Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Моделювання систем

Лабораторна робота №2

Виконав студент 3-го курсу

Групи ІПС-31

Поляков Артем Євгенович

Варіант №8

2022

**Постановка задачі**

Матрицю X будемо iнтерпретувати як двовимiрне вхiдне зображення,  
а матрицю Y – як вихiдне зображення. Потрiбно побудувати лiнiйний  
оператор перетворення вхiдного сигналу X у вихiдний сигнал Y на основi  
формули (3.9).

1. Вивчити означення псевдооберненої матрицi i її основнi властивостi.

2. Створити програму, яка за заданими двома зображеннями знаходить лiнiйний оператор переходу мiж цими зображеннями. Основою для  
програми є формула (3.9), де V – довiльна матриця (наприклад, нульова). Псевдообернену матрицю в (3.9) шукати двома методами: на основi  
формули Мура-Пенроуза (див. (3.3) або (3.4)) i на основi формули Гревiля. Правильнiсть знаходження псавдооберненої матрицi перевiрити за  
допомогою теореми 3.1 про характеристичну властивiсть псевдооберненої матрицi.

3. Вивести вихiдне зображення i образ вхiдного зображення при одержаному перетвореннi. Зробити порiвняння. Проаналiзувати одержаний  
результат.

4. Оформити в друкованiй формi звiт про виконання роботи, в якому  
викласти результати проведених обчислень.

**Код програми**

%#ok<\*NOPTS>

clear % видалення попередніх змінних

eps = 10^(-6);

X = imread("x1.bmp");

figure("Name", "X");

imshow(X);

X = double(X);

X = cat(1, X, ones(1,size(X,2)));

Y = imread("y8.bmp");

figure("Name", "Y");

imshow(Y);

Y = double(Y);

calculate(X, Inverse\_with\_Grevil(X), Y, "Grevil")

calculate(X, Inverse\_with\_Moor(X), Y, "Moor")

function calculate(X, X\_inv, Y, method\_name)

A = Y\*X\_inv + zeros(size(Y,1), size(X,1))\*Z(X\_inv,X);

figure("Name", method\_name);

imshow(uint8(A\*X));

end

% ------------------------------------------

function res = Z(A, A\_inv)

res = eye(size(A\_inv, 1)) - A\_inv\*A;

end

function res = R(A\_inv)

res = A\_inv \* (A\_inv');

end

function res = Grevil(A, a, A\_inv)

r = R(A\_inv);

z = Z(A,A\_inv);

if ((a'\*z\*a) < eps)

res = ((r\*a)\*a'\*A\_inv) / (1+a'\*r\*a);

res = A\_inv - res;

res = cat(2, res, (r\*a)/(1+a'\*r\*a));

else

res = ((z\*a)\*a'\*A\_inv) / (a'\*z\*a);

res = A\_inv - res;

res = cat(2, res, (z\*a)/(a'\*z\*a));

end

end

function res = Inverse\_with\_Grevil(A)

scalar = A(1,:) \* A(1,:)';

if (scalar == 0)

A\_inv = A(1,:);

else

A\_inv = A(1,:) / scalar;

end

A\_inv = A\_inv';

M = A(1,:);

for i = 2:size(A,1)

A\_inv = Grevil(M, A(i,:)', A\_inv);

M = cat(1, M, A(i,:));

end

res = A\_inv;

end

% ------------------------------------------

function res = Approximation(A, d)

res = A' \* (A\*A' + (d^2)\*eye(size(A,1)))^(-1);

end

function res = Inverse\_with\_Moor(A)

d = 10.0;

A\_curr = Approximation(A, d);

while 1

d = d/2;

A\_next = Approximation(A, d);

if (norm(A\_next-A\_curr) < eps)

break

end

A\_curr = A\_next;

end

res = A\_next;

end

% ------------------------------------------

**Постановка завдання**

|  |  |
| --- | --- |
|  |  |
| Input  *x1.bmp* | Output  *y8.bmp* |

**Результати виконання**

|  |  |
| --- | --- |
|  |  |
| Формула Гревіля | Означення Мура-Пенроуза |