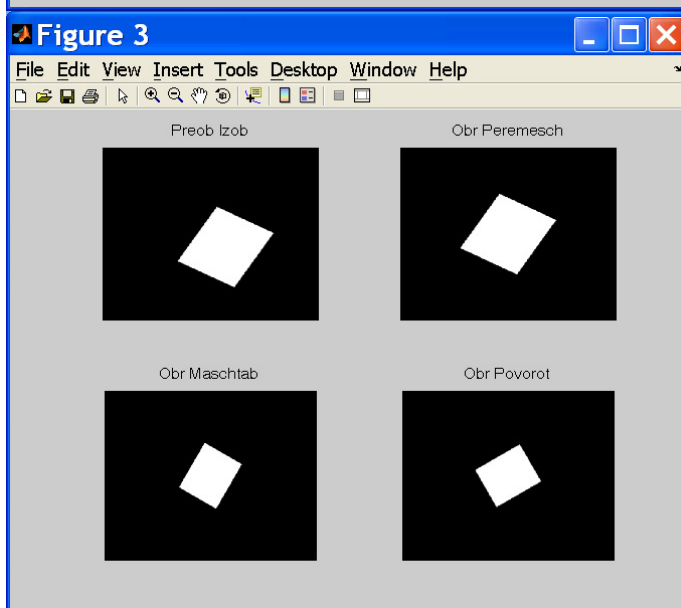
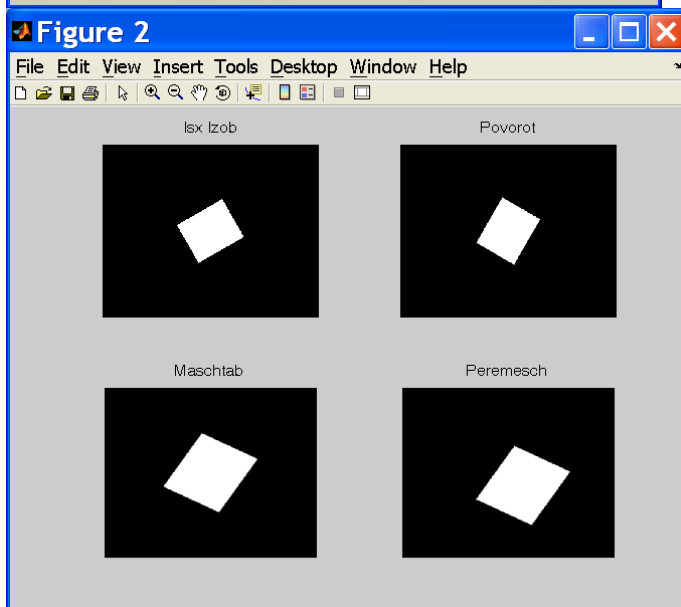
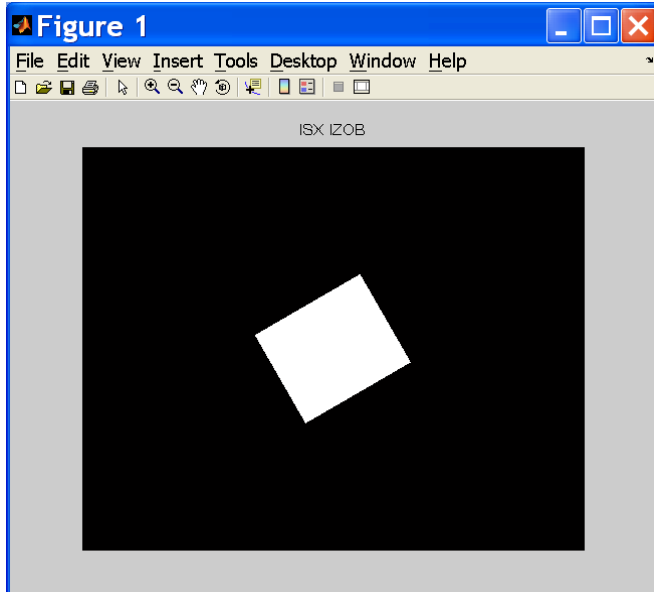
% обратное геометрическое преобразование преоб изображения
Im2DRestore1=imtransform (Im2DTransform3,TFormT2DInv,...
    'UData', [-round (N/2)+1 round (N/2)], 'VData', [-round (M/2)+1
round (M/2)], ...
    'XData', [-round (N/2)+1 round (N/2)], 'YData', [-round (M/2)+1
round (M/2)]);
Im2DRestore2=imtransform (Im2DRestore1,TFormS2DInv,...
    'UData', [-round (N/2)+1 round (N/2)], 'VData', [-round (M/2)+1
round (M/2)], ...
    'XData', [-round (N/2)+1 round (N/2)], 'YData', [-round (M/2)+1
round (M/2)]);
Im2DRestore3=imtransform (Im2DRestore2,TFormR2DInv,...
    'UData', [-round (N/2)+1 round (N/2)], 'VData', [-round (M/2)+1
round (M/2)], ...
    'XData', [-round (N/2)+1 round (N/2)], 'YData', [-round (M/2)+1
round (M/2)]);

% вывод восстановленного изображения
figure; subplot (2,2,1); imshow (Im2DTransform3); title ('Preob
Izob');
subplot (2,2,2); imshow (Im2DRestore1); title ('Obr Peremesch');
subplot (2,2,3); imshow (Im2DRestore2); title ('Obr Mashtab');
subplot (2,2,4); imshow (Im2DRestore3); title ('Obr Povорот');

% получение координат точек
figure; [XIsx, YIsx, PIsx]=impixel (Im2D)
[XTransform3, YTransform3, PTransform3]=impixel (Im2DTransform3)
[XRestore3, YRestore3, PRestore3]=impixel (Im2DRestore3)

```

$M=400$; $N=500$; $H=120$; $L=100$; $\text{Alfa}=30$; $\text{Tetaz}=-30$; $\text{Tetay}=0$; $\text{Tetax}=0$;
 $\text{Tz}=10$; $\text{Ty}=30$; $\text{Tx}=35$; $\text{Sz}=1$; $\text{Sy}=1.2$; $\text{Sx}=1.5$;



XIsx =

173
276
327
223

YIsx =

188
127
214
274

PIsx =

1	1	1
1	1	1
1	1	1
1	1	1

XTransform3 =

175
265
394
305

YTransform3 =

263
139
199
323

PTransform3 =

0.5916	0.5916	0.5916
0.9717	0.9717	0.9717
0.9300	0.9300	0.9300
0.5615	0.5615	0.5615

XRestore3 =

173
276
325
222

YRestore3 =

187
127
213
273

PRestore3 =

0.5964	0.5964	0.5964
0.7326	0.7326	0.7326
0.9800	0.9800	0.9800
0.6676	0.6676	0.6676

>>

Трёхмерные преобразования

```
% исходные данные
```

```
clear all;
```

```
M=400; N=500; H=120; L=100; Alfa=0;
```

```
Tetaz=-30; Tetay=-30; Tetax=0;
```

```
Tz=10; Ty=30; Tx=35;
```

```
Sz=0.8; Sy=1.2; Sx=1.5;
```

```
% формирование исходного изображения
```

```
% Im2D=im2double(imread('c:\АБВ.bmp'));
```

```
% [M, N]=size(Im2D);
```

```
Im2D=zeros(M,N);
```

```
Im2D(round(M/2-L/2):round(M/2+L/2),round(N/2-H/2):round(N/2+H/2))=1;
```

```
Im2D=imrotate(Im2D,Alfa,'crop'); % в градусах против час стр  
figure; imshow(Im2D); title('ISX IZOB');
```

```
% матрицы прямого геометрического преобразования
```

```
R3DX=[1 0 0 0  
0 cos(Tetax*pi/180) sin(Tetax*pi/180) 0  
0 -sin(Tetax*pi/180) cos(Tetax*pi/180) 0  
0 0 0 1]; % в рад по час
```

```
стр
```

```
R3DY=[cos(Tetay*pi/180) 0 -sin(Tetay*pi/180) 0  
0 1 0 0  
sin(Tetay*pi/180) 0 cos(Tetay*pi/180) 0  
0 0 0 1];
```

```
% в рад по час стр
```

```
R3DZ=[cos(Tetaz*pi/180) sin(Tetaz*pi/180) 0 0  
-sin(Tetaz*pi/180) cos(Tetaz*pi/180) 0 0  
0 0 1 0  
0 0 0 1]; % в рад по час
```

```
стр
```

```
R3D=R3DZ*R3DY*R3DX;
```

```
S3D=[Sx 0 0 0  
0 Sy 0 0  
0 0 Sz 0  
0 0 0 1]; % растяжение раз
```

```
T3D=[1 0 0 0  
0 1 0 0  
0 0 1 0  
Tx Ty Tz 1]; % перенос вправо и вниз
```

```
TFormR3D=maketform('affine',R3D);
```

```
TFormS3D=maketform('affine',S3D);
```

```
TFormT3D=maketform('affine',T3D);
```

```
MP3D=[1 0 0 0  
0 1 0 0  
0 0 0 0  
0 0 0 1]; % ортографическая проекция
```

```
TFormMP3D=maketform('projective',MP3D);
```

```
% прямое геометрическое преобразование исх изображения
```

```
A3D=zeros(M*N,3);
```

```

k=0;
for j=1:M
    for i=1:N
        if Im2D(j,i)>=0.5
            k=k+1;
            A3D(k,1)=i-round(N/2)+1; A3D(k,2)=j-round(M/2)+1;
        end;
    end;
end;
B3D=A3D(1:k,:);
C3D1=tformfwd(TFormR3D, B3D);
C3D2=tformfwd(TFormS3D, C3D1);
C3D3=tformfwd(TFormT3D, C3D2);
D3D1=tformfwd(TFormMP3D, C3D1);
D3D2=tformfwd(TFormMP3D, C3D2);
D3D3=tformfwd(TFormMP3D, C3D3);
Im2DTransform1=zeros(M,N);
for i=1:k
    x=round(D3D1(i,1)+round(N/2)-1); y=round(D3D1(i,2)+round(M/2)-1);
    Im2DTransform1(y,x)=1;
end;
Im2DTransform2=zeros(M,N);
for i=1:k
    x=round(D3D2(i,1)+round(N/2)-1); y=round(D3D2(i,2)+round(M/2)-1);
    Im2DTransform2(y,x)=1;
end;
Im2DTransform3=zeros(M,N);
for i=1:k
    x=round(D3D3(i,1)+round(N/2)-1); y=round(D3D3(i,2)+round(M/2)-1);
    Im2DTransform3(y,x)=1;
end;

% вывод преобразованного изображения
figure; subplot(2,2,1); imshow(Im2D); title('Isx Izob');
subplot(2,2,2); imshow(Im2DTransform1); title('Proek Povorot');
subplot(2,2,3); imshow(Im2DTransform2); title('Proek Mashtab');
subplot(2,2,4); imshow(Im2DTransform3); title('Proek Peremesch');

% матрицы обратного геометрического преобразования
% R3DXInv=[1 0 0 0
%          0 cos(Tetax*pi/180) -sin(Tetax*pi/180) 0
%          0 sin(Tetax*pi/180) cos(Tetax*pi/180) 0
%          0 0 0 1]; % в рад по
% час стр
% R3DYInv=[cos(Tetay*pi/180) 0
%          sin(Tetay*pi/180) 0
%          0 1 0
%          -sin(Tetay*pi/180) 0
%          cos(Tetay*pi/180) 0

```

```

%           0           0           0
1]); % в рад по час стр
% R3DZInv=[cos(Tetaz*pi/180) -sin(Tetaz*pi/180) 0 0
%           sin(Tetaz*pi/180)  cos(Tetaz*pi/180) 0 0
%           0           0           1 0
%           0           0           0 1]; % в рад по
час стр
% R3DInv=R3DZInv*R3DYInv*R3DXInv;
% S3DInv=[1/Sx  0  0  0
%          0  1/Sy 0  0
%          0  0  1/Sz 0
%          0  0  0  1]; % растяжение раз
% T3DInv=[1  0  0  0
%          0  1  0  0
%          0  0  1  0
%          -Tx -Ty -Tz  1]; % перенос вправо и вниз
% TFormR3DInv=maketform('affine',R3DInv);
% TFormS3DInv=maketform('affine',S3DInv);
% TFormT3DInv=maketform('affine',T3DInv);

% обратное геометрическое преобразование преоб изображения
C3D2Restore=tforminv(TFormT3D, C3D3);
C3D1Restore=tforminv(TFormS3D, C3D2Restore);
B3DRestore=tforminv(TFormR3D, C3D1Restore);
D3D3Restore=tformfwd(TFormMP3D, C3D2Restore);
D3D2Restore=tformfwd(TFormMP3D, C3D1Restore);
D3D1Restore=tformfwd(TFormMP3D, B3DRestore);
Im2DRestore3=zeros(M,N);
for i=1:k
    x=round(D3D3Restore(i,1)+round(N/2)-1);
    y=round(D3D3Restore(i,2)+round(M/2)-1);
    Im2DRestore3(y,x)=1;
end;
Im2DRestore2=zeros(M,N);
for i=1:k
    x=round(D3D2Restore(i,1)+round(N/2)-1);
    y=round(D3D2Restore(i,2)+round(M/2)-1);
    Im2DRestore2(y,x)=1;
end;
Im2DRestore1=zeros(M,N);
for i=1:k
    x=round(D3D1Restore(i,1)+round(N/2)-1);
    y=round(D3D1Restore(i,2)+round(M/2)-1);
    Im2DRestore1(y,x)=1;
end;

% вывод восстановленного изображения
figure; subplot(2,2,1); imshow(Im2DTransform3); title('Preob
Izob');
subplot(2,2,2); imshow(Im2DRestore3); title('Obr Peremesch');
subplot(2,2,3); imshow(Im2DRestore2); title('Obr Mashtab');
subplot(2,2,4); imshow(Im2DRestore1); title('Obr Povорот');

```

```

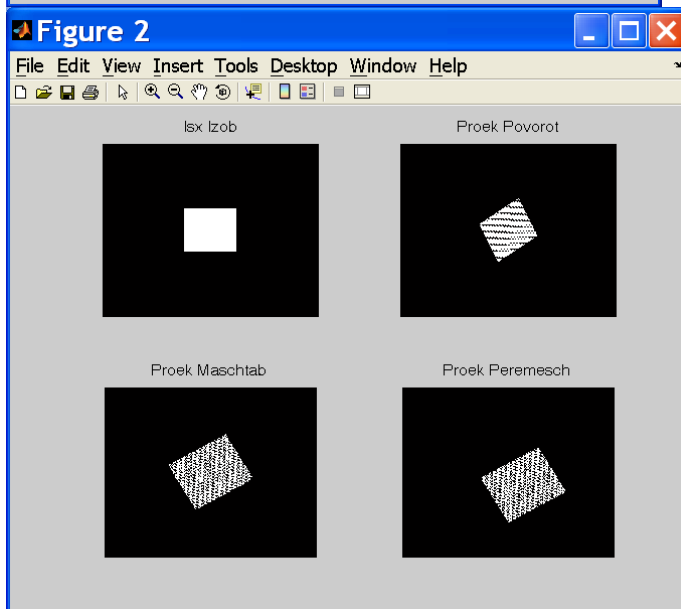
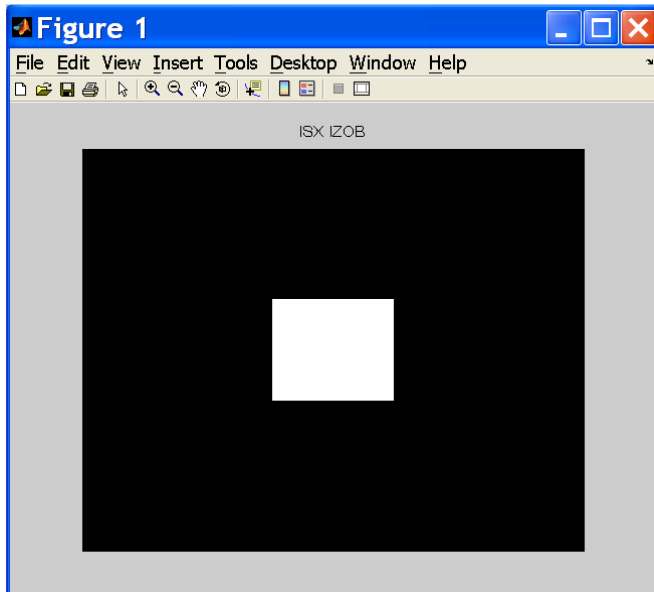
% получение координат точек
figure; [XIsx, YIsx, PIsx]=impixel(Im2D)
[XTransform3, YTransform3, PTransform3]=impixel(Im2DTransform3)
[XRestore3, YRestore3, PRestore3]=impixel(Im2DRestore1)

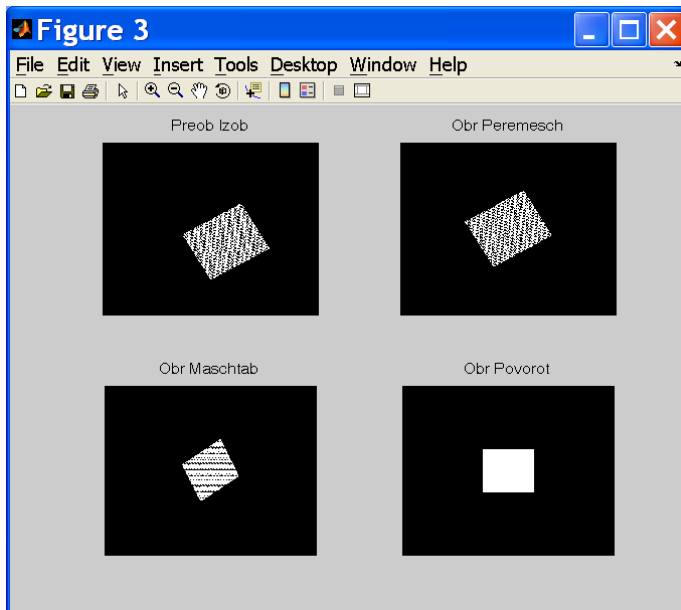
```

```

M=400; N=500; H=120; L=100; Alfa=0;
Tetaz=-30; Tetay=-30; Tetax=0;
Tz=10; Ty=30; Tx=35;
Sz=0.8; Sy=1.2; Sx=1.5;

```





XIsx =

```

188
310
189
311

```

YIsx =

```

149
149
251
251

```

PIsx =

```

0    0    0
0    0    0
0    0    0
0    0    0

```

XTransform3 =

```

185
320
387
251

```

YTransform3 =

```

214

```


141
245
318

PTransform3 =

0	0	0
0	0	0
0	0	0
0	0	0

XRestore3 =

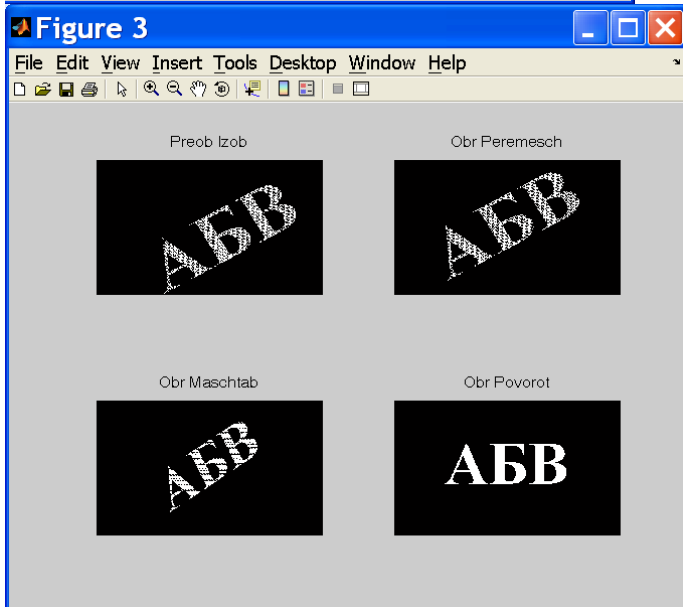
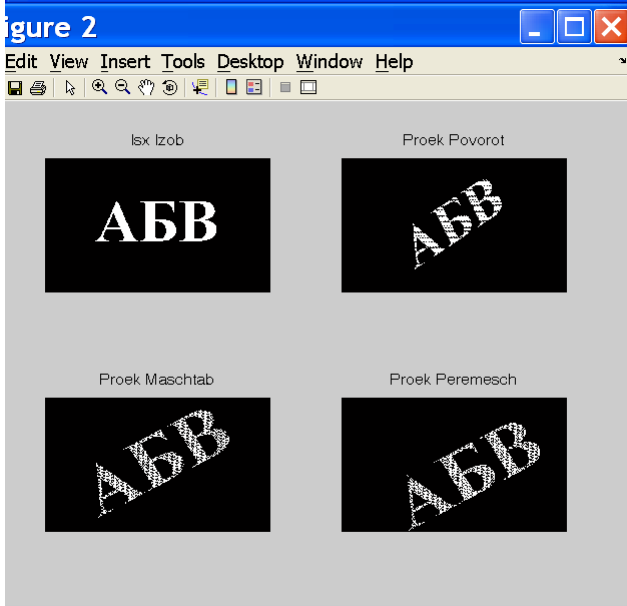
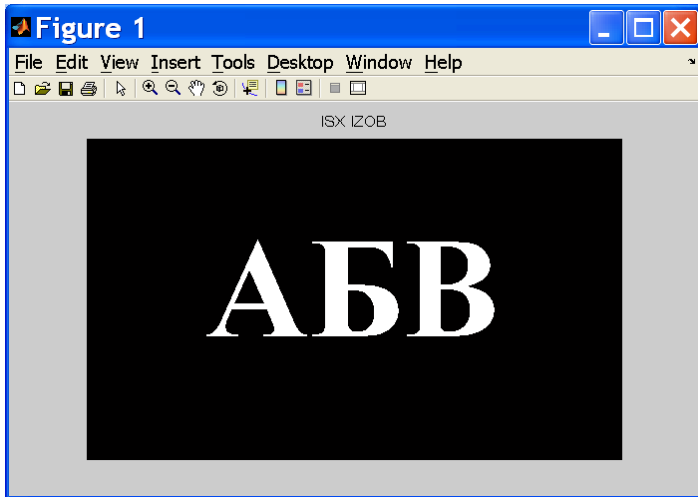
189
310
189
310

YRestore3 =

150
149
251
251

PRestore3 =

0	0	0
0	0	0
0	0	0
0	0	0



XIsx =

172
120
221
225
307
227
321
322

YIsx =

103
198
198
105
104
198
102
199

PIsx =

1	1	1
0	0	0
1	1	1
0	0	0
1	1	1
1	1	1
0	0	0
0	0	0

XTransform3 =

157
162
273
222
282
308
327
387

YTransform3 =

187
317
259

157
254
110
99
197

PTransform3 =

1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	1	1
0	0	0

XRestore3 =

172
121
219
225
225
306
321
322

YRestore3 =

103
197
198
103
198
104
102
198

PRestore3 =

1	1	1
1	1	1
1	1	1
0	0	0
0	0	0
1	1	1
0	0	0
1	1	1

>>