**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

до лабораторної роботи №3

**на тему:** *“Використання OpenCV для реєстрації руху на відеострімах.”*

**з дисципліни** *“Штучний інтелект в ігрових застосунках”*

**Лектор:**

асис. каф. ПЗ

Бауск О. Є.

**Виконав:**

ст. гр. ПЗ-33

Юшкевич. А.І.

**Прийняв:**

ст. викл. каф. ПЗ

Бауск О. Є.

«\_\_» \_\_\_\_\_\_\_\_\_ 2024р.

∑=\_\_\_\_\_

Львів – 2024

**Тема роботи:** Використання OpenCV для реєстрації руху на відеостримах.

**Мета роботи:** Ознайомитись з основами функціонування системи OpenCV, навчитися використовувати її для реєстрації руху на відеострімах.

# Теоретичні відомості

**Що таке OpenCV?**

OpenCV (Open Source Computer Vision Library) — це бібліотека з відкритим вихідним кодом, яка містить понад 2500 алгоритмів для комп'ютерного зору та машинного навчання. Вона широко використовується для обробки зображень та відео, розпізнавання об'єктів, відстеження руху, аналізу сцен та інших задач, пов'язаних з комп'ютерним зором.

**Відстеження особливих точок (Feature Tracking)**

Відстеження особливих точок — це процес визначення та слідкування за рухом певних ключових точок (features) на послідовних кадрах відео. Це дозволяє аналізувати рух об'єктів, стабілізувати відео, розпізнавати жести та багато іншого. Одним з популярних алгоритмів для відстеження особливих точок є алгоритм Лукаса-Канаде (LucasKanade), який реалізований в OpenCV у функції cv2.calcOpticalFlowPyrLK().

**Алгоритм Лукаса-Канаде**

Алгоритм Лукаса-Канаде базується на припущенні, що рух точок між двома послідовними кадрами є малим і приблизно однаковим у локальному околі точки. Він використовує піраміду зображень для ефективного відстеження руху точок на різних масштабах. Основні етапи алгоритму Лукаса-Канаде: 1. Вибір ключових точок на першому кадрі (наприклад, за допомогою алгоритму Ші-Томасі або Харріса). 2. Відстеження цих точок на наступних кадрах за допомогою оптичного потоку. 3. Оновлення позицій точок та повторення процесу для наступних кадрів.

**Щільний оптичний потік**

Щільний оптичний потік — це метод, який обчислює оптичний потік для кожного пікселя зображення. Це дозволяє отримати повну картину руху на всьому зображенні.

**Розріджений оптичний потік**

Розріджений оптичний потік — це метод, який обчислює оптичний потік для деяких ключових точок на зображенні, а не для всіх пікселів. Це дозволяє зменшити обчислювальну складність і збільшити швидкість роботи алгоритму.

**Особливі точки і алгоритм Ші-Томасі**

Алгоритм Ші-Томасі (Shi-Tomasi) — це вдосконалена версія детектора кутів Харріса, що використовується для знаходження особливих точок на зображенні. Цей алгоритм оцінює "якість" кутів на основі мінімальних власних значень матриці градієнтів, що робить його більш надійним для відстеження точок між кадрами.

Основні особливості алгоритму Ші-Томасі:

1. Для кожного пікселя обчислюється матриця градієнтів у локальному вікні.
2. Обчислюються власні значення λ₁ та λ₂ цієї матриці.
3. Якщо min(λ₁, λ₂) > порогове\_значення, то піксель вважається кутом (особливою точкою).

В OpenCV алгоритм Ші-Томасі реалізований у функції cv2.goodFeaturesToTrack() , яка знаходить "сильні" кути на зображенні. Ці точки потім можна відстежувати між кадрами за допомогою алгоритму Лукаса-Канаде ( cv2.calcOpticalFlowPyrLK() ).

Переваги алгоритму Ші-Томасі:

* Більш стабільне відстеження точок порівняно з детектором Харріса
* Краща продуктивність при зміні освітлення та невеликих деформаціях
* Ефективний для задач відстеження руху об'єктів на відеостримах.

Цей алгоритм широко використовується в системах комп'ютерного зору для відстеження об'єктів, стабілізації відео, створення панорамних зображень та інших застосувань, де потрібно відстежувати рух між кадрами.

# Завдання

1. Виявити та виправити помилку в коді програми, пов'язану з ініціалізацією функції create\_default\_lk\_flow .

2. Дослідити три режими роботи програми для відстеження руху:

Режим 1: Щільний оптичний потік з візуалізацією HSV

Режим 2: Щільний оптичний потік з візуалізацією ліній

Режим 3: Метод розрідженого оптичного потоку за Лукасом-Канаде

3. Проаналізовати код програми та визначити місце виклику функції v2.calcOpticalFlowPyrLK() для реалізації методу Лукаса-Канаде.

4. Порівняти переваги та недоліки щільного та розрідженого оптичного потоку:

* Швидкість обчислення
* Точність відстеження
* Ресурсоємність
* Стійкість до шумів

5. Вивчити параметри функції calcOpticalFlowPyrLK та їх вплив на результати роботи алгоритму.

6. Експериментально дослідити вплив різних значень параметрів feature\_params на якість відстеження точок у режимі розрідженого оптичного потоку.

7. Отримати практичні навички роботи з алгоритмами комп'ютерного зору для відстеження руху в відеопотоці.

# Індивідуальне завдання

Використати параметри оптичного потоку в залежності від імені та прізвища автора звіту:

Параметр A = Номер в алфавіті першоі літери вашого імені + номер в алфавіті першої літери вашого прізвища

feature\_params=dict(

maxCorners= A \* 2, # кількість особливих точок

qualityLevel= A / 100, # якість особливих точок, які братимуть участь у відстеженні

minDistance= A \* 5, # мінімальна відстань між особливими точками

blockSize=A \* 2, # розмір блоку для оцінки особливих точок

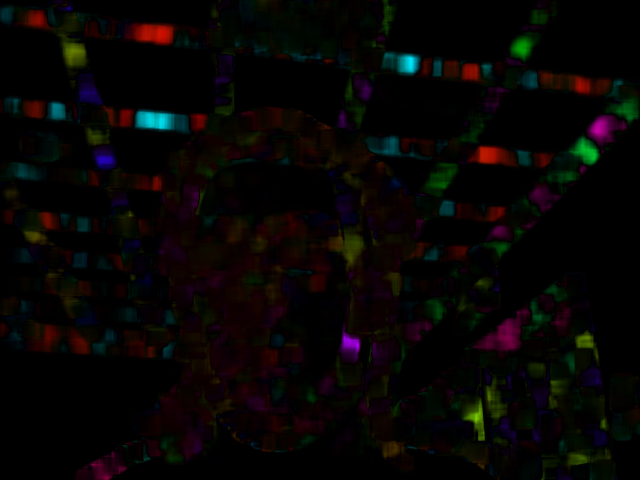
useHarrisDetector=False # НЕ використовувати Harris детектор

)

# Хід роботи

**2. Дослідити три режими роботи програми для відстеження руху:**

Режим 1: Щільний оптичний потік з візуалізацією HSV



*Рис. 1. HSV no movement*

**

*Рис. 2. HSV with movement*

Режим 2: Щільний оптичний потік з візуалізацією ліній

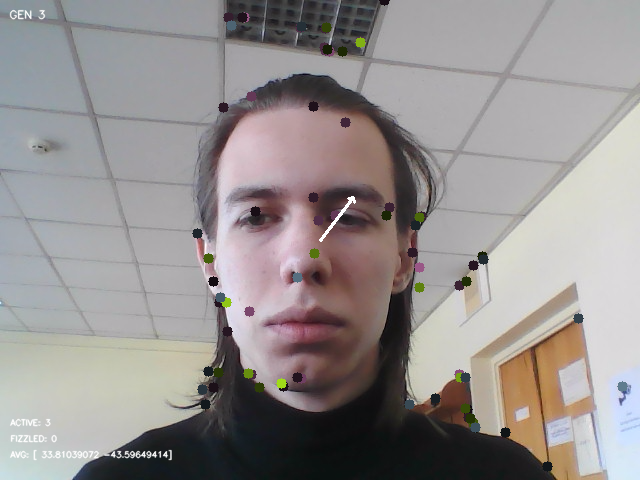
**

*Рис. 3. Lines no movement*

**

*Рис. 4. Lines with movement*

Режим 3: Метод розрідженого оптичного потоку за Лукасом-Канаде

**

*Рис. 5. Lucas-Kanade no movement*

**

*Рис. 3. Lucas-Kanade with movement*

**3. Проаналізовати код програми та визначити місце виклику функції v2.calcOpticalFlowPyrLK() для реалізації методу Лукаса-Канаде.**

Параметри, задані мною в п.1 до цієї лабораторної роботи використовуються при ініціалізації класу всередині якого, ймовірно і викликається вищезазначена функція (89 рядок):  
  
lk\_pipeline = [

GenerationalLKFlow(create\_default\_lk\_flow),

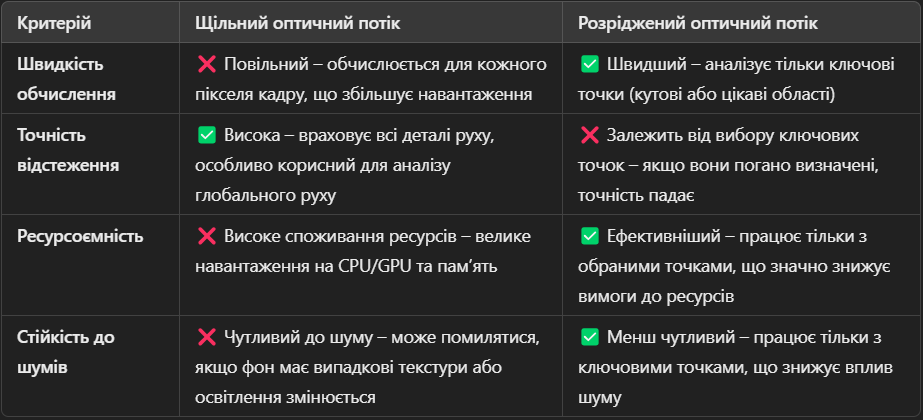
[

GenerationalVectorInference(CompoundTrackerInstantiator),

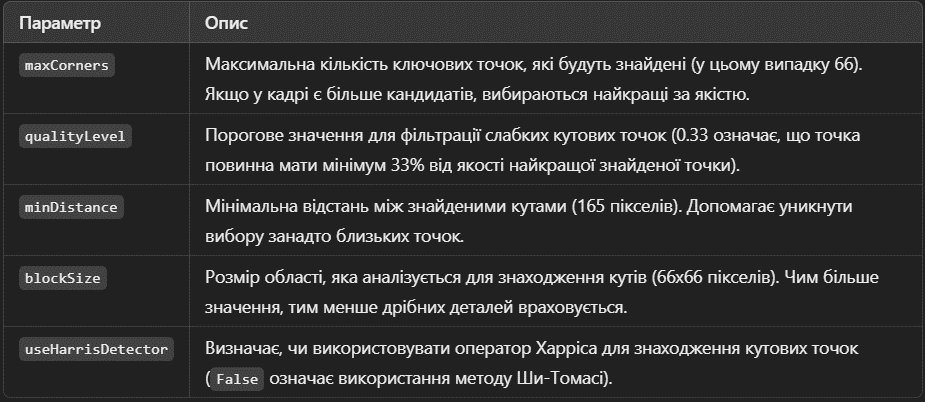
],

GenerationalSparseHUD(),

]

**4. Порівняти щільний та розріджений оптичні потоки:**  


**5. Вивчити параметри функції calcOpticalFlowPyrLK та їх вплив на результати роботи алгоритму**



Отже ми можемо зробити висновки, що мої параметри мають завелику minDistance=165, що робить алгоритм непридатним до використання з відео, яке має розширення 640x480

**Висновок:** Щільний оптичний потік має переваги у точності обчислення, що може бути ефективним при відстеженні будь якого руху, в той час як розріджений потік має вищу ефективність у співвідношенні якість/швидкість та володіє стійкістю до шумів, перепадів світла, ін.