

Лабораторна робота № 7

ГЕОМЕТРИЧНІ І ПРОСТОРОВІ ПЕРЕТВОРЕННЯ ЗОБРАЖЕНЬ

7.1. Мета роботи

Вивчити основні методи геометричних і просторових перетворень та принципи побудови апаратних засобів, що реалізують ці методи.

Дослідити можливість використання геометричних і просторових перетворень для відеоінформації системи MatLab.

7.2. Основні теоретичні відомості

7.2.1. Геометричні перетворення

До найбільш поширених функцій геометричних перетворень відноситься кадрування зображень (`imcrop`), зміна розмірів (`imresize`) і поворот зображення (`imrotate`).

Суть кадрування полягає в тому, що функція `imcrop` дозволяє за допомогою миші в інтерактивному режимі вирізати частину зображення і помістити її в нове вікно перегляду рис.7.1.

```
L=imread('original.jpg');  
imshow(L);  
imcrop;
```

Функція `imcrop` повертає зображення, обмежене заданим прямокутником. Якщо вихідні параметри не задані, наприклад `imcrop(I)`, то результат відображається в новому графічному вікні.

Якщо не визначені вхідні параметри, наприклад `D = imcrop`, то функція `imcrop` оперує із зображенням в поточному графічному вікні.

Використовуючи функції `D = imcrop(S, rect)` і `Xd = imcrop(Xs, map, rect)`, можна явно визначити прямокутник, що обмежує, де `rect` - вектор з чотирьох елементів: `[xmin ymin wh]`, які задають положення лівого верхнього кута (`xmin ymin`) прямокутника, а також його ширину (`w`) і висоту (`h`) в просторових координатах.

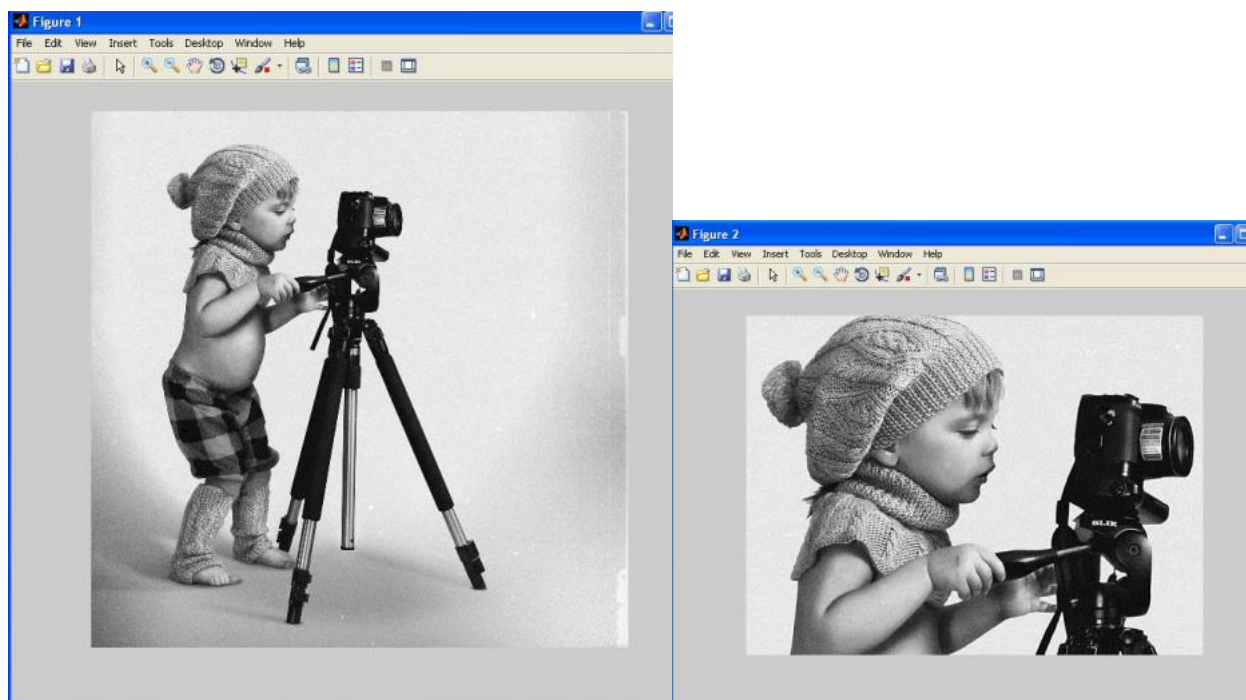


Рис. 7.1

Функція зміни розмірів зображення `imresize` дозволяє, використовуючи спеціальні методи інтерполяції, змінювати розмір будь-якого типу зображення рис.7.2.

```
L=imread('original.jpg');
J = imresize(L, [64 NaN]);
figure,imshow(L);
figure,imshow(J);
```

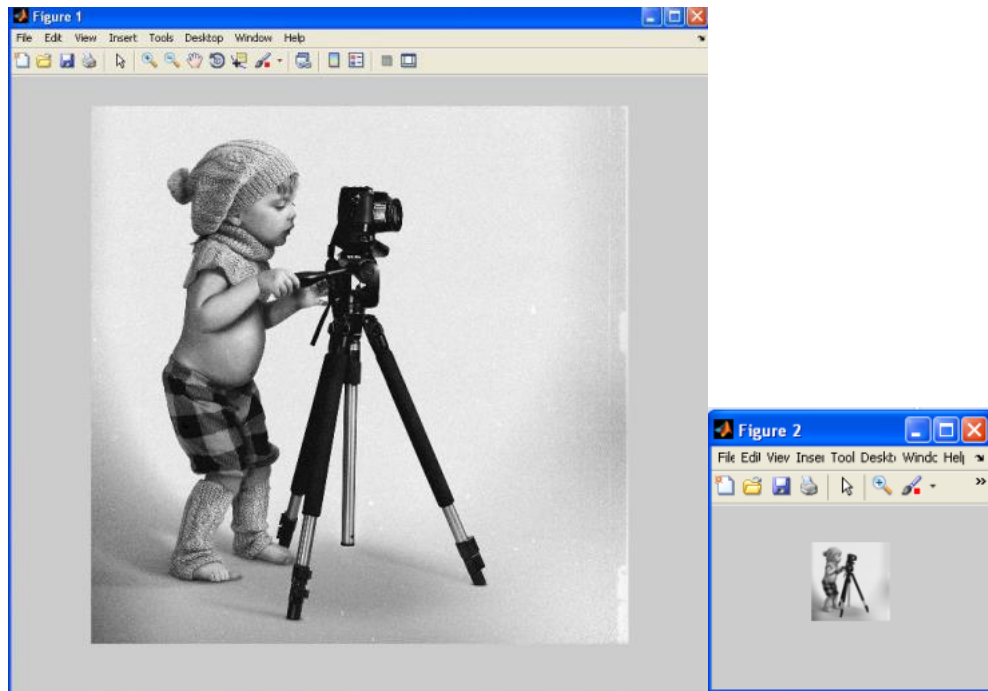


Рис. 7.2

У пакеті Image Processing Toolbox існує функція `imrotate`, яка здійснює поворот зображення на заданий кут (рис. 7.3).

```
L=imread('original.jpg');
L1=imrotate(L,30,'bicubic');
figure,imshow(L1)
figure,imshow(L);
```

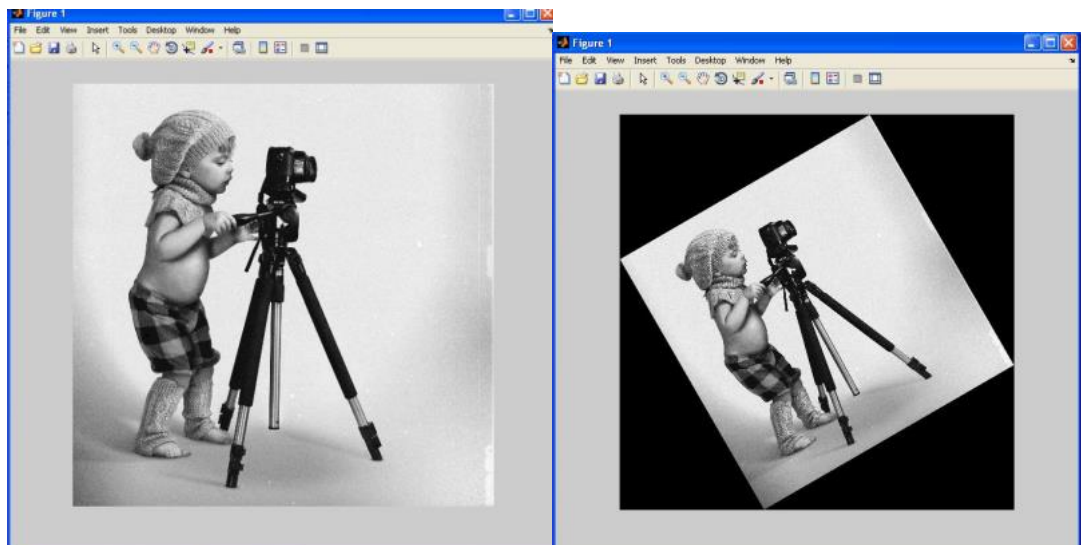


Рис. 7.3.

Таким чином, наведені вище функції дозволяють повертати, вирізати частини, масштабувати, тобто працювати з цілим масивом зображення.

7.2.2. Просторові перетворення

Просторові перетворення полягають у перетворенні з однієї системи координат в іншу. Кожне перетворення проводиться в різних координатних системах. При обчисленні геометричних перетворень вихідне зображення знаходиться в координатах (x, y) , а перетворене зображення - в координатах (u, v) .

Наведемо демонстраційні приклади, які включатимуть:

- Зображення 1: Застосування лінійних конформних перетворень.
- Зображення 2: Застосування афінних перетворень.
- Зображення 3: Застосування проекційних перетворень.
- Зображення 4: Застосування поліноміальних перетворень.
- Зображення 5: Застосування кусково-лінійних перетворень.
- Зображення 6: Застосування синусоїдальних перетворень.

Зображення 1: Застосування лінійних конформних перетворень

Лінійні конформні перетворення можуть включати поворот, масштабування і зсув. Контури та кути залишаються постійними. Паралельні лінії залишаються паралельними, прямі лінії залишаються прямими.

Для лінійних конформних перетворень: $[uv] = [xy \ 1] T$.

Параметр T представляє собою матрицю з розмірністю 3×2 , яка залежить від чотирьох параметрів.

```
% Чотири параметри
scale=1.2;          % коефіцієнт масштабування
angle=40*pi/180;    % кут повороту
tx=0;               % зсув по x
ty=0;               % зсув по y

sc=scale*cos(angle);
ss=scale*sin(angle);

T=[ sc -ss;
    ss  sc;
    tx  ty];
```

Далі лінійні конформні перетворення використовуються як підмножина афінних перетворень.

Після цього створюється структура TFORM (рис.7.4, b):

```
t_lc=maketform('affine', T);
I_linearconformal=imtransform(I, t_lc, 'FillValues', .3);
subplot(332)
imshow(I_linearconformal);
title('linear conformal')
```

Що стосується зсуву, то коли зміни в одному з напрямків tx або ty відмінні від нуля, то це не впливає на результуюче зображення. Щоб побачити координати, які відповідають перетворенням, включаючи зсув, потрібно провести наступні обчислення (рис.7.4, c):

```
[I_linearconformal, xdata, ydata]=imtransform(I, t_lc, 'FillValues', .3);
figure, imshow(xdata, ydata, I_linearconformal), axis on
```

Відзначимо, що параметри $xdata$ та $ydata$ відповідають зсуву. Для опису тієї частини зображення, яку потрібно розглянути (проаналізувати), у функції *imtransform* використовуються параметри 'XData' та 'YData'.

Зображення 2: Застосування афінних перетворень

При афінних перетвореннях розмірності x та y можна масштабувати або вирізати при зсуві. Паралельні лінії залишаються паралельними, прямі лінії залишаються прямими. Лінійні конформні перетворення є підмножиною афінних перетворень.

Вираз для афінних перетворень аналогічно виразу для лінійних конформних перетворень: $[uv] = [xy \ 1] T$. Параметр T представляє собою матрицю 3×2 з шістьма різними елементами (рис.7.4, d).

```
T=[1 0.1;  
   1 1;  
   0 0];  
t_aff=maketform('affine', T);  
I_affine=imtransform(I, t_aff, 'FillValues', .3);  
subplot(333)  
imshow(I_affine)  
title('affine')
```

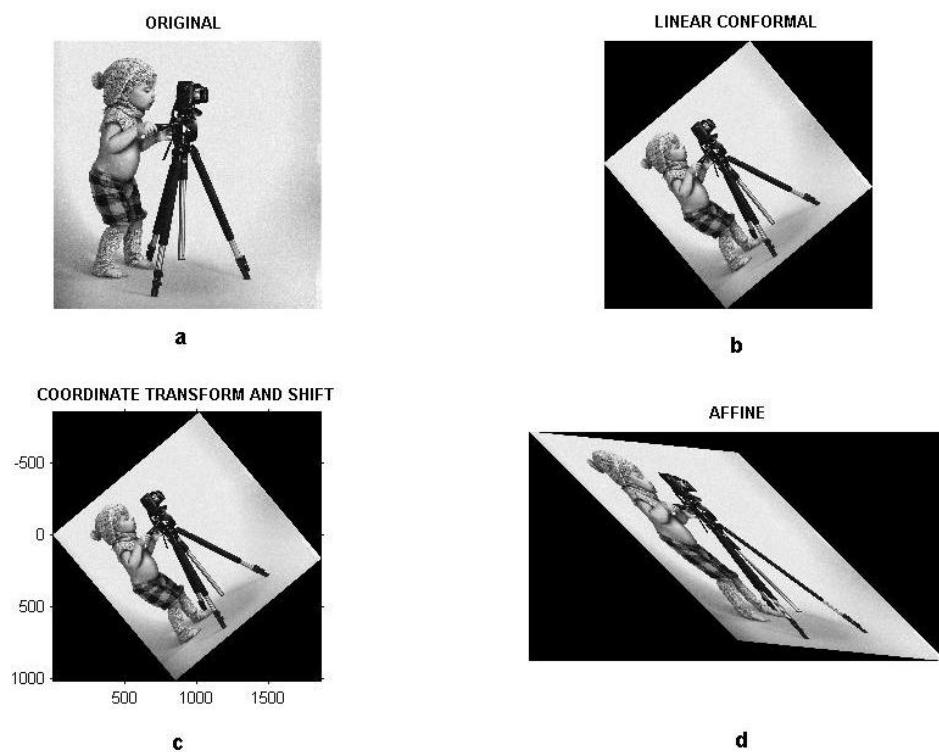


Рис.7.4

Зображення 3: Застосування проекційних перетворень

При проекційних перетвореннях чотирикутник залишається чотирикутником. Прямі лінії залишаються прямими лініями. Афінне перетворення є підмножиною проекційних перетворень (рис.7.5, a).

Для проекційних перетворень: $[u \ v \ w] = [x \ y \ 1] T$, де
 $u = u_p / w_p$
 $v = v_p / w_p$.

Параметр T представляє собою матрицю з розмірністю 3×3 з дев'ятьма різними елементами.

$$T = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

```

u = (Ax + By + C) / (Gx + Hy + 1)
v = (Dx + Ey + F) / (Gx + Hy + 1)

T=[1  0 0.008;
    1  1 0.01;
    0  0  1];
t_proj=maketform('projective', T);
I_projective=imtransform(I, t_proj, 'FillValues', .3);
subplot(334)
imshow(I_projective)
title('projective')

```

Зображення 4: Застосування поліноміальних перетворень

При поліноміальних перетвореннях поліноміальна функція від x і y визначає спосіб відображення (рис.7.5, b).

Для поліноміальних перетворень другого порядку:

$$[u \ v] = [1 \ x \ y \ x^2 \ y^2] T$$

Обидва параметра u та v є поліномами другого порядку щодо x та y . Кожен поліном другого порядку характеризується, в свою чергу, шістьма параметрами. При визначенні всіх коефіцієнтів розмірність T становить 6×2 .

```

xybase=reshape(randn(12, 1), 6, 2);
t_poly=cp2tform(xybase, xybase, 'polynomial', 2);
% Дванадцять елементів T.
T= [0 0;
    1 0;
    0 1;
    0.001 0;
    0.02 0;
    0.01 0];
t_poly.tdata=T;
I_polynomial=imtransform(I, t_poly, 'FillValues', .3);
subplot(335)
imshow(I_polynomial)
title('polynomial')

```

Зображення 5: Застосування кусково-лінійних перетворень

При кусково-лінійних перетвореннях, лінійні перетворення застосовуються окремо до різних частин зображення. У цьому прикладі права частина зображення є розтягнутою, а ліва частина не змінена (рис.7.5, c).

```

mid=round(size(I, 2)/2);
I_left=I(:, 1:imid);
stretch=1.5; % Коефіцієнт розтягнення
size_right=[size(I, 1) round(stretch*imid)];
I_right=I(:, imid+1:end);
I_right_stretched=imresize(I_right, size_right);
I_piecelinear=[I_left I_right_stretched];
subplot(336)
imshow(I_piecelinear)
title('piecewise linear')

```

Зображення 6: Застосування синусоїдальних перетворень

Зображення представлено на рис.7.5, d.

```
[nrows, ncols]=size(I);  
[xi, yi]=meshgrid(1:ncols, 1:nrows);  
a1=5; % амплітуда синусоїди.  
a2=3;  
u=xi+a1*sin(pi*xi/imid);  
v=yi-a2*sin(pi*yi/imid);  
tmap_B=cat(3, u, v);  
resamp=makeresampler('linear', 'fill');  
I_sinusoid=tformarray(I, [], resamp, [2 1], [1 2], [], tmap_B, .3);  
subplot(337)  
imshow(I_sinusoid)  
title('sinusoid')
```

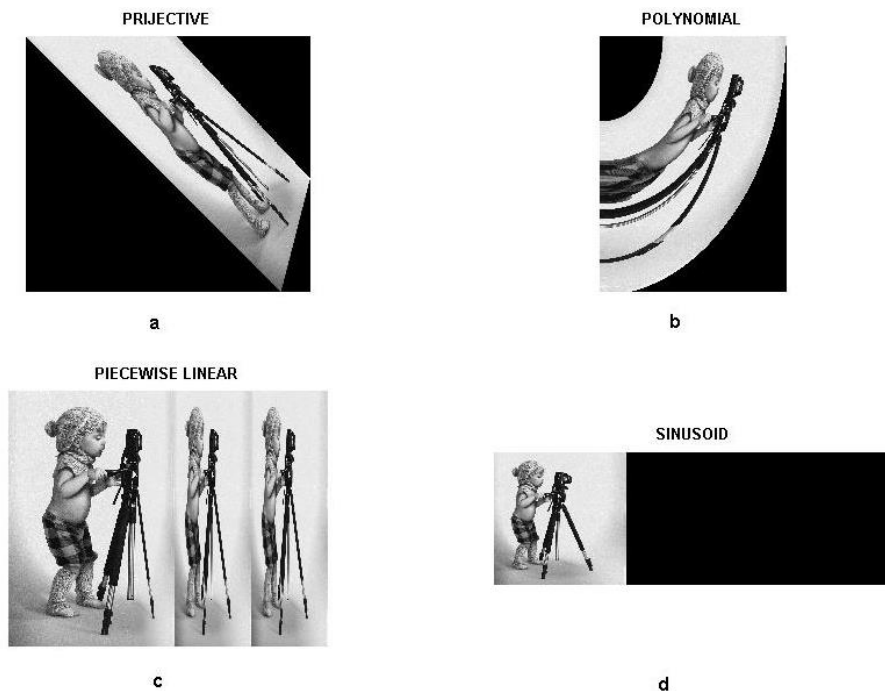


Рис. 7.5

7.3. Підготовка до роботи

Вивчити основні методи геометричних і просторових перетворень зображень та принципи побудови апаратних засобів, що реалізують ці методи (розділ 7.2 даного методичного посібника).

Виконати попередній аналіз початкових даних індивідуального завдання (табл. 7.1 і 7.2) і розробити програми для виконання цих завдань.

7.4. Виконання роботи

1. Геометричні перетворення відео зображень

1.1. Завантажити в оперативну пам'ять початкове кольорове відеозображення, задане викладачем. Виконати геометричні перетворення відео зображення.

1.2. Перетворити задане відеозображення використовуючи кадрівання, змінюючи межі прямокутника згідно свого варіанту (табл. 7.1).

1.3. Перетворити задане відеозображення використовуючи поворот зображення на заданий кут, згідно свого варіанту (табл. 7.1). 1.4. Вивчити основні можливості системи MatLab по геометричному перетворенню цифрових відеозображень.

1.5. Порівняти отримані результати, зробити висновки.

Приклад програми, що виконує геометричні перетворення відео зображення наведено у додатку 1.

наведено у додатку 1.

Таблиця 7.1

Варіант	Розмір зображення	Межі прямокутника	Кут
1	1200x1200	0 0 100 200	10
2	640x640	0 50 200 300	15
3	512x512	0 50 300 400	20
4	1200x1200	50 50 100 200	25
5	640x640	50 100 200 300	30
6	512x512	100 100 200 300	35
7	1200x1200	100 100 300 400	40
8	640x640	150 200 300 400	45
9	512x512	150 200 400 400	50
10	1200x1200	200 200 400 400	55
11	640x640	200 250 400 500	60
12	512x512	250 100 250 250	65
13	1200x1200	250 250 250 250	70
14	640x640	250 300 300 300	75
15	512x512	300 300 300 300	80
16	1200x1200	300 300 400 400	-10
17	640x640	300 400 400 400	-15
18	512x512	400 400 400 400	-20
19	1200x1200	400 450 450 450	-25
20	640x640	400 450 500 500	-30
21	512x512	500 500 400 400	-35
22	1200x1200	500 500 500 500	-40
23	640x640	500 600 600 600	-45
24	512x512	600 100 400 400	-50
25	1200x1200	600 200 400 400	-55
26	640x640	600 300 400 400	-60
27	512x512	700 400 500 500	-65
28	1200x1200	700 700 100 100	-70
29	640x640	700 100 500 500	-75
30	512x512	800 300 200 200	-80

2. Просторові перетворення відеозображень

2.1. Завантажити в оперативну пам'ять початкове кольорове відеозображення, задане викладачем. Виконати просторові перетворення відео зображення.

2.2. Перетворити задане відеозображення використовуючи спеціальні методи просторового перетворення, змінюючи дані згідно свого варіанту (табл.7.2).

3. Вивчити основні можливості системи MatLab по просторовому перетворенню цифрових відеозображень.

4. Порівняти отримані результати, зробити висновки.

Приклад програми, що виконує просторові перетворення відео зображення наведено у додатку 2.

Таблиця 7.2

Варіант	Scale, angle	stretch	a1, a2
1	$0,1/5*\pi/180$	0,1	1,1
2	$0,2/10*\pi/180$	0,2	2,1
3	$0,5/15*\pi/180$	0,3	3,2
4	$1,0/20*\pi/180$	0,4	5,2
5	$1,1/25*\pi/180$	0,5	5,3
6	$1,2/30*\pi/180$	0,6	6,2
7	$1,5/35*\pi/180$	0,7	6,3
8	$2,0/40*\pi/180$	0,8	6,4
9	$2,5/45*\pi/180$	0,9	7,2
10	$3/50*\pi/180$	1	7,3
11	$3,5/55*\pi/180$	1,1	7,4
12	$5/60*\pi/180$	1,2	7,5
13	$6/65*\pi/180$	1,3	8,3
14	$7/70*\pi/180$	1,4	8,4
15	$8/75*\pi/180$	1,5	8,5
16	$10/-5*\pi/180$	2,5	8,6
17	$12/-10*\pi/180$	3	9,2
18	$15/-15*\pi/180$	3,5	9,3
19	$20/-20*\pi/180$	4	9,4
20	$25/-25*\pi/180$	4,5	9,5
21	$30/-30*\pi/180$	5	9,6
22	$35/-35*\pi/180$	5,5	9,7
23	$40/-40*\pi/180$	6	9,8
24	$45/-45*\pi/180$	6,5	10,2
25	$50/-50*\pi/180$	7	10,3
26	$55/-55*\pi/180$	7,5	10,4
27	$60/-60*\pi/180$	8	10,5
28	$70/-65*\pi/180$	8,5	10,6
29	$80/-70*\pi/180$	9	10,7
30	$100/-75*\pi/180$	10	10,8

7.5. Зміст звіту

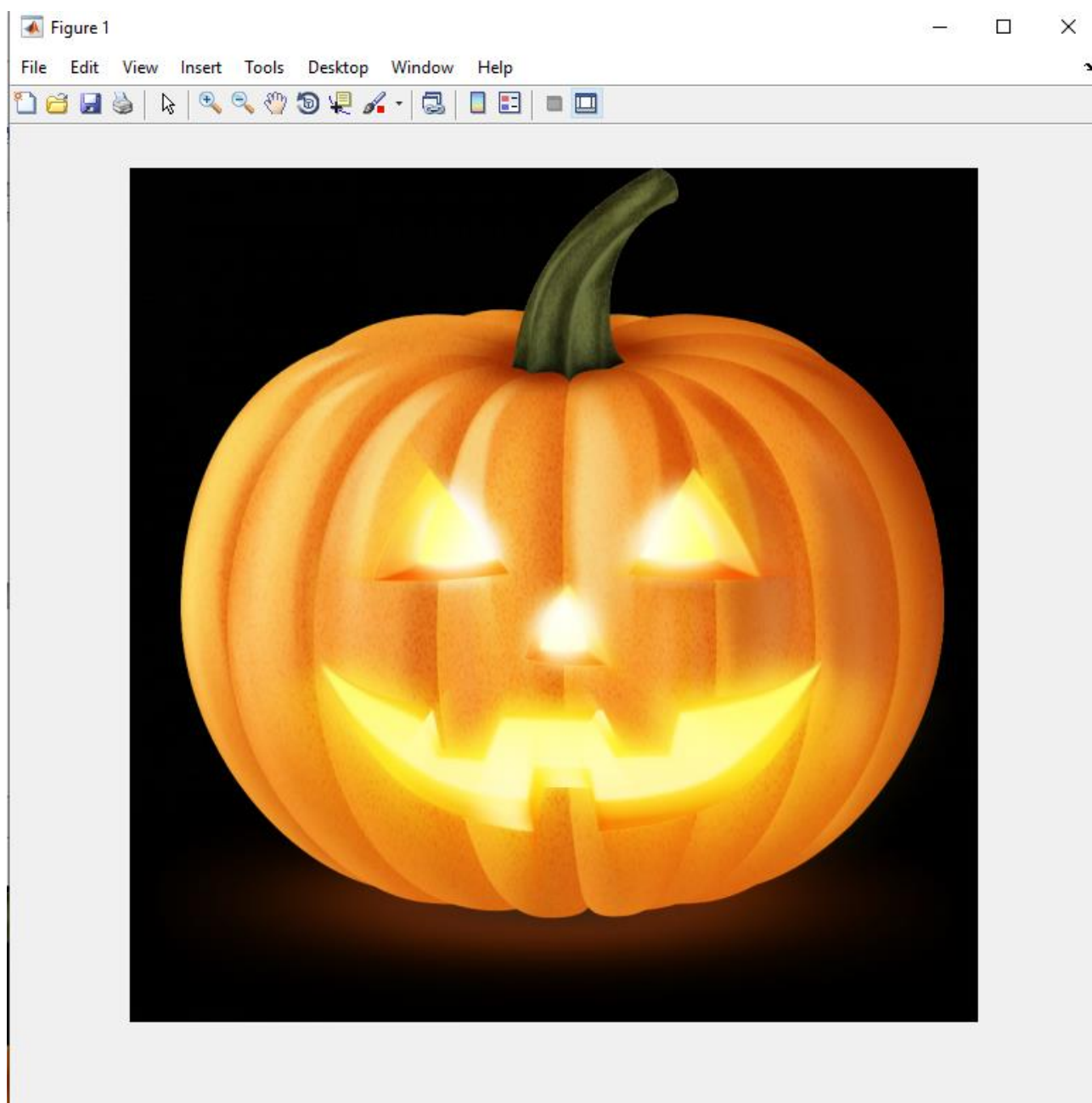
1. Найменування і мета роботи.
2. Програми для виконання дій з геометричними і просторовими перетвореннями цифрових відеозображень.
3. Результати досліджень на основі використання вказаної програми, відповідно до порядку виконання роботи.
4. Висновки по роботі.

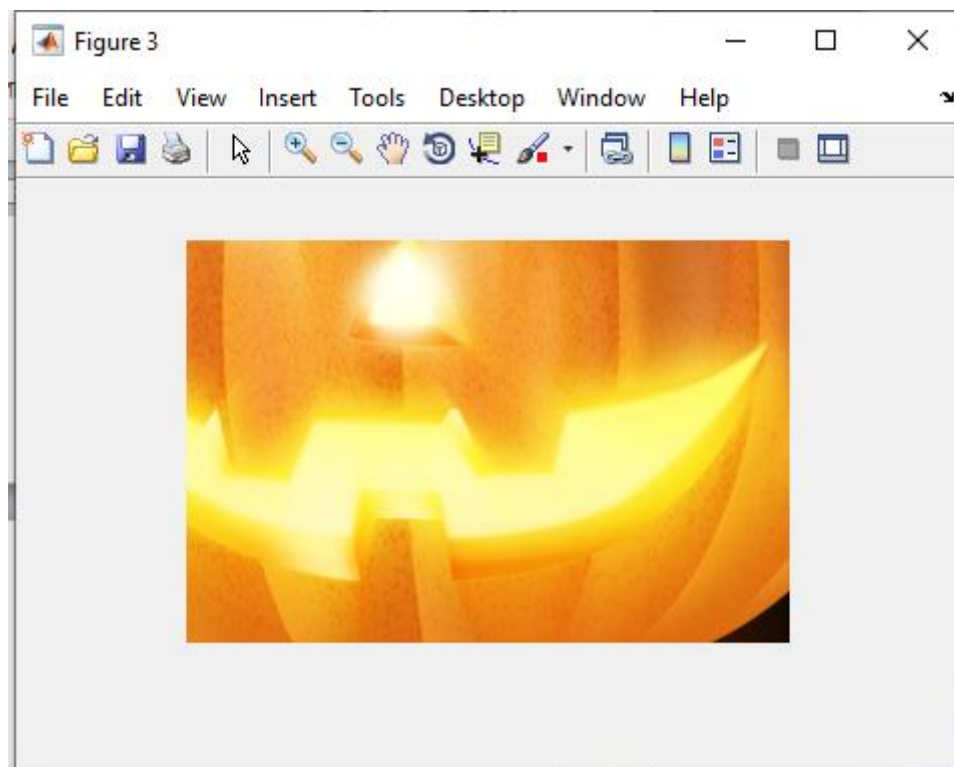
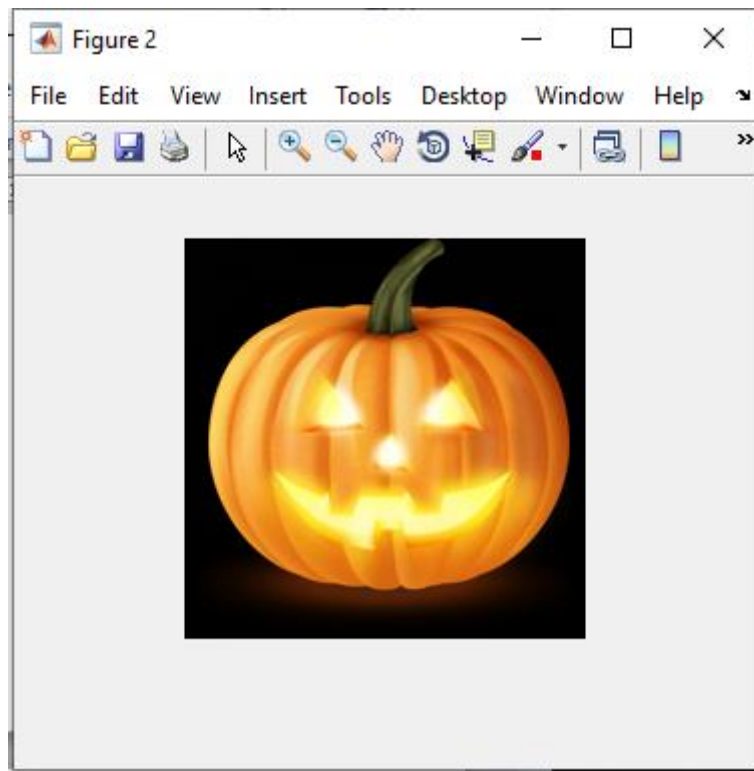
7.6. Контрольні запитання

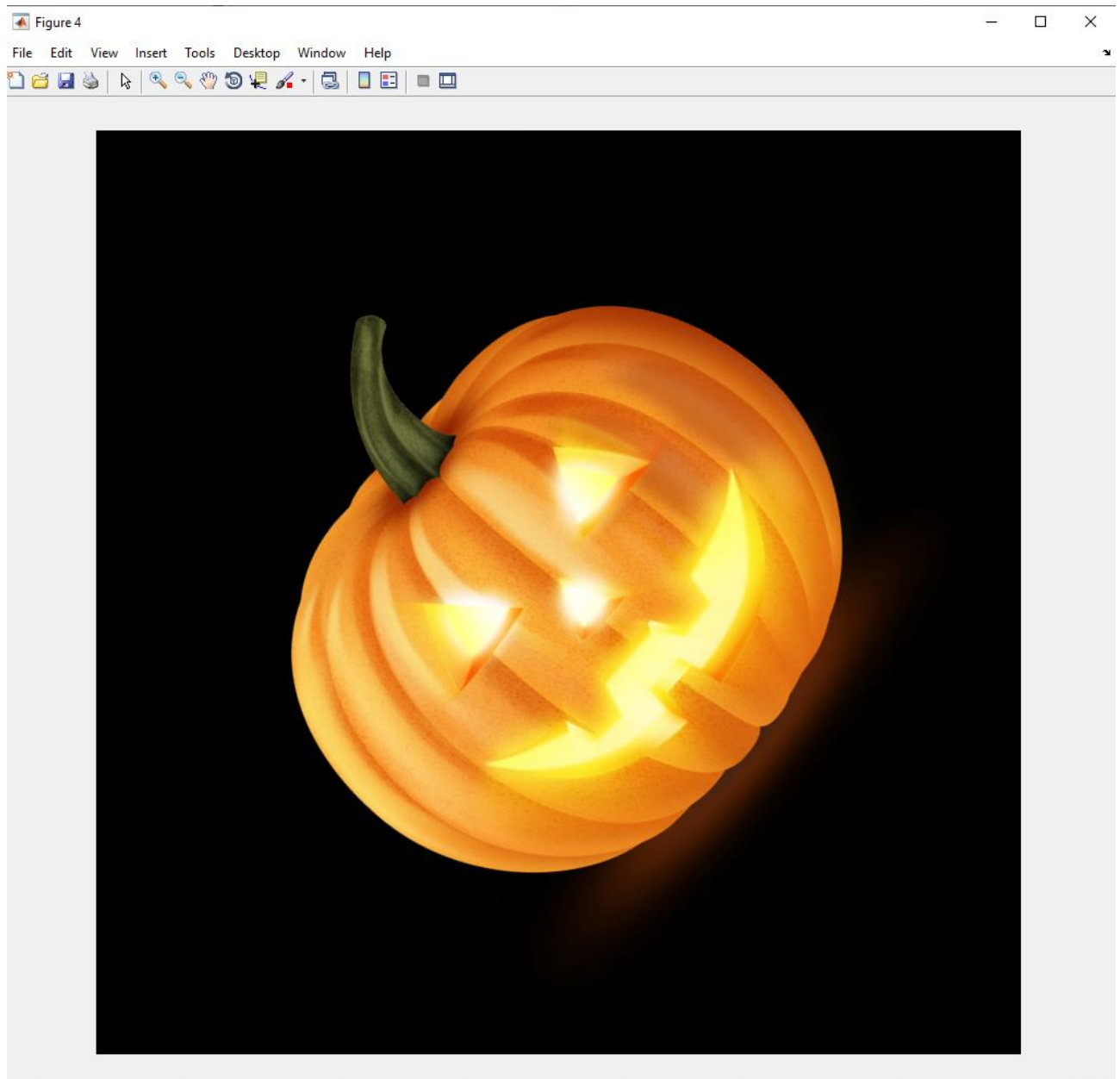
1. З яких основних функцій складається геометричне і просторове перетворення зображень?
2. Що являє собою геометричне і просторове перетворення?
3. Назвіть прикладні області, в яких може ефективно застосовуватися геометричне і просторове перетворення відеозображень.
4. Поясніть, як застосування геометричних і просторових параметрів впливає на характеристики отриманих відеозображень.
5. Назвіть функції геометричних і просторових перетворень, що підтримуються пакетом прикладних програм MatLab/Image Processing Toolbox.

Приклад програми, що виконує геометричні перетворення відео зображення

```
L=imread('halloween.png');  
J = imresize(L, [200 200]);  
H=imcrop(L,[200 300 300 200]);  
G=imrotate(L,410,'bicubic');  
figure,imshow(L);  
figure,imshow(J);  
figure,imshow(H);  
figure,imshow(G)
```







Додаток 2

Приклад програми, що виконує просторові перетворення відео зображення

```
I=imread('leaf.jpg');  
% Чотири параметри  
scale=1.2; % коефіцієнт масштабування  
angle=40*pi/180; % кут повороту  
tx=0; % зсув по x  
ty=0; % зсув по y  
sc=scale*cos(angle);  
ss=scale*sin(angle);  
T=[ sc -ss;  
    ss sc;  
    tx ty];  
subplot(331)  
imshow(I);  
title('Original image')
```

```

t_lc=maketform('affine', T);
I_linearconformal=imtransform(I, t_lc, 'FillValues', .3);
subplot(332)
imshow(I_linearconformal);
title('linear conformal')
[I_linearconformal, xdata, ydata]=imtransform(I, t_lc, 'FillValues', .3);
T=[1 0.1;
   1 1;
   0 0];
t_aff=maketform('affine', T);
I_affine=imtransform(I, t_aff, 'FillValues', .3);
subplot(333)
imshow(I_affine)
title('affine')
T=[1 0 0.008;
   1 1 0.01;
   0 0 1];
t_proj=maketform('projective', T);
I_projective=imtransform(I, t_proj, 'FillValues', .3);
subplot(334)
imshow(I_projective)
title('projective')
xybase=reshape(randn(12, 1), 6, 2);
t_poly=cp2tform(xybase, xybase, 'polynomial', 2);
% Дванадцять елементів T.
T= [0 0;
    1 0;
    0 1;
    0.001 0;
    0.02 0;
    0.01 0];
t_poly.tdata=T;
I_polynomial=imtransform(I, t_poly, 'FillValues', .3);
subplot(335)
imshow(I_polynomial)
title('polynomial')
imid=round(size(I, 2)/2);
I_left=I(:, 1:imid);
stretch=1.5; % Коефіцієнт розтягнення
size_right=[size(I, 1) round(stretch*imid)];
I_right=I(:, imid+1:end);
I_right_stretched=imresize(I_right, size_right);
I_piecelinear=[I_left I_right_stretched];
subplot(336)
imshow(I_piecelinear)
title('piecewise linear')
[nrows, ncols]=size(I);
[xi, yi]=meshgrid(1:ncols, 1:nrows);
a1=5; % амплітуда синусоїди.
a2=3;
u=xi+a1*sin(pi*xi/imid);
v=yi-a2*sin(pi*yi/imid);
tmap_B=cat(3, u, v);
resamp=makeresampler('linear', 'fill');
I_sinusoid=tformarray(I, [], resamp, [2 1], [1 2], [], tmap_B, .3);
subplot(337)
imshow(I_sinusoid)
title('sinusoid')

```

