Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
про виконання лабораторної роботи №5
Основи цифрового представлення зображень та їх обробки

Виконала
студентка групи ФеС-21
Шавало А.А.

Перевірив
Вдовиченко В. М.

Львів 2024 р.

Мета роботи
Ознайомитись із основними методами представлення та обробки зображень.

Теоретичні відомості :

Зображення як багатовимірні сигнали

Найбільш розповсюдженим прикладом багатовимірного сигналу є зображення. В такому сигналі значення сигналу є функцією двох координат в просторі $I_1(x, y)$. Наприклад, якщо мова іде про сірошкальне зображення, то значення сигналу є яскравістю точки в просторі. Як правило, яскравість – число від 0 до 1, де 0 відповідає точці чорного кольору, а 1 – відповідає точці білого кольору (максимальна яскравість). Проміжні значення між 0 та 1 говорять про насиченість кольору: 0.5 відповідає сірому кольору, 0.2 – темно-сірому, 0.8 – світло-сірому і т.д.

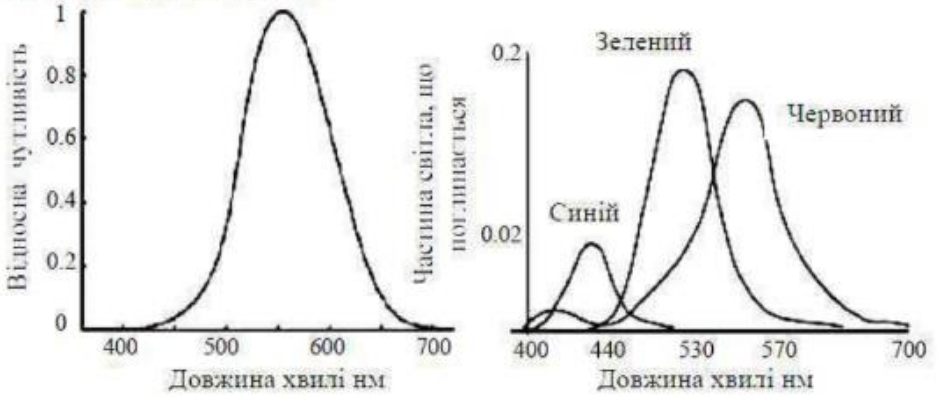
Всі методи аналізу та обробки одновимірних сигналів, які розглядались до цього, можуть бути застосовані до багатовимірних сигналів, із врахуванням відповідної розмірності.

Представлення зображень

З точки зору теорії сигналів, зображення – це функція двох змінних $I_1(x, y)$, де x та y – координати на площині. Значення $I_1(x, y)$ в довільній точці площини називається яскравістю (інтенсивністю) зображення. Як і у випадку одновимірних сигналів, $I_1(x, y)$ називається неперервним зображенням в разі, коли координати x та y можуть приймати довільні значення, а дискретним – в разі, коли x та y визначені лише для деякої множини значень.

Модель кольорових зображень

Поняття кольору базується на сприйнятті очима людини електромагнітних хвиль в певному діапазоні частот. Сприймається денне світло має довжини хвиль λ від 400 нм (фіолетовий) до 780 нм (червоний). Описом світлового потоку може служити його спектральна функція $I(\lambda)$. Світло називається монохроматичним, якщо його спектр має тільки одну певну довжину хвилі.



Колірна система CIE XYZ.

Міжнародний стандарт представлення кольору CIE (CIE - Commission Internationale de l'Eclairage) був прийнятий в 1931 році Міжнародною комісією з освітлення. В ньому визначаються три базисні функції ρX (λ), ρY (λ), ρZ (λ), що залежать від довжини хвилі , лінійні комбінації яких з невід'ємними коефіцієнтами (X, Y і Z) дозволяють отримати всі видимі людиною кольори. Для людини дана система кольорів є первинною, а всі решта систем кольору лише наближують її.

У тривимірному просторі колірна система CIE утворює конус в першому квадраті і застосовується для високоякісного відображення кольорових зображень.

CMYK

CMYK (скорочено від англ. Cyan, Magenta, Yellow, BlacK color) — субтрактивна колірна модель, використовується у поліграфії, перш за все при багатофарбовому (повноколірному) друці. Вона застосовується у друкарських машинах і кольорових принтерах. Кольори візуально не ідентичні із загальноприйнятими назвами кольорів: маджента - це лише один з пурпурових відтінків; жовтий і блакитний — абсолютно певні відтінки.

Оскільки модель CMYK застосовують в основному в поліграфії при кольоровому друці, а папір і інші друкарські матеріали є поверхнями, що відображають світло, зручніше рахувати яку кількість світла (і кольори) відбилося від тієї або іншої поверхні, ніж скільки поглинається.

Модель CMYK забезпечує менше колірне охоплення, ніж модель RGB, проте легше реалізується за допомогою фарб.

Колірна модель RGB

Колірна модель RGB (Red, Green, Blue - червоний, зелений, блакитний) в машинній графіці в даний час є найпоширенішою. У цій моделі спектральна функція представляється як сума кривих чутливості для кожного типу колб з невід'ємними ваговими коефіцієнтами (з нормування від 0 до 1), які так і позначаються - R, G і B. Модель характеризується властивістю адитивності для отримання нових кольорів. Наприклад, кодування спектральних функцій:

- ◊ - чорного кольору: $f_{black} = 0$, $(R,G,B) = (0,0,0)$;
- ◊ - фіолетового кольору $f_{violet} = f_{red} + f_{blue}$, $(R,G,B) = (1,0,1)$;
- ◊ - білого кольору $f_{white} = f_{red} + f_{green} + f_{blue}$, $(R,G,B) = (1,1,1)$.

Кількість градацій кожного каналу залежить від розрядності бітового значення RGB. Зазвичай використовують 24-бітну модель, у котрій визначається по 8 біт на кожен канал, і тому кількість градацій дорівнює 256, що дозволяє закодувати $256^3 = 16\,777\,216$ кольорів.

Переваги моделі

- ◊ Апаратна близькість із монітором, сканером, проектором, іншими пристроями;
- ◊ Велика кольорова гама, близька до можливостей людського зору;
- ◊ Доступність багатьох функцій обробки зображення (фільтрів) у програмах растрової графіки;
- ◊ Невеликий (порівняно до моделі CMYK) обсяг, проте ширший спектр кольорів.

В силу особливостей сприйняття світла рецепторами не всі кольори, видимі людиною, представимо в цій моделі. Однак частина відтворених кольорів значно більше, ніж кольори не представлені у моделі.

YUV

YUV — кольорова модель, в якій колір представляється як 3 компоненти — яскравість (Y) і дві сигнали різниці кольорів (U і V).

Перехід у RGB і назад здійснюється за наступними формулами:

$$\begin{aligned} R &= Y + 1.13983\ (V - 128); \\ G &= Y - 0.39465\ (U - 128) - 0.58060\ (V - 128); \\ B &= Y + 2.03211\ (U - 128); \end{aligned} \quad (1)$$

$$\begin{aligned} Y &= 0.299\ R + 0.587\ G + 0.114\ B; \\ U &= -0.14713\ R - 0.28886\ G + 0.436\ B + 128; \\ V &= 0.615\ R - 0.51499\ G - 0.10001\ B + 128; \end{aligned} \quad (2)$$

Де R, G, B — відповідно інтенсивності кольорів червоного, зеленого та синього, Y — яскравість, U і V — різницеві сигнали.

Модель широко застосовується в телебаченні та зберіганні/обробці відеоданих. Яскравість зберігає зображення у відтінках сірого, а компоненти U та V, містять інформацію для відновлення кольору. У колірному просторі YUV найбільш важливою компонентою є яскравість Y. Зазвичай вона передається із найбільшою роздільною здатністю, а компоненти U та V можуть передаватись і зберігатись із значно меншою роздільною здатністю і глибиною кольору.

Дискретизація зображень

Заміну неперервного зображення дискретним можна виконати різними способами. Можна, наприклад, вибрати яку-небудь систему ортогональних функцій і, вирахувавши коефіцієнти представлення зображення за цією схемою (за цим базисом), замінити ним зображення. Різноманітність базисів уможливлює утворення різних дискретних представлень неперервного зображення. Однак найуживанішою є періодична дискретизація, а саме дискретизація з прямокутним растром. Такий спосіб дискретизації може розглядатись як один із варіантів використання ортогонального базису, у якому використовуються як базисні елементи зсунуті δ –функції.

У найпростішому випадку чорно-білих зображень ми маємо двовимірний масив $s_a(x, y)$. Для кольорових зображень в моделі RGB, враховуючи властивість адитивності при додаванні кольорів, кожен шар R, G і B також може розглядатися і оброблятися, як двовимірний масив, з подальшим сумуванням результатів. Із способів узагальнення одновимірної періодичної дискретизації на двовимірний випадок найбільш простим є періодична дискретизація в прямокутних координатах:

$$s(n, m) = s_a(n \Delta x, m \Delta y), \quad (3)$$

де Δx и Δy - горизонтальний і вертикальний інтервали дискретизації двовимірного неперервного сигналу $s_a(x, y)$ з неперервними координатами x і y. Нижче значення Δx і Δy, як і в одновимірному випадку, приймаються рівними 1.

Дискретизація двовимірного сигналу також призводить до періодизації його спектра і навпаки. Зберігається і умова інформаційної рівноцінності координатного і частотного уявлень дискретного сигналу при рівній кількості точок дискретизації в головних діапазонах сигналу.

Сигнал з необмеженим спектром також може бути дискретизований, проте в цьому випадку має місце накладення спектрів в суміжних періодах, при цьому високі частоти, більших за частоту Найквіста, будуть "маскуватися", як і в одновимірному випадку, під низькі частоти головного періоду. Особливо наочно цей ефект можна спостерігати на різких контрастних змінах яскравості.

Завдання :

Хід роботи

1. Завантажити у графічний редактор задане кольорове зображення.

2. Виділити для кольорового зображення його компоненти у системі кольорів RGB та YUV.

3. Зменшити роздільну здатність для компонентів зображення удвічі.

4. Завантажити у графічний редактор задане сірошкальне зображення.

5. Провести еквіалізацію його гістограми для підвищення контрасту, а також спробувати ввести гамма-корекцію зображення із показником ступеня 0.5 та 2. Пояснити отримані результати.

6. Добавити до початкового сірошкального зображення шум типу чорні та білі точки ("сіль та перець") так щоб замінити коло 5 % точок на точки із шумом.

---

7. Провести еквіалізацію такого зображення. Пояснити відмінність отриманого зображення від зображення отриманого раніше.

8. Провести фільтрацію зашумленного зображення за допомогою згладжуючого лінійного фільтра 3x3 та за допомогою медіанного фільтра. Пояснити відмінність.

9. Виявити границі областей зображення на основі лінійного та нелінійного фільтра (за вибором).

Лістинг програми :

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def salt_and_pepper_noise(image, prob):
    output = np.copy(image)
    white = np.random.rand(image.shape[0],
image.shape[1]) < prob
    black = np.random.rand(image.shape[0],
image.shape[1]) < prob
    output[white] = 255
    output[black] = 0
```

```
    return output

gray_image = cv2.imread('img.png',
cv2.IMREAD_GRAYSCALE)

equalized_image = cv2.equalizeHist(gray_image)

gamma1 = 0.5
gamma2 = 2.0
gamma_corrected1 = np.array(255 * (gray_image / 255)
** gamma1, dtype='uint8')
gamma_corrected2 = np.array(255 * (gray_image / 255)
** gamma2, dtype='uint8')

noisy_image = salt_and_pepper_noise(gray_image, 0.05)

equalized_noisy_image = cv2.equalizeHist(noisy_image)

blurred_image = cv2.blur(noisy_image, (3, 3))
median_filtered = cv2.medianBlur(noisy_image, 3)

sobelx = cv2.Sobel(gray_image, cv2.CV_64F, 1, 0,
ksize=3)
sobely = cv2.Sobel(gray_image, cv2.CV_64F, 0, 1,
ksize=3)
sobel_combined = cv2.magnitude(sobelx, sobely)

edges = cv2.Canny(gray_image, 100, 200)

color_image = cv2.imread('img.png')

img_rgb = cv2.cvtColor(color_image,
cv2.COLOR_BGR2RGB)
img_yuv = cv2.cvtColor(color_image,
cv2.COLOR_BGR2YUV)
```

```
Y, U, V = cv2.split(img_yuv)
R, G, B = cv2.split(img_rgb)

fig, axes = plt.subplots(16, 1, figsize=(8, 32))
axes = axes.ravel()

titles = ['Original RGB', 'Equalized Image', 'Gamma
Corrected 0.5', 'Gamma Corrected 2.0',
          'Red Channel', 'Green Channel', 'Blue
Channel', 'Sobel Edges',
          'Y (Luminance)', 'U (Chrominance)', 'V
(Chrominance)', 'Noisy Image',
          'Equalized Noisy Image', 'Blurred Image
(3x3)', 'Median Filtered Image', 'Canny Edges']

images = [img_rgb, equalized_image, gamma_corrected1,
gamma_corrected2,
          R, G, B, sobel_combined,
          Y, U, V, noisy_image,
          equalized_noisy_image, blurred_image,
median_filtered, edges]

for i in range(len(images)):
    cmap = 'gray' if len(images[i].shape) == 2 else
None
    axes[i].imshow(images[i], cmap=cmap)
    axes[i].set_title(titles[i], fontsize=14)
    axes[i].axis('off')

plt.tight_layout()
plt.show()
```

Результат виконання :

Original RGB

Gamma Corrected 2.0

Equalized Image

Red Channel

Gamma Corrected 0.5

Green Channel

Blue Channel

U (Chrominance)

Sobel Edges

V (Chrominance)

Y (Luminance)

Noisy Image

Equalized Noisy Image

Blurred Image (3x3)

Median Filtered Image

Canny Edges