**ЛАБОРАТОРНА РОБОТА №1**

**Градаційні перетворення**

Градаційні перетворення відносяться до найпростіших з всіх методів поліпшення зображень. Значення пікселів до і після обробки будуть позначатися символами r і s відповідно. Ці величини пов'язані виразом:

s=Т(r), (1)

де T є перетворенням, що відображає значення пікселя r в значення пікселя s. Оскільки ми маємо справу з дискретними (квантовим) величинами, то значення функції перетворення, як правило, зберігаються в одновимірному масиві, і відображення з r в s здійснюється по таблиці.

**1. Перетворення зображення в негатив**

Перетворення зображення в негатив з яскравостями в діапазоні [0, - 1] здійснюється з використанням негативного перетворення, показаного на Мал. 1, і описується виразом

s =L-1-r. (2)

Подібна зміна рівнів яскравості зображення створює еквівалент фотографічного негативу. Цей тип обробки особливо підходить для посилення білих або сірих деталей на тлі темних областей зображення, особливо коли темні області мають переважаючі розміри.

**2. Логарифмічне перетворення**

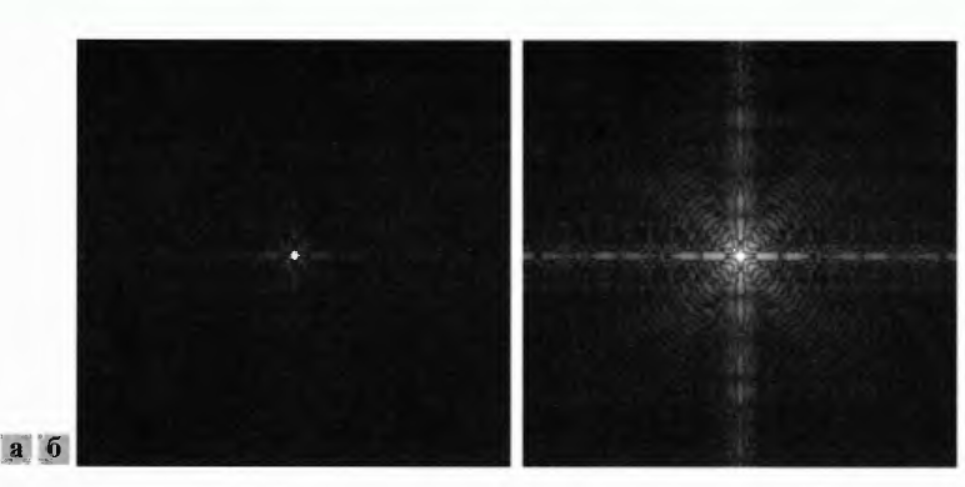
Загальний вид логарифмічного перетворення, показаного на Мал.1, виражається формулою:

s = c log(1 + r), (3)

де с - константа і передбачається, що r ≥ 0. Форма логарифмічної кривої на Мал.1 показує, що дане перетворення відображає вузький діапазон малих значень яскравостей на вихідному зображенні в більше широкий діапазон вихідних значень. Для великих значень вхідного сигналу вірно протилежне твердження. Цей тип перетворення використовується для розтягування діапазону значень темних пікселів на зображенні з одночасним стисненням діапазону значень яскравих пікселів. Навпаки, при використанні зворотного логарифмічного перетворення відбувається розтягнення діапазону яскравих пікселів і стиснення діапазону темних пікселів.

Будь-яка крива, має загальний вигляд близький до показаної на Мал. 1 логарифмічної функції, буде здійснювати таке розтягнення / стиснення діапазонів яскравості на зображенні. Класичним прикладом зображень , що мають великий динамічний діапазон, є спектр Фур'є. Його спектр, значень змінюються в діапазоні від 0 до 106 або більше. Якщо обробка подібної сукупності значень не представляє проблеми для комп'ютера, то система відтворення зображень зазвичай не здатна правильно відобразити настільки великий діапазон значень інтенсивності. Результуючий ефект такий, що при звичайному відтворенні спектра Фур'є значне число деталей втрачається.

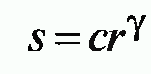
Як ілюстрацію логарифмічного перетворення, на Мал.1 наведено зображення спектра Фур'є, яке набуває значення в діапазоні від 0 до 1,5\*106.



Мал.1(а)- Спектр Фурьє. (б) Результат застосування логарифмічного перетворення за формулою з с=1.

**3. Степеневі перетворення**

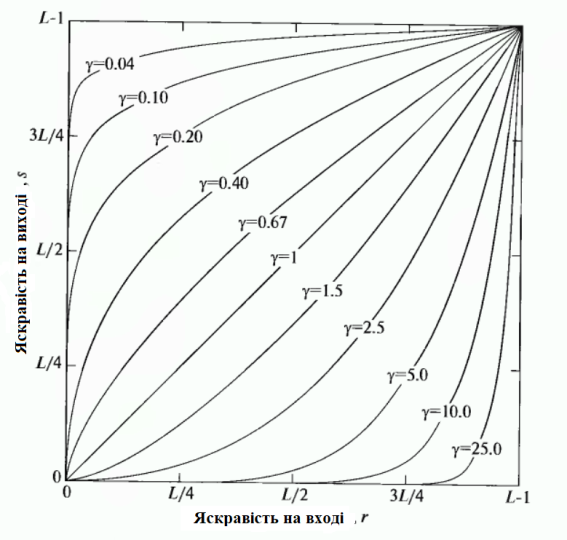
Степеневі перетворення мають вигляд:

 (4)

де с і у є позитивними константами. Іноді рівняння (4) записується в вигляді

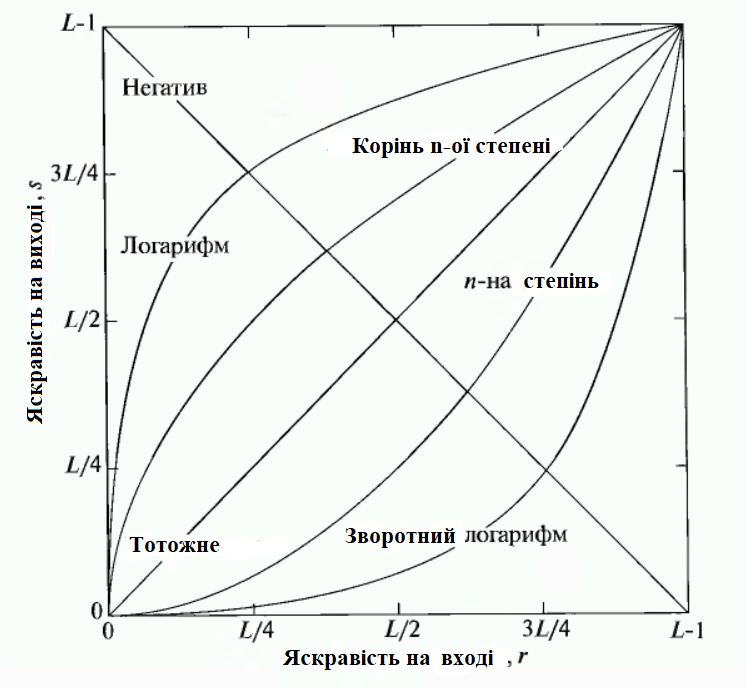
 (5)

для того, щоб ввести зміщення, тобто ненульовий вихід, коли на в ході нуль. Графіки залежностей s від r при різних значеннях  показані на Мал.2. Так ж як в випадку логарифмічного перетворення, криві степеневих залежностей при малих ** відображають вузький діапазон малих вхідних значень в широкий діапазон вихідних значень, при цьому для великих вхідних значень вірно зворотне твердження. Проте, на відміну від логарифмічних функцій, тут виникає ціле сімейство кривих можливого перетворення, отримуваних простою зміною параметра **. Як і слід було очікувати, на Мал.2 видно, що криві, отримані зі значеннями  > 1 дають прямо протилежний ефект по порівняно з тими, які отримані при  < 1. Відзначимо, що рівняння (4) зводиться до тотожному перетворенню при с = ** = 1.



Мал.3 Графіки для виразу (4), при різних значеннях ** (c=1 у всіх випадках).

На мал.4 для співставлення представлено всі основні види градаційних перетворень.



Мал. 4. Деякі основні градаційні перетворення, що використовуються використаних для покращення зображень.

**Завдання до роботи:**

1. Ознайомитись з основними способами градаційних перетворень.
2. Використовуючи представлений фрагмент коду перетворення зображення в негатив, запустити програму для градаційного перетворення.
3. Скласти аналогічну програму для градаційних перетворень описаних у пунктах 2-3
4. Проаналізувати для якого типу зображень ці перетворення є найбільш ефективні.

ФРАГМЕНТ КОДУ

using System;

using System.Collections.Generic;

using System.Text;

using System.Drawing;

using System.Drawing.Drawing2D;

namespace ImageProcessing

{

public class ImageHandler

{

private string \_bitmapPath;

private Bitmap \_currentBitmap;

private Bitmap \_bitmapbeforeProcessing;

public Bitmap CurrentBitmap

{

get

{

if (\_currentBitmap == null)

\_currentBitmap = new Bitmap(1, 1);

return \_currentBitmap;

}

set { \_currentBitmap = value; }

}

public string BitmapPath

{

get { return \_bitmapPath; }

set { \_bitmapPath = value; }

}

public void ResetBitmap()

{

if (\_currentBitmap != null && \_bitmapbeforeProcessing != null)

{

Bitmap temp = (Bitmap)\_currentBitmap.Clone();

\_currentBitmap = (Bitmap)\_bitmapbeforeProcessing.Clone();

\_bitmapbeforeProcessing = (Bitmap)temp.Clone();

}

}

public void SaveBitmap(string saveFilePath)

{

\_bitmapPath = saveFilePath;

if (System.IO.File.Exists(saveFilePath))

System.IO.File.Delete(saveFilePath);

\_currentBitmap.Save(saveFilePath);

}

public void RestorePrevious()

{

\_bitmapbeforeProcessing = \_currentBitmap;

}

public void SetInvert()

{

Bitmap temp = (Bitmap)\_currentBitmap;

Bitmap bmap = (Bitmap)temp.Clone();

Color c;

for (int i = 0; i < bmap.Width; i++)

{

for (int j = 0; j < bmap.Height; j++)

{

c = bmap.GetPixel(i, j);

bmap.SetPixel(i, j, Color.FromArgb(255 - c.R, 255 - c.G, 255 - c.B));

}

}

\_currentBitmap = (Bitmap)bmap.Clone();L

}

}

}