

ФАКУЛЬТЕТ ГІСУТ

Кафедра геоінформатики і фотограмметрії  
193 «Геодезія та землеустрій»  
Освітній рівень «бакалавр»

Дипломна робота на тему:

Розроблення функції Python для автоматизації процесу оцифрування  
горизонталей для побудови цифрових моделей рельєфу

Виконав студент групи ГСТ - 41  
Карнарук К. А.  
Керівник:  
Максимова Ю. С.

Київ 2022 р.

# Мета та завдання диплому

**Мета** - створення функції на мові python для автоматизації оцифрування карт та створення ЦМР.

## **Завдання:**

1. Огляд та характеристика видів, способів побудови, інструментів для створення ЦМР.
2. Огляд синтаксису мови python та бібліотек мови python для обробки зображень.
3. Побудова технологічної схеми автоматизації процесу оцифрування горизонталей для побудови цифрових моделей рельєфу з використанням мови Python.
4. Дослідна реалізація функції Python для автоматизації процесу оцифрування горизонталей для побудови цифрових моделей рельєфу та її апробації

# Поняття цифрової моделі рельєфу

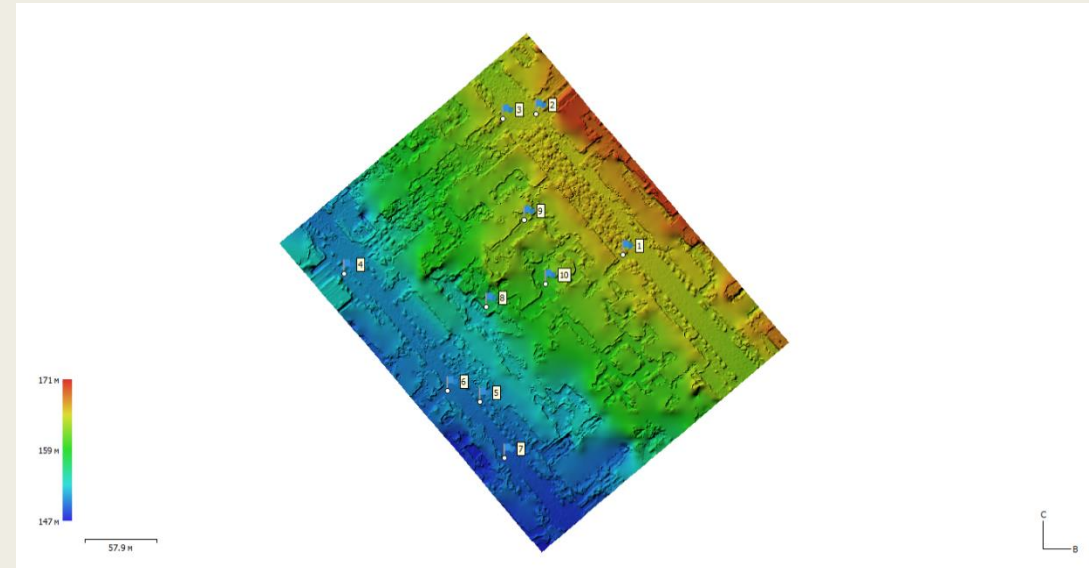
**Цифрова модель рельєфу** (анг. Digital Terrain Model) - це математичне представлення поверхні землі, найчастіше у вигляді звичайної сітки, в якій кожному пікселю призначається унікальне значення висоти.

Головною метою створення ЦМР є при мінімальній кількості точок моделі забезпечення максимально можливої точності відображення рельєфу місцевості.

# Приклади застосування ЦМР

ЦМР можуть застосовуватися для:

- Аналізу та прогнозування повенів – ЦМР допомагає більш точно прогнозувати повені у долинах річок, та аналізувати як боротися з їх наслідками.
- Класифікації ґрунтового покриву – за допомогою ЦМР зручно досліджувати чинник які впливають на формування рельєфу.
- Космічних досліджень – завдяки ЦМР науковці аналізують поверхні веземних тіл.
- Візуалізації рельєфу місцевості – за допомогою 3D моделі краще проілюструвати особливості місцевості ніж на топографічній карті.

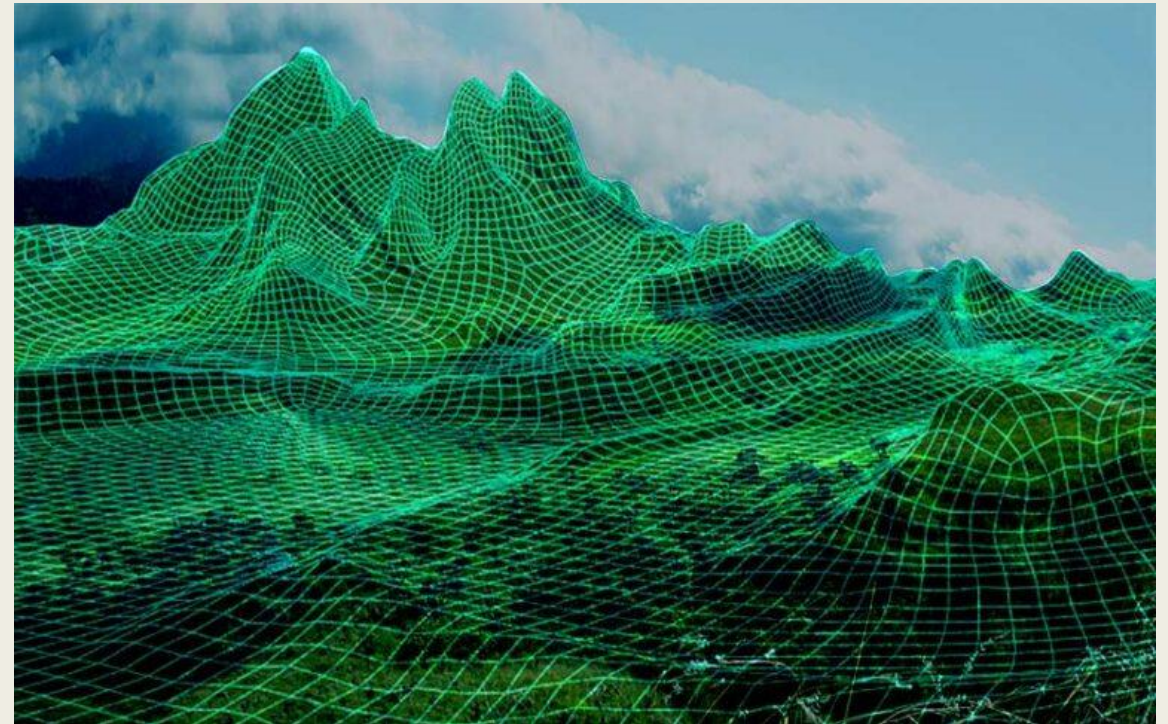


# Шляхи створення ЦМР

Методи	Переваги	Недоліки
Фотограмметричний метод	Швидкість обробки великих площ при достатньо потужному комп'ютері	Залежність точності побудови ЦМР від можливостей комп'ютера
Метод Лазерного сканування	Автоматизований процес зйомки місцевості, швидкість зйомки	Залежність можливостей сканера від його моделей
Метод геодезичних вишукувань	Достатньо велика точність вимірів	Повільний процес зйомки місцевості (у випадку тахеометричного знімання)
Картометричний методот	Відсутність польових робіт	Дрібномасштабні карти практично не придатні для створення ЦМР

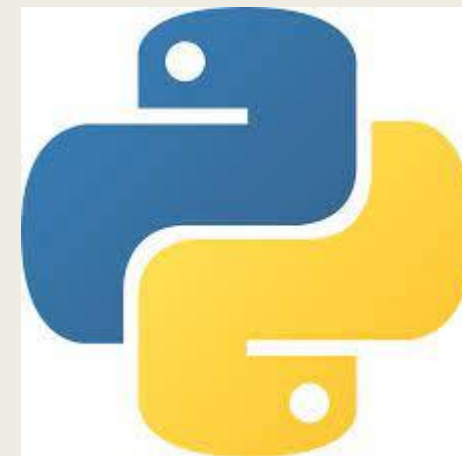
# Способи цифрового зображення місцевості

- **GRID** моделі – регулярне розташуванням точок на прямокутних, трикутниках чи гексагональних сітках.
- **TIN** моделі – нерегулярним розміщення точок по структурних лініях, профілях, центрах майданчиків, локальних точках або випадкових сітках.
- **TGRID** моделі - ізолінійне поданням точок, що розташовані рівномірно на ізолініях, або з урахуванням кривизни горизонталей.



# Мова програмування Python

**Python** — це високорівнева, інтерпретована мова програмування загального призначення, вона динамічно типується та за допомогою неї можна створювати структуровані і функціональні програми.



## Переваги:

- Python не залежить від платформи на якій запускається.
- Python це open source проект.
- Наявна велика кількість бібліотек класів.

## Недоліки:

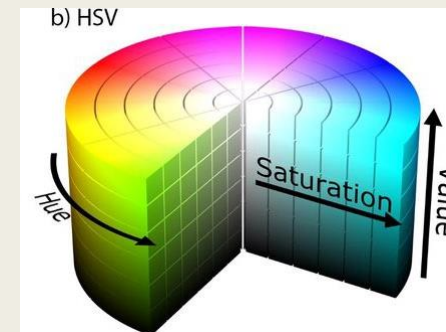
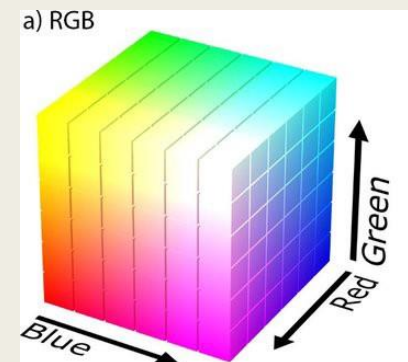
- Низька швидкість виконання задач (порівняно C та C++).
- Динамічна типізація створює труднощі при написанні складних програм.



# Колірні моделі зображень

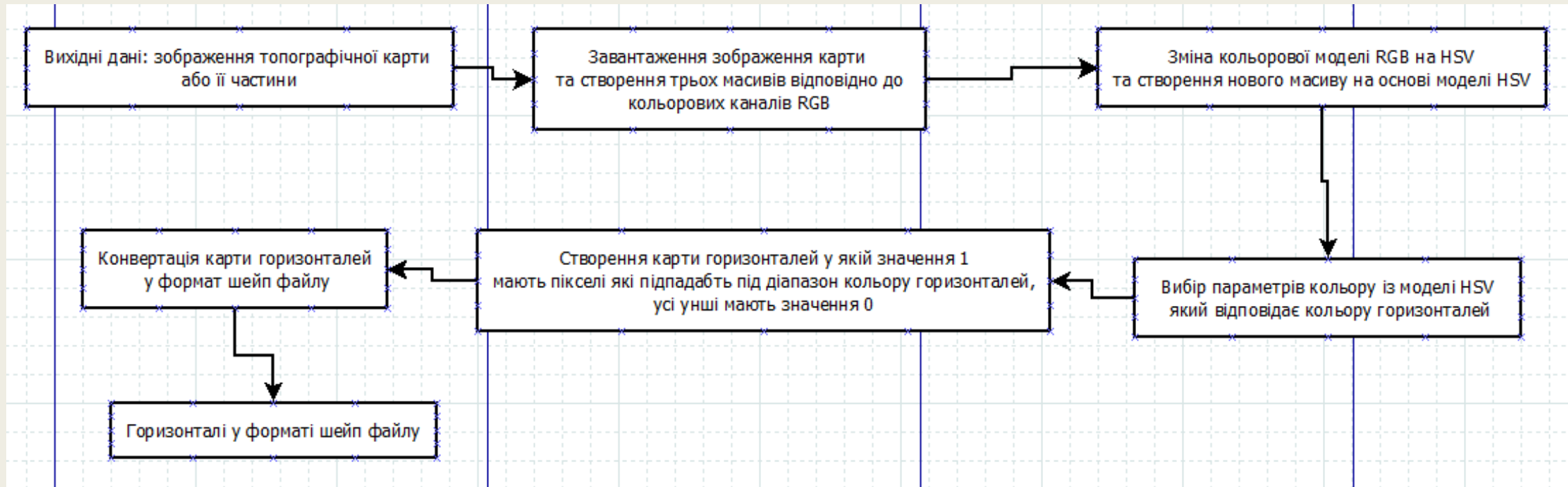
Для *автоматичного розпізнавання* горизонталей *необхідно використовувати ознаки*, за якими горизонталі відрізняються від інших позначень на карті. Найзручнішою ознакою горизонталі є її *колір*, який може бути виміряний за допомогою певної колірної моделі.

- RGB - колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори.
- HSV - колірна модель, побудована на трьох характеристиках кольору: колірному тоні (Hue), насиченості (Saturation) і яскравості (Value).





# Блок-схема процесу автоматизації оцифрування горизонталей за допомогою мови програмування Python



# Функція на мові програмування Python

```
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import matplotlib.colors as mc
6 from PIL import Image
7
8
9 print('Введіть шлях зберігання карти')
10 name = input()
11
12 map = np.asarray(Image.open(name).convert('RGB'))
13 map_r = np.zeros_like(map)
14 map_r[:, :, 0] = map[:, :, 0]
15 map_g = np.zeros_like(map)
16 map_g[:, :, 1] = map[:, :, 1]
17 map_b = np.zeros_like(map)
18 map_b[:, :, 2] = map[:, :, 2]
19 map_hsv = mc.rgb_to_hsv(map_r[:, :, 0], map_g[:, :, 1], map_b[:, :, 2])
20 if map == np.where(map == [146, 134, 82]):
21     horz1 = []
22 else:
23     horz0 = []
24 horz_map = np.asarray(Image.new(horz1))
25 shp_map = shp.convert(horz_map)
26
27 plt.imshow(horz_map)
28 plt.show()
29 map_h = map.shape
30 print(map_h)
```

# Функція оцифрування горизонталей за допомогою Python, частина 1

- ❑ Для початку потрібно прочитати фрагмент карти і створити 3 масиви даних.
- ❑ Кожний масив сприймає значення яскравості у каналах R, G, B.
- ❑ Наступним кроком є перехід до кольорової моделі HSV.

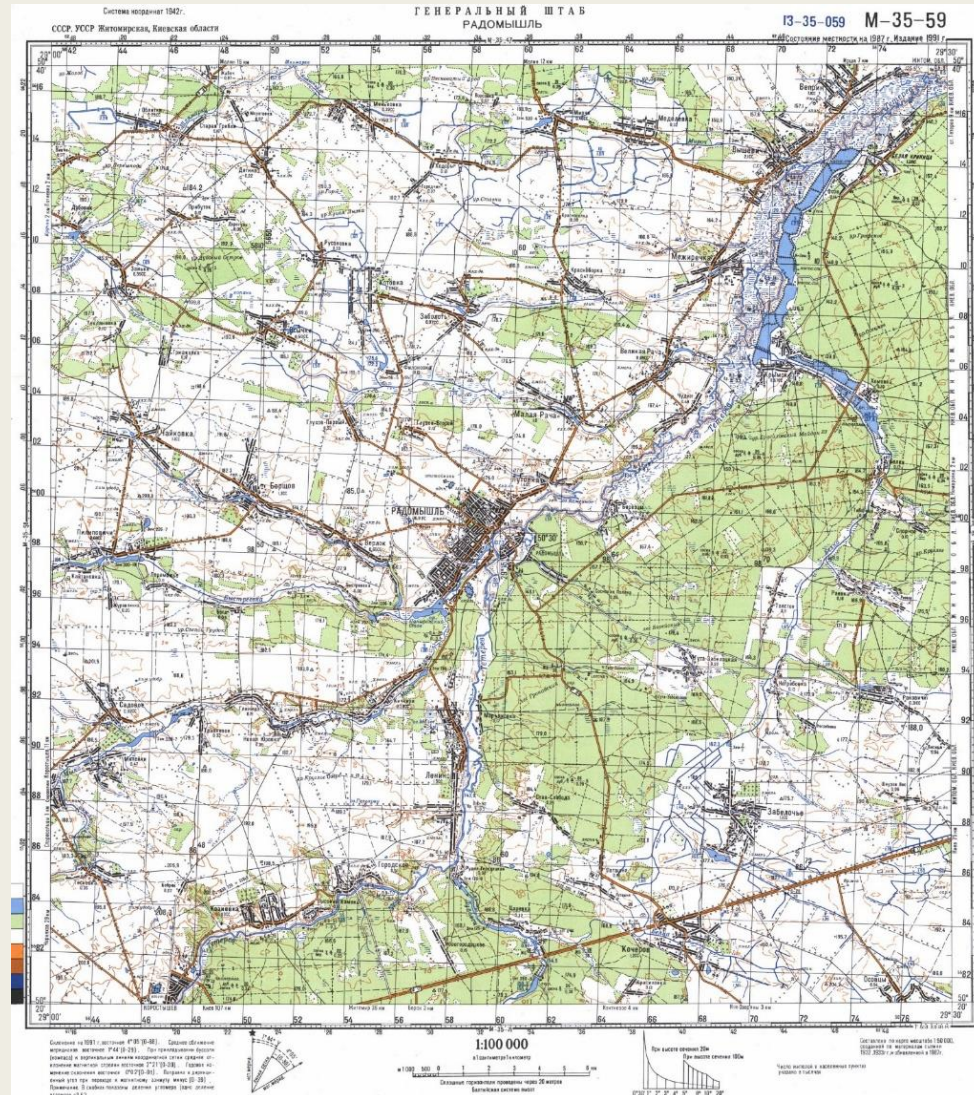
Функція	Пояснення
<pre>import numpy as np import scipy as sc from PIL import Image</pre>	Завантаження всіх необхідних бібліотек до функції
<pre>map = np.asarray(Image.open(name).convert('RGB'))</pre>	Завантаження карти до функції
<pre>map_r = np.zeros_like(map) map_r[:, :, 0] = map[:, :, 0] #зелений канал map_g = np.zeros_like(map) map_g[:, :, 1] = map[:, :, 1] #червоний канал map_b = np.zeros_like(map) map_b[:, :, 2] = map[:, :, 2] #синій канал</pre>	Створення трьох масивів кожен з яких відповідає своєму кольоровому каналу RGB
<pre>map_hsv = mc.rgb_to_hsv(map)</pre>	Перехід до кольорової моделі HSV

## Функція оцифрування горизонталей за допомогою Python, частина 2

Потім потрібно провести класифікацію по значенням HSV. На цьому етапі потрібно вручну встановити діапазон для кожного значення HSV, які відповідають кольору горизонталей. Далі створюється карта горизонталей, у якій значення 1 мають пікселі які відповідають діапазону за всіма параметрами HSV, решті надається значення 0.

Функція	Пояснення
<pre>if map == np.where(map == [146, 134, 82]):     horz1 = [] else:     horz0 = []</pre>	Визначення та відбір кольору горизонталей
<pre>horz_map = np.asarray(Image.new(horz1))</pre>	Створення карти горизонталей
<pre>shp_map = shp.convert(horz_map)</pre>	Конвертація у шейп файл
<pre>plt.imshow(horz_map) plt.show() map_h = map.shape print(map_h)</pre>	Вивід усіх даних та розмірів ділянки карти у пікселях

# Апробація функції Python для оцифрування горизонталей. Вихідні дані



За вихідні дані використано ділянку  
карту Житомирської області,  
масштабу 1:100000, М – 35 – 059



# Апробація функції Python для оцифрування горизонталей. Результати



Вхідні дані



Карта  
горизонталей



ЦМР

# Висновки

- Розглянуто поняття цифрової моделі місцевості, шляхи створення та використання ЦМР.
- Розглянуто особливості мови програмування Python, її переваги та недоліки, бібліотеки, які доповнюють можливості оброблення графічних даних.
- Побудовано технологічну схему процесу автоматизації оцифрування горизонталей.
- Реалізовано та апробовано функцію на мові Python для автоматизації процесу оцифрування горизонталей ділянки карти.