

Лабораторна робота №2

Звіт

з дисципліни

“Моделювання систем”

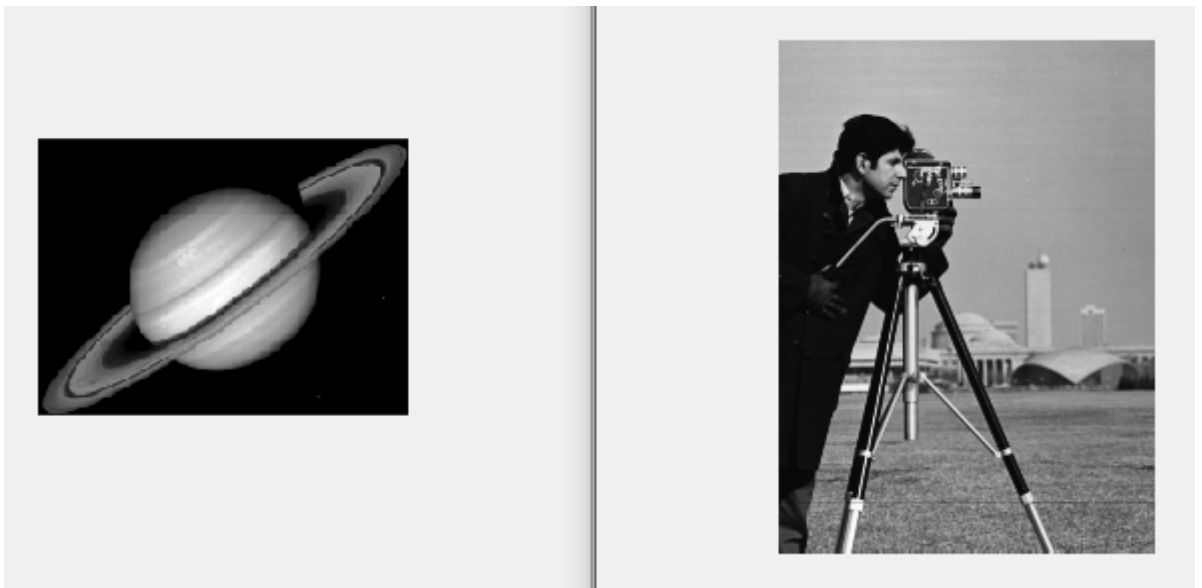
студента групи ІПС-31

Самойлича Євгенія

Варіант №9

1. Зчитуємо початкові вхідне та вихідне зображення та виводимо на екран.

```
X = double(imread("x1.bmp"));
Y = double(imread("y9.bmp"));
figure
imshow(uint8(X));
figure
imshow(uint8(Y));
```



2. Додаємо до матриці вхідного зображення рядок одиниць.

```
X = [X; ones(size(X, 2), 1)];
```

3. Визначаємо псевдообернену матрицю на основі формули Мура-Пенроуза.

```
delta = 1e-5;
eps = 1e-12;
A_g_inv_cur = A' / (A * A' + delta * eye(size(A, 1)));
A_g_inv_prev = 2 * A_g_inv_cur;
while (max(max((A_g_inv_cur - A_g_inv_prev) .^ 2)) > eps)
    A_g_inv_prev = A_g_inv_cur;
```

```

        delta = delta / 2.0;
        A_g_inv_cur = A' / (A * A' + delta * eye(size(A, 1)));
    end

```

4. Перевіряємо правильність псевдооберненої матриці через перевірку виконання характеристичних властивостей.

```

fprintf("%d\n", norm(X - temp));
temp = X_g_inv * X * X_g_inv;
fprintf("%d\n", norm(X_g_inv - temp));
temp = X * X_g_inv;
fprintf("%d\n", norm(temp - temp'));
temp = X_g_inv * X;
fprintf("%d\n", norm(temp - temp'));

```

```

8.452600e-07
1.546010e-06
1.626548e-10
4.502535e-14

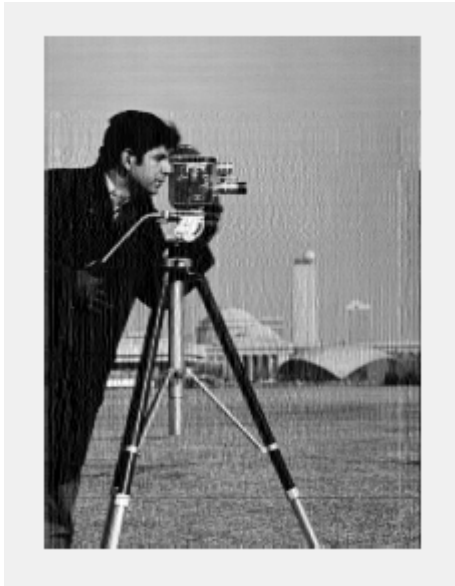
```

5. Знаходимо лінійний оператор переходу за допомогою знайденої псевдооберненої матриці, діємо оператором на вхідне зображення та виводимо результат на екран.

```

A = Y * X_g_inv + ones(size(Y, 1), size(X, 1)) * Z(X', X_g_inv)';
result = A * X;
figure
imshow(uint8(result));

```



6. Визначаємо псевдообернену матрицю на основі формули Гревіля.

```

a = A(1, :);
if ((a' * a) == 0)
    A_g_inv = zeros(size(A, 2), 1);
else
    A_g_inv = a / (a' * a);
end
for i = 2:size(A, 1)
    a = A(i, :);
    z = Z(A(1:(i - 1), :), A_g_inv);
    r = A_g_inv * A_g_inv';
    condition = a' * z * a;
    if (condition == 0)
        A_g_inv = [A_g_inv - (r * a * a' * A_g_inv) / (1 +
a' * r * a), (r * a) / (1 + a' * r * a)];
    else
        A_g_inv = [A_g_inv - (z * a * a' * A_g_inv) /
condition, (z * a) / condition];
    end
end
end

```

7. Перевіряємо правильність псевдооберненої матриці через перевірку виконання характеристичних властивостей.

```
fprintf("%d\n", norm(X - temp));  
temp = X_g_inv * X * X_g_inv;  
fprintf("%d\n", norm(X_g_inv - temp));  
temp = X * X_g_inv;  
fprintf("%d\n", norm(temp - temp'));  
temp = X_g_inv * X;  
fprintf("%d\n", norm(temp - temp'));
```

1.416382e-08

5.095575e-11

3.848193e-09

3.318322e-11

8. Знаходимо лінійний оператор переходу за допомогою знайденої псевдооберненої матриці, діємо оператором на вхідне зображення та виводимо результат на екран.

```
A = Y * X_g_inv + ones(size(Y, 1), size(X, 1)) * Z(X', X_g_inv)';  
result = A * X;  
figure  
imshow(uint8(result));
```

