**Лекція 3. Геометричні перетворення зображень в комп’ютерній графіці.**

**КУРСОВА РОБОТА «Розробка методів геометричних перетворень об’єктів»**

**Геометричні перетворення двовимірних об’єктів на площині**

1) Формування зображення на площині (отримуємо **початкове зображення**)

2) Визначення координат контрольних точок на початковому зображенні (розрахунок за формулами та вимірювання на зображенні)

3) Пряме геометричне перетворення – поворот, масштабування, переміщення (отримуємо **перетворене зображення**)

4) Визначення координат контрольних точок на перетвореному зображенні (розрахунок за формулами та вимірювання на зображенні)

5) Зворотнє геометричне перетворення – переміщення, масштабування, поворот (отримуємо **відновлене зображення**)

6) Визначення координат контрольних точок на відновленому зображенні (розрахунок за формулами та вимірювання на зображенні)

7) Порівняння результатів визначення координат контрольних точок

- в пп. 2), 4) та 6) для двох варіантів визначення (розрахунок за

формулами та вимірювання на зображенні)

- результатів пп. 2) і 6) для початкового і відновленого зображень

**Дані варіанту завдання**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер варіанту | Розмір зображення | | Двовимірна геометрична фігура – прямокутник | | |
| ширина *N*,  дискр. точок | висота *M*,  дискр. точок | ширина *H*,  диск. точок | висота *L*,  диск. точок | кут повороту *alfa*, градусів відносно *Ох*  проти год. стрілки |
| 3 | 350 | 400 | 90 | 120 | 20 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варіанту | Параметри геометричного перетворення | | | | | | | | |
| Зсув, дискр. точок | | | Масштабування | | | Кут повороту, градусів проти год. стрілки відносно вказаної осі | | |
| ***Tx*** | ***Ty*** | *Tz* | ***Sx*** | ***Sy*** | *Sz* | *Tetax* | *Tetay* | ***Tetaz*** |
| 3 | **30** | **30** | 20 | **0,7** | **0,9** | 1,5 | 30 | 45 | **30** |

% початкові дані

clear all;

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

Tetaz=-30;

Ty=30; Tx=35;

Sy=1.2; Sx=1.5;

**Формування початкового зображення**

C:\Documents and Settings\jup\Рабочий стол\Microsoft Visio Drawing.emf

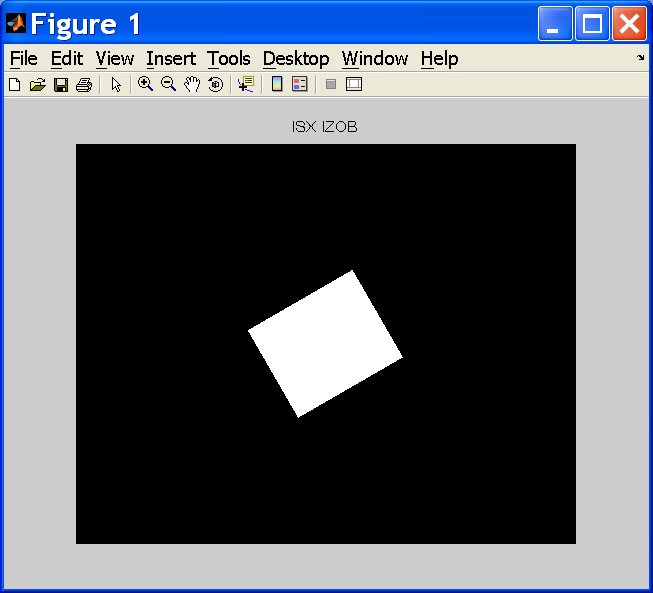
% Формування початкового зображення

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 проти год. стр.

figure; imshow(Im2D); title('ISX IZOB');

****

**Теоретичні основи геометричних перетворень на площині**

Однорідні координати використовуються для опису геометричних перетворень

Векторно-матрична форма геометричних перетворень

Координати точки А1(х1, у1) – це вектор-рядок [x1 y1]

Афінне перетворення (поворот, масштабування, переміщення або їх композиція) – це матриця

Декартові координати точки А1(х1, у1) – це вектор-рядок [x1д y1д]

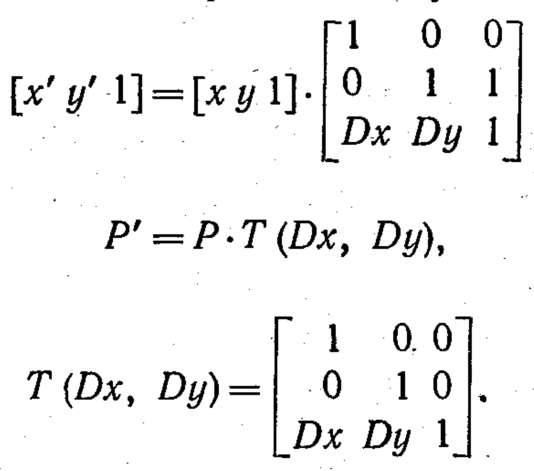
**Однорідні координати** точки А1(х1, у1) – це вектор-рядок [x1о y1о w1],

де w1 – масштабний коефіцієнт

x1д=x1o/w1 y1д=y1o/w1

В найпростішому випадку w1=1, однорідні координати [x1 y1 1]

**Матричний опис переміщення** (зсуву) на Dх точок праворуч та Dу точок вгору (якщо ліворуч або вниз, то величини Dх і Dу зі знаком мінус)



**Приклад**

Координати точки до переміщення [10 -40], однорідні координати [10 -40 1]

Переміщення на 20 точок праворуч і 30 точок вгору

| 1 0 0 |

Матриця переміщення | 0 1 0 |

| 20 30 1 |

x’=10×1+(-40)×0+1×20=30

y’=10×0+(-40)×1+1×30=-10

w’=10×0+(-40)×0+1×1=1

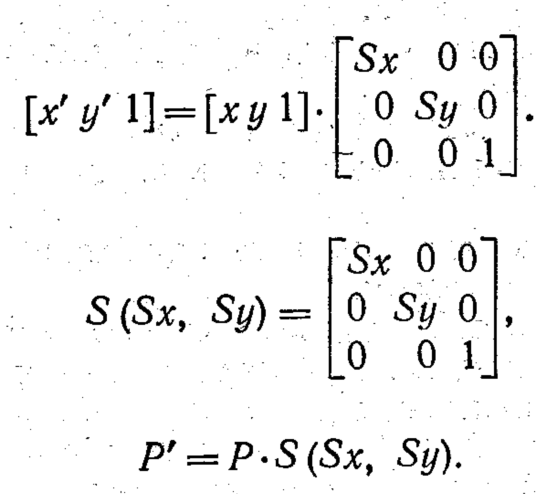
Однорідні координати точки після переміщення [30 -10 1]

Декартові координати [30/1 -10/1]= [30 -10]

**Матричний опис масштабування** в Sх разів по горизонталі та Sу разів по вертикалі

Розтяг (збільшення) об’єкту Sх>1 Sу>1

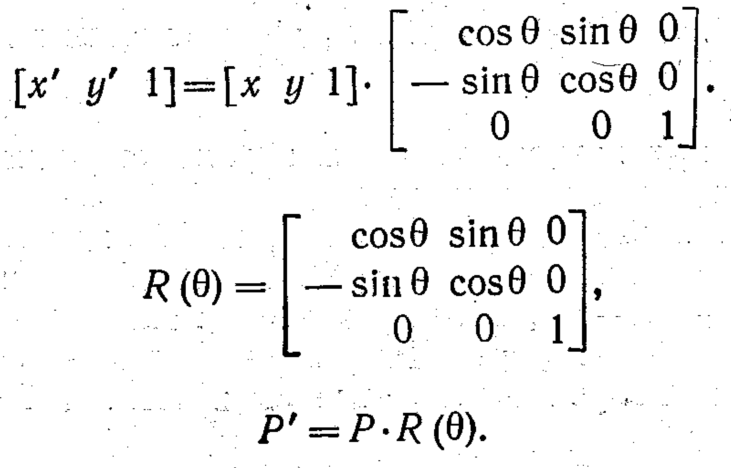
Стиснення (зменшення) об’єкту 0<Sх<1 0<Sу<1



**Матричний опис повороту** на Тетаz градусів

Проти год. стр. Тетаz>0

За год. стр. Тетаz<0



**Композиція геометричних перетворень**

A’ = A×R×S×T

% 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)]);

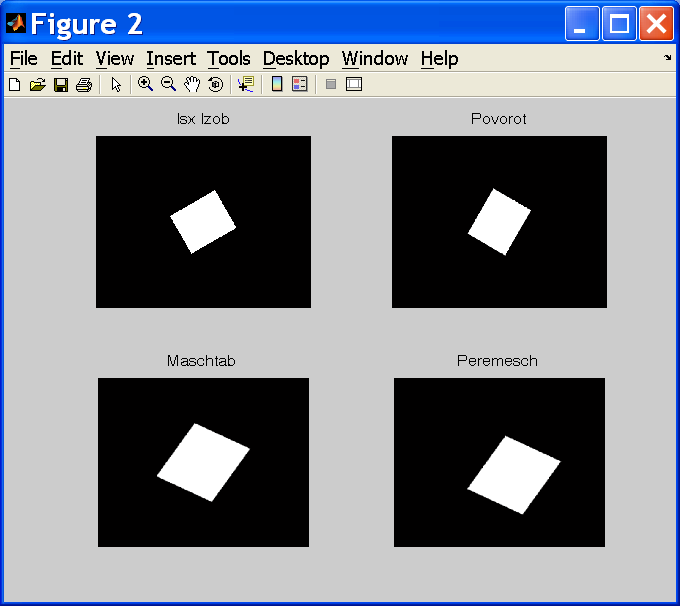
% вывод преобразованного изображения

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('Maschtab');

subplot(2,2,4); imshow(Im2DTransform3); title('Peremesch');

****

**Зворотнє геометричне перетворення**

Порядок застосування афінних перетворень

Пряме – Поворот, масштабування, переміщення

Зворотнє – Переміщння, масштабування, поворот

В матрицях афінних перетворень потрібно змінити:

Переміщення Dх і Dу взяти зі знаком мінус

Dхзв= -Dхпр Dузв= -Dупр

Масштабування Sх і Sу взяти обернено пропорційні величини

Sхзв= 1/Sхпр Sузв= 1/Sупр

Поворот

Тетаz зв = - Тетаz пр

та врахувати, що sin непарна функція, cos парна

**Композиція зворотніх геометричних перетворень**

Aвідн = A’×Tзв ×Sзв×Rзв

% 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)]);

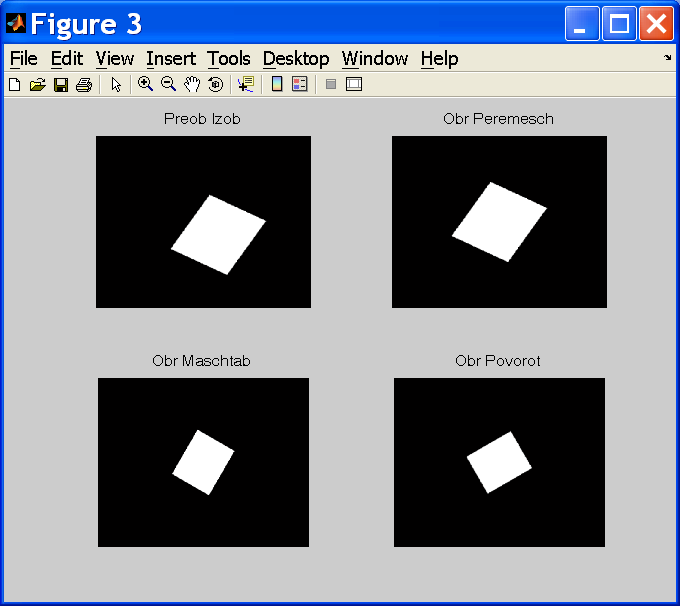
% вывод восстановленного изображения

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 Maschtab');

subplot(2,2,4); imshow(Im2DRestore3); title('Obr Povorot');

****

**Порівняння координат точок**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Контроль­на точка (вершини прямокут­ника) | Початкове зображення | | Перетворене зображення | | Відновлене зображення | |
| Розраху­нок за формула­ми | Вимірю­вання на зобра­женні | Розраху­нок за формула­ми | Вимірю­вання на зобра­женні | Розраху­нок за формула­ми | Вимірю­вання на зобра­женні |
| А1 |  |  |  |  |  |  |
| A2 |  |  |  |  |  |  |
| A3 |  |  |  |  |  |  |
| A4 |  |  |  |  |  |  |

А1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А2

| |

| |

| |

A4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ A3

**Вимірювання на зображенні**

% получение координат точек

figure; [XIsx, YIsx, PIsx]=impixel(Im2D)

[XTransform3, YTransform3, PTransform3]=impixel(Im2DTransform3)

[XRestore3, YRestore3, PRestore3]=impixel(Im2DRestore3)

**cpselect**

IMPIXEL

Определение значения пикселов

Синтаксис:

P=impixel(I)

[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 пикселов изображения, находящегося в текущем окне. У полутоновых изображений значения составляющих совпадают. Пикселы необходимо определять интерактивно. Для выбора пиксела следует установить на него курсор и нажать на левую клавишу мыши. Таким образом можно выбрать несколько пикселов. 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 пикселов. Также завершить процесс выбора пикселов без указания последнего пиксела можно нажатием на клавишу Enter. Когда процесс выбора пикселов завершен, функция impixel возвращает в Р матрицу Nx3, в которой содержатся значения R-, G-, B-составляющих N выбранных пикселов.

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 пикселов.

**XIsx =**

**173**

**276**

**327**

**223**

**YIsx =**

**188**

**127**

**214**

**274**

**PIsx =**

**1 1 1**

**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**

**Геометричні перетворення об’єктів в тривимірному просторі**

1) Формування зображення на площині (отримуємо **початкове зображення**) та його перенос у тривимірний простір за умови z=0

2) Визначення координат контрольних точок на початковому зображенні (вимірювання на зображенні)

3) Пряме геометричне перетворення у тривимірному просторі – поворот, масштабування, переміщення. Шляхом проекційного перетворення отримуємо **перетворене зображення тривимірних об’єктів**

4) Визначення координат контрольних точок на перетвореному зображенні (вимірювання на зображенні)

5) Зворотнє геометричне перетворення у тривимірному просторі – переміщення, масштабування, поворот. Шляхом проекційного перетворення отримуємо **відновлене зображення тривимірних об’єктів**

6) Визначення координат контрольних точок на відновленому зображенні (вимірювання на зображенні)

7) Порівняння результатів визначення координат контрольних точок

- результатів пп. 2) і 6) для початкового і відновленого зображень

**Дані варіанту завдання**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер варіанту | Розмір зображення | | Двовимірна геометрична фігура – прямокутник | | | Зображення літер |
| ширина *N*,  дискр. точок | висота *M*,  дискр. точок | ширина *H*,  диск. точок | висота *L*,  диск. точок | кут повороту *alfa*, градусів відносно *Ох*  проти год. стрілки |
| 3 | 350 | 400 | 90 | 120 | 20 | ВГД |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варіанту | Параметри геометричного перетворення | | | | | | | | |
| Зсув, дискр. точок | | | Масштабування | | | Кут повороту, градусів проти год. стрілкі відновно вказаної осі | | |
| *Tx* | *Ty* | *Tz* | *Sx* | *Sy* | *Sz* | *Tetax* | *Tetay* | *Tetaz* |
| 3 | 30 | 30 | 20 | 0,7 | 0,9 | 1,5 | 30 | 45 | 30 |

**% исходные данные**

**clear all;**

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

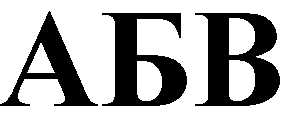
**Tetaz=-30; Tetay=-30; Tetax=0;**

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

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

**Формування початкового зображення**

В растровому графічному редакторі, збереження в графічному файлі, завантаження файла в робочий простір Матлаб

****

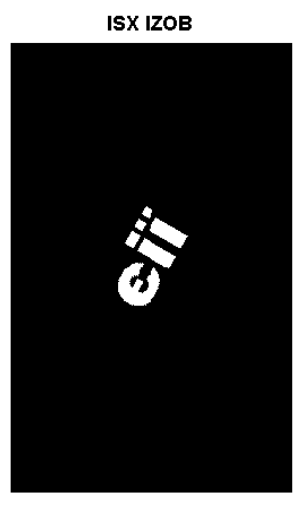
**% формированаие исходного изображения**

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

**[M, N]=size(Im2D);**

**Im2D=imrotate(Im2D,Alfa,'crop'); % в градусах против час стр**

**figure; imshow(Im2D); title('ISX IZOB');**



**Теоретичні основи геометричних перетворень в тривимірному просторі**

Однорідні координати використовуються для опису геометричних перетворень

Векторно-матрична форма геометричних перетворень

Координати точки А1(х1, у1, z1) – це вектор-рядок [x1 y1 z1]

Афінне перетворення (поворот, масштабування, переміщення або їх композиція) – це матриця

Декартові координати точки А1(х1, у1, z1) – це вектор-рядок [x1д y1д z1д]

**Однорідні координати** точки А1(х1, у1, z1) –

це вектор-рядок [x1о y1о z1o w1],

де w1 – масштабний коефіцієнт

x1д=x1o/w1 y1д=y1o/w1 z1д=z1o/w1

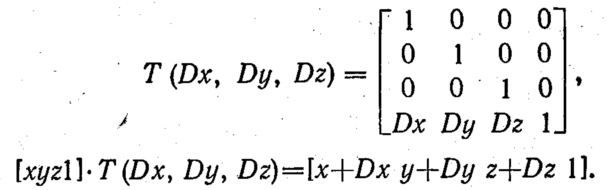
В найпростішому випадку w1=1, z1д=0, однорідні координати [x1 y1 0 1]

**Матричний опис переміщення** (зсуву) на Dх точок праворуч, Dу точок вгору та Dz точок **до себе** (якщо ліворуч, вниз або від себе, то величини Dх, Dу і Dz зі знаком мінус)

Це для лівосторонньої системи координат (вісь Оz спрямована на себе).

У правосторонній - вісь Оz спрямована від себе

Всі обчислення в одній системі.



**Приклад**

Координати точки до переміщення [10 -40 0],

однорідні координати [10 -40 0 1]

Переміщення на 20 точок праворуч і 30 точок вгору, на себе 0 точок

| 1 0 0 0 |

Матриця переміщення | 0 1 0 0 |

| 0 0 1 0 |

| 20 30 0 1 |

x’=10×1+(-40)×0+0×0+1×20=30

y’=10×0+(-40)×1+0×0+1×30=-10

z’=10×0+(-40)×0+0×1+1×0=0

w’=10×0+(-40)×0+0×0+1×1=1

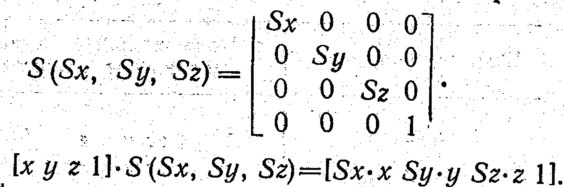
Однорідні координати точки після переміщення [30 -10 0 1]

Декартові координати [30/1 -10/1 0/1]= [30 -10 0]

**Матричний опис масштабування** в Sх разів по горизонталі, Sх разів по вертикалі та Sz разів до себе

Розтяг (збільшення) об’єкту Sх>1 Sу>1 Sz>1

Стиснення (зменшення) об’єкту 0<Sх<1 0<Sу<1 0<Sz<1



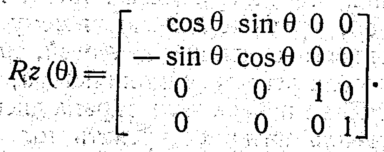
**Матричний опис повороту** на Тетаz, Тетаy, Тетаx градусів

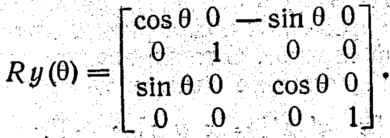
Проти год. стр. Тетаz>0 Тетаy>0 Тетаx>0

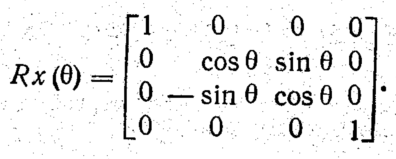
За год. стр. Тетаz<0 Тетаy<0 Тетаx<0

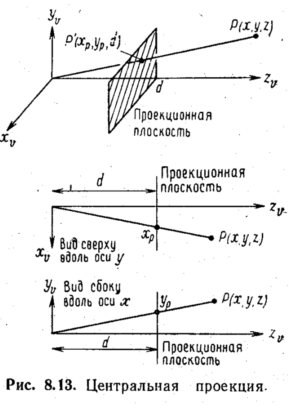
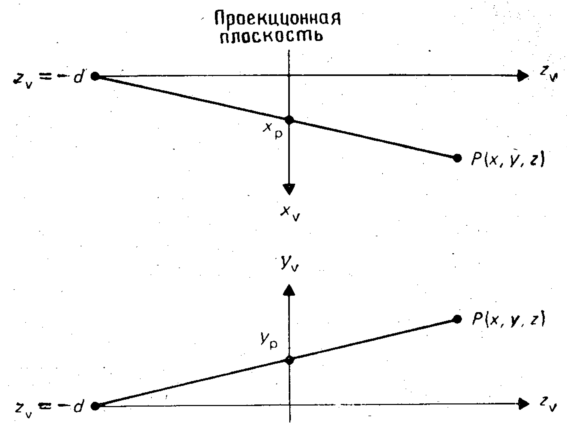
Композиція поворотів відносно трьох осей

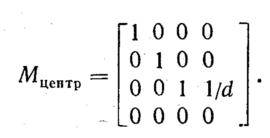
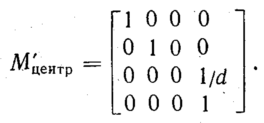
R = Rz ×Ry×Rx

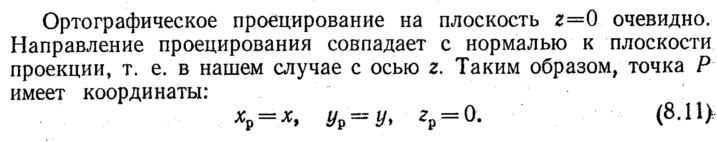


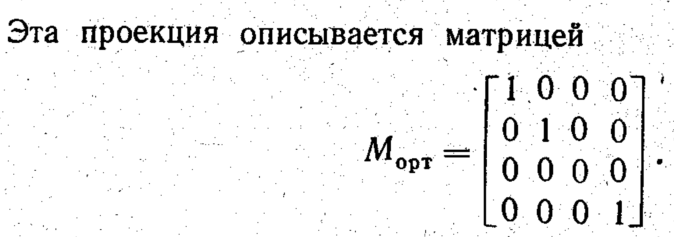








Використовується в курсовій роботі ортографічна проекція.

**% 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);**

Формуємо список координат точок зображення, по крокам афінні перетворення tformfwd і їх проекції, перетворення списку точок в зображення для кожної проекції

% 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;

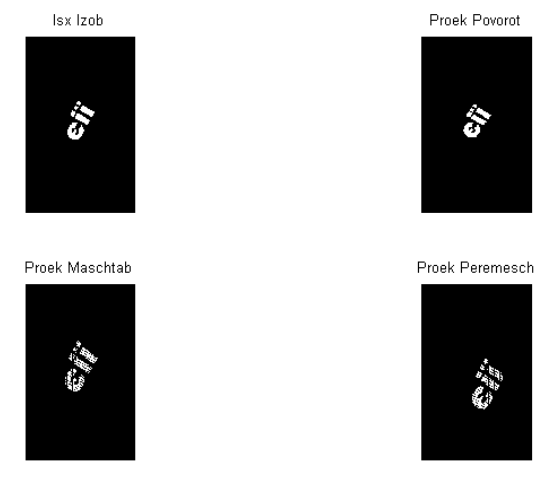
% вывод преобразованного изображения

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 Maschtab');

subplot(2,2,4); imshow(Im2DTransform3); title('Proek Peremesch');



зворотне перетворення, матриці прямого перетворення та tforminv

% 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;

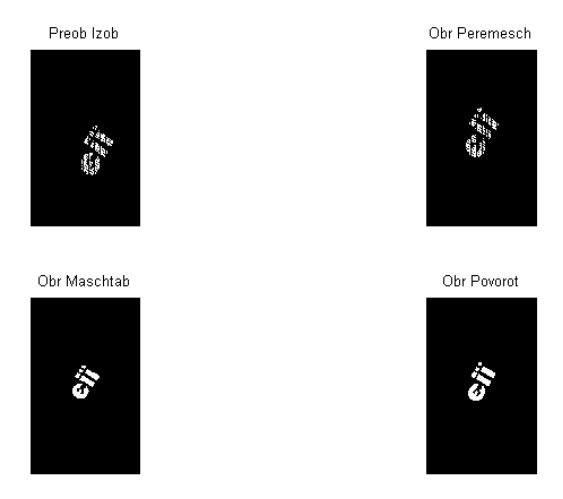
% вывод восстановленного изображения

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 Maschtab');

subplot(2,2,4); imshow(Im2DRestore1); title('Obr Povorot');



Виведення координат контрольних точок

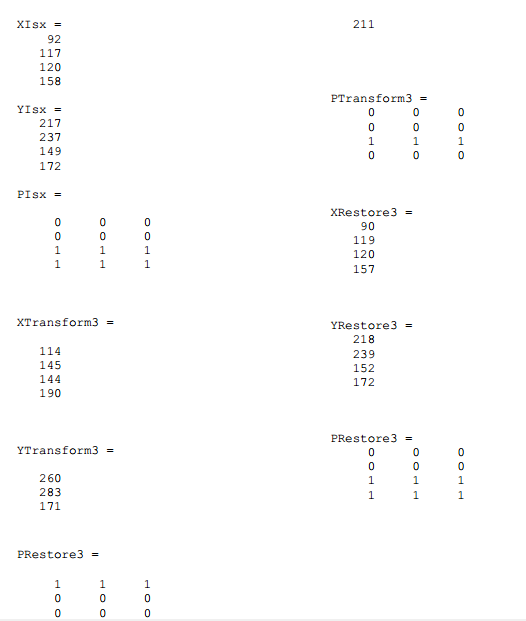
% получение координат точек

figure; [XIsx, YIsx, PIsx]=impixel(Im2D)

[XTransform3, YTransform3, PTransform3]=impixel(Im2DTransform3)

[XRestore3, YRestore3, PRestore3]=impixel(Im2DRestore1)





cpselect

