

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра медіаінженерії та інформаційних радіоелектронних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(позначення документа)

Вплив енкодерів на якість зображення прямої трансляції на стрімінгових
платформах

(тема)

Виконав:

студент 2 курсу, групи СТМм-22-1
Руслан ОБЧАРЕНКО

(прізвище, ініціали)

Спеціальність 171 Електроніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи, технології та
комп'ютерні засоби мультимедіа

(повна назва освітньої програми)

Керівник ст. викл. Костянтин КОЛІСНИК
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Володимир КАРТАШОВ
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформації
Кафедра Медіаінженерії та інформаційних радіоелектронних систем

Рівень вищої освіти другий (магістерський)
Спеціальність 171 Електроніка

(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма " Системи, технології та комп'ютерні засоби мультимедіа "

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Овчаренко Руслану Руслановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Вплив енкодерів на якість зображення прямої трансляції на стрімінгових платформах

Затверджена наказом по університету від " 20 " листопада 2023 р. № 1371 Ст _____

2. Термін здачі студентом закінченої роботи 08.01.2024

3. Вихідні дані до роботи:

1.Провести дослідження стиснення зображення, методи та алгоритми 2.Провести дослідження існуючих енкодерів, їх особливостей, налаштувань 3.Розробити рекомендації до ПК для проведення прямих трансляцій 4. Провести налаштування OBS для подальшого дослідження 5.Провести дослідження існуючих можливостей для проведення мультитрансляції 6. Визначити методіку тестування енкодерів 7. Зробити порівняння якості вихідного відео при різних енкодерах 8. Зробити порівняння якості вихідного відео на різних платформах

4. Зміст розрахунково - пояснювальної записки (перелік питань, які підлягають розробці):

1. СТИСНЕННЯ ЗОБРАЖЕННЯ 2. ЕНКОДЕРИ І ЇХ ВИДИ 3. КОМПЛЕКТУЮЧІ КОМП'ЮТЕРА ДЛЯ ПРЯМОЇ ТРАНСЛЯЦІЇ 4. НАЛАШТУВАННЯ OBS 5. ТРАНСЛЯЦІЯ НА ДЕКІЛЬКА СТРІМІНГОВИХ ПЛАТФОРМ ОДРАЗУ 6. МЕТОДИКА ТЕСТУВАННЯ ЕНКОДЕРІВ 7. ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ X264 8. ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ NVENC H.264 9. ПОРІВНЯННЯ ДЕКОДЕРІВ ПЛАТФОРМ TWITCH ТА YOUTUBE

5. Перелік графічного матеріалу (з точним визначенням обов'язкових креслень): 1. Порівняння пресетів Ultrafast-Medium енкодеру x264 у першій грі 2. Порівняння пресетів P1-P7 енкодеру NVENC у першій грі 3. Порівняння якості трансляції на платформах Twitch і YouTube у першій грі 4. Порівняння пресетів Ultrafast-Medium енкодеру x264 у другій грі 5. Порівняння пресетів P1-P7 енкодеру NVENC у другій грі 6. Порівняння якості трансляції на платформах Twitch і YouTube у другій грі 7. Порівняння пресетів Ultrafast-Medium енкодеру x264 у третій грі 8. Порівняння пресетів P1-P7 енкодеру NVENC у третій грі 9. Порівняння якості трансляції на платформах Twitch і YouTube у першій грі

6. Дата видачі завдання: 26.09.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Дослідження методів стиснення зображення	30.09.2023 - 14.10.2023	
2	Дослідження існуючих енкодерів, їх особливостей, налаштувань.	14.09.2023 – 24.10.2023	
3	Розробка рекомендацій до ПК для проведення прямих трансляцій	01.11.2023 – 03.11.2023	
4	Налаштування OBS для подальшого дослідження енкодерів	03.11.2023 – 05.11.2023	
5	Дослідження існуючих можливостей для проведення мультитрансляції	05.11.2023 – 15.11.2023	
6	Визначення методики тестування енкодерів	15.11.2023 – 16.11.2023	
7	Порівняння якості вихідного відео при різних енкодерах	16.11.2023 – 18.11.2023	
8	Порівняння якості вихідного відео на різних платформах	20.11.2023 – 22.11.2023	
9	Графічна частина роботи	22.11.2023 – 30.12.2023	
10	Перевірка керівником	03.12.2023 – 05.12.2023	
11	Перевірка на академічний плагіат	05.12.2023 – 07.12.2023	
12	Перевірка зав. кафедри, рецензування	09.12.2023 – 11.12.2023	

Студент



(підпис)

Руслан ОВЧАРЕНКО
(прізвище, ім'я, по батькові)

Керівник роботи

(підпис)

Костянтин КОЛІСНИК
(прізвище, ім'я, по батькові)

«__» _____ 20__ р.

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи має: 77 сторінок, 68 рисунків, 7 формул, 1 таблицю, 19 джерел.

СТИСНЕННЯ, ЗОБРАЖЕННЯ, ПРЯМА ТРАНСЛЯЦІЯ, СТРИМ, ВІДЕО, ІНТЕРНЕТ, ЕНКОДЕР, РОЗДІЛЬНА ЗДАТНІСТЬ, БІТРЕЙТ

Об'єкт дослідження – мультимедійні системи.

Предмет дослідження – якість відео при кодуванні різними енкодерами.

Мета кваліфікаційної роботи – аналіз налаштувань прямої трансляції та дослідження впливу енкодерів на якість відео.

Методи дослідження – теоретичний аналіз, експериментальне дослідження.

У даній роботі проведено аналітичний огляд засобів, технологій та методів, які використовуються при проведенні прямої трансляції. Проведено огляд та аналіз алгоритмів стиснення відео, можливостей програми OBS, можливостей сторонніх сервісів для проведення трансляцій на декілька платформ одразу, енкодерів та їх налаштувань. Було проведено налаштування програмного забезпечення OBS для проведення прямої трансляції. В ході роботи були створені відео та фото порівняння якості відео після його кодування та порівняння якості відео на різних платформах.

ABSTRACT

The explanatory note of the qualification work has: 77 pages, 68 figures, 7 formulas, 1 table, 19 sources.

COMPRESSION, PICTURE, LIVE BROADCAST, STREAM, VIDEO, INTERNET, ENCODER, RESOLUTION, BITRATE

Object of research - multimedia systems.

Subject of research - video quality when encoding with different encoders

The purpose of the qualification work is to analyze the settings of live streaming and study the impact of encoders on video quality.

Research methods - theoretical analysis, experimental study.

This work provides an analytical review of the tools, technologies, and methods used for live broadcasting. Reviews and analyzes video compression algorithms, capabilities of the OBS program, capabilities of third-party services for broadcasting several platforms at once, encoders and their settings. The OBS software was configured for live broadcasting. In the course of the work, we created videos and photos comparing the quality of the video after encoding and comparing the quality of video on different platforms.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- ДКМ - диференційна імпульсно-кодова модуляція.
- ІКМ - імпульсно-кодова модуляція.
- ДПФ – дискретне перетворення Фур'є.
- ШПФ – швидке перетворення Фур'є.
- ДКП - дискретне косинусне перетворення.
- RLE – Run-Length Encoding
- FULL HD (Full High Definition) – роздільна здатність 1920x1080 точок.
- 1080p60 – скорочення якості відео full hd (1920x1080) 60 кадрів за секунду.
- RTMP (Real Time Messaging Protocol) – протокол потокової передачі даних.
- ПК – персональний комп'ютер.
- КБ\С – кілобіт за секунду.
- ЦП – центральний процесор.
- CPU (Central Processing Unit) – центральний процесор.
- GPU (Graphics Processing Unit) – графічний процесор або відеокарта.
- RAM (Random Access Memory) – оперативна пам'ять.
- SSD (Solid State Drive) – твердотільний накопичувач.
- FPS (Frames Per Second) - кількість кадрів за секунду.
- CBR (Constant Bit Rate) – постійний бітрейт.
- VBR (Variable Bitrate) – змінний бітрейт.
- X264 – програмний енкодер який використовує потужності процесора для кодування.
- NVENC (Nvidia Encoder) – апаратний енкодер який використовує потужності відеокарти для кодування.

ЗМІСТ

ВСТУП.....	8
1. СТИСНЕННЯ ЗОБРАЖЕННЯ.....	9
1.1 Надмірності відеоінформації.....	9
1.2 Внутрішньокадрове стиснення відео.....	14
1.3 Дослідження внутрішньокадрового стиснення.....	25
2. ЕНКОДЕРИ І ЇХ ВИДИ	27
3. КОМПЛЕКТУЮЧІ КОМП'ЮТЕРА ДЛЯ ПРЯМОЇ ТРАНСЛЯЦІЇ.....	29
3.1 Вибір комплектуючих	29
3.2 Комплектуючі комп'ютера на якому проводилось дослідження	30
4. НАЛАШТУВАННЯ OBS	31
4.1 Підключення акаунту	31
4.2 Роздільна здатність та частота оновлення.....	32
4.3 Вхідний бітрейт і їх види.....	33
4.4 Енкодери і їх налаштування.....	35
4.5 Налаштування сцени	39
5. ТРАНСЛЯЦІЯ НА ДЕКІЛЬКА СТРІМІНГОВИХ ПЛАТФОРМ ОДРАЗУ	44
5.1 Сторонні сервіси.....	45
5.2 Плагіни OBS.....	46
6. МЕТОДИКА ТЕСТУВАННЯ ЕНКОДЕРІВ	48
7. ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ H264	50
8. ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ NVENC H.264.....	56
9. ПОРІВНЯННЯ ДЕКОДЕРІВ ПЛАТФОРМ TWITCH ТА YOUTUBE	66
ВИСНОВКИ	72
ПЕРЕЛІК ДЖЕРЕЛ, ПОСИЛАНЬ.....	75
ДОДАТКИ.....	78
ДОДАТОК А.....	79
ДОДАТОК Б	88

ВСТУП

Зі швидким розвитком цифрових технологій і поширенням високошвидкісного інтернету стрімінгові платформи, визначають новий етап споживання контенту в галузі відео. Стрімінгові платформи дають змогу користувачам надавати та споживати відео контент будь де, в тому числі у режимі реального часу, взаємодіючи один з одним через коментарі та чати.

Енкодер, як ключовий елемент у технології стрімінгу, відіграють рішучу роль у забезпеченні високої якості та ефективного використання пропускну здатності мережі. Вони відповідають за перетворення великого масиву відеоданих у формат, який може бути легко переданий через інтернет. Ефективне використання енкодерів дозволяє забезпечити високу якість зображення, зберігаючи при цьому економічність передачі даних.

З огляду на доступні нині енкодері і їхні зростаючі можливості, стає питання про вибір оптимальних налаштувань для конкретних умов. Також слід враховувати специфіку таких стрімінгових платформ, як Twitch і YouTube, та їхні вимоги до параметрів кодування відео. Вивчення впливу різних кодувальників на якість зображення під час прямої трансляції на цих платформах є необхідним для оптимізації процесу стрімінгу та покращення досвіду глядачів.

Основною метою дипломного проекту є глибоке дослідження взаємодії енкодерів та їхнього впливу на якість відео під час прямих трансляцій на стрімінгових платформах. Дослідження включатиме аналіз різних типів енкодерів, їхніх параметрів та взаємодії з особливостями платформ. Twitch та YouTube.

На основі отриманих даних та результатів дослідження планується розробка конкретних рекомендацій для вибору та налаштування енкодерів для оптимальної якості відео на стрімінгових платформах. Це допоможе покращити якість стрімінгу, забезпечуючи користувачам якісний досвід перегляду контенту у режимі реального часу.

1 СТИСНЕННЯ ЗОБРАЖЕННЯ

1.1 Надмірності відеоінформації

Відеозображення має три види надмірності:

- Статистична, пов'язана з кореляцією (схожістю) сусідніх рядків та елементів у рядку (внутрішньокадрова, просторова) та кореляцією зображень у сусідніх кадрах (міжкадрова, часова);
- Психовізуальна, пов'язана з властивостями людського зору, внаслідок чого не всі елементи зображення потребують однакової точності квантування;
- Структурна, пов'язана з наявністю службової інформації – сигналів синхронізації, захисту від помилок, запасу за рівнем між сигналами зображення та синхронізації тощо.

У всіх методах стиснення [13] головну увагу приділяють першим двом видам надмірності:

Статистична надмірність

Для зменшення статистичної надмірності застосовується диференціальна імпульсно-кодова модуляція (ДІКМ). У цьому методі кодується різниця між поточним та попереднім відліками сигналу.

Більш ефективно усувають надмірність методи ДІКМ з передбаченням (Рис. 1.1), у яких кодуванню підлягає різниця між попереднім відліком та передбаченим значенням наступного пікселя. Як передбачення береться значення попереднього елемента по горизонталі (одномірне передбачення). У складніших випадках передбачення ґрунтується на інформації у попередньому рядку (двовимірне передбачення) чи наступному кадрі (міжкадровий прогноз). Вхідним сигналом є сигнал ІКМ. Відліки вихідного сигналу кодуються методом ІКМ. Для кожного вхідного відліку $s(x)$ точки x провісник за $N-1$ попередніми відліками визначає передбачене значення $\hat{s}(x)$. У цьому випадку передбачене значення x пікселя

визначається за формулою:

$$\hat{s}(x) = \sum_{i=1}^{N-1} c_i \cdot \hat{s}(x - i),$$

де c_i – вагові коефіцієнти передбачення, які вибираються щоб мінімізувати дисперсію помилки передбачення $e(x) = s(x) - \hat{s}(x)$.

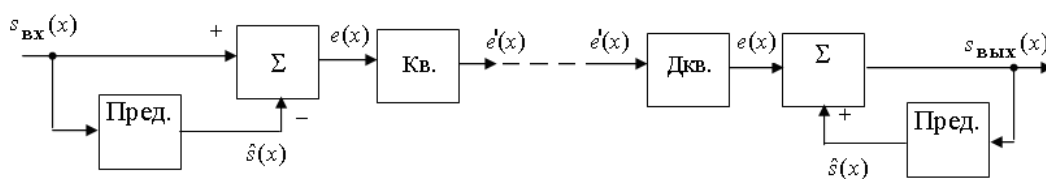


Рисунок 1.1 – Спрощена структурна схема ДІКМ із передбаченням [13]

При застосуванні ДІКМ помилка має меншу величину, ніж сам відлік сигналу і за її квантування можна використовувати менше біт. Квантований сигнал помилки передбачення надходить у деквантувач, у якому відновлюється вихідне число двійкових розрядів. Таким чином, каналом зв'язку передається різниця між дійсним значенням поточного відліку сигналу і значенням цього ж відліку, передбаченим за попередніми відліками.

Корисна дія ДІКМ становить 3-5 біт на один елемент, але метод знижує стійкість до перешкод, що призводить до спотворень відновленого зображення (Рис. 1.2). Вирішити це можна шляхом частішої передачі “опорних” відліків, тобт, самих значень елементів зображення. У цьому випадку дія перешкоди зображення припиняється з появою найближчого істинного значення сигналу. Збільшення числа опорних значень сигналу ДІКМ знижує ефективність стиснення.

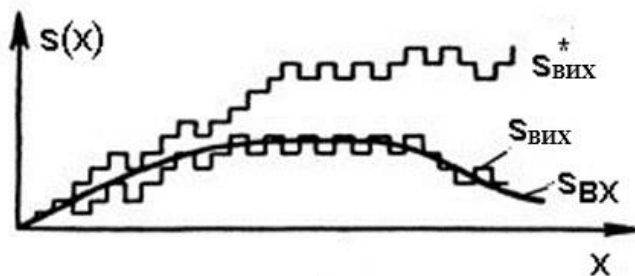


Рисунок 1.2 – Поява треків помилки при ДКМ [13]

Ще одним недоліком ДКМ є перевантаження по крутизні (Рис. 1.3), коли сигнал змінюється швидше, ніж його здатна передати квантована помилка передбачення із зменшеним числом біт.



Рисунок 1.3 – Поява перевантаження по крутості при ДКМ [13]

Щоб уникнути появи цих спотворень, максимальна крутизна вхідного сигналу не повинна перевищувати граничного значення.

$$\Delta s_{\text{пред}}(x) = \delta \cdot \Delta x ,$$

де δ – максимальна величина квантованої помилки передбачення, що може передати система.

Спотворення, спричинені помилкою передбачення, можуть проявляти себе як усередині кадру (у просторовій області), так і між кадрами (часовій області), залежно від того, де буде застосована ДКМ (Рис. 1.4, 1.5).

Просторова надмірність – колір багатьох сусідніх точок однаковий.

Часова надмірність – сусідні кадри дуже схожі.



Рисунок 1.4 – Сусідні кадри фільму (Термінатор-2)[3]



Рисунок 1.5 – Міжкадрова різниця (Термінатор-2)[3]

Саме такі кадри (з урахуванням поправки на компенсацію руху) і стискає кодек. Їх понад 99% у потоці. Амплітуди малі, це дає можливість зменшити кількість біт на відлік. Зображення майже однорідне, що зменшує просторову збитковість.

Психовізуальна надмірність

Психовізуальна надмірність при спостереженні одного кадру полягає у погіршенні характеристик зору при спостереженні дрібних деталей.

При передачі інформації про колір психовізуальна надмірність кольорових

сигналів скорочується в 4 рази в порівнянні зі смугою сигналу яскравості. Подібний підхід застосовується і під час субдискретизації сигналів кольоровості у форматах 4:2:2, 4:2:0, 4:1:1, 4:1:0.

Поріг контрастної чутливості зору ($\varepsilon = \Delta L/L$ – мінімальне відносне збільшення яскравості, сприймане зором) дорівнює 2 - 3%. Однак це значення справедливе лише для великих деталей спотворення. Для дрібних деталей потрібно більше відносне збільшення яскравості $\Delta L/L$, щоб наш зір був здатний розрізнити їх.

На (Рис. 1.6) показаний графік залежності порога контрастної чутливості (вісь зліва) та зворотної величини – контрастної чутливості $L/\Delta L$ (вісь праворуч) від просторової частоти зображення (горизонтальна вісь).

Просторова частота вимірюється в циклах (число перепадів яскравості) на один градус зображення, що спостерігається. Чим більша просторова частота, тим дрібніші деталі зображення. На графіку умовно показано поділ деталей на великі, дрібні та середні.

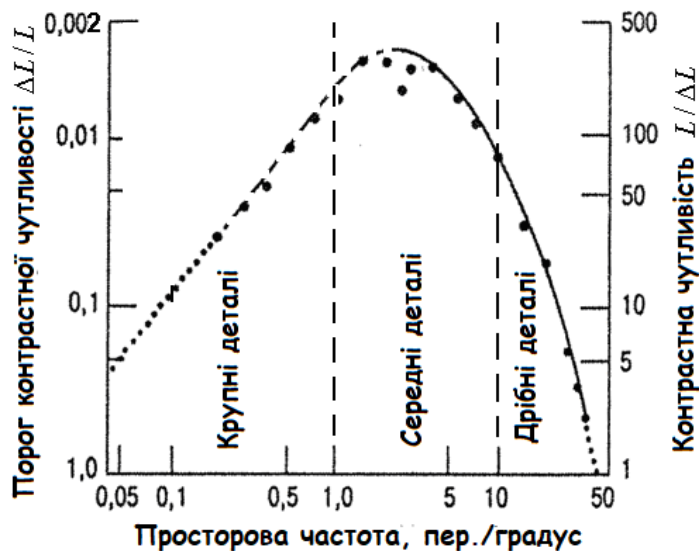


Рисунок 1.6 – Частотно-контрастна характеристика[15]

Для області високих просторових частот (дрібні деталі) характерна як низька контрастна чутливість, а й мале число одночасно помітних градацій яскравості. При розмірі деталей, що спостерігаються, близько 10 кут. хв. кількість помітних

градацій яскравості становить лише 2–4 рівня. На межі дозволу контрастна чутливість вкрай низька і зір може сприйняти лише три рівні яскравості: чорне та біле щодо сірого фону, а також фон (як відсутність сигналу).

Знання про смугу просторових частот, що сприймаються, дозволяє стиснути смугу переданих частот у сигналі зображення.

Чутливість зору сприйняття дрібних деталей також залежить від часу спостереження зображення. На динамічних сценах людині не вистачає часу, щоби почати розглядати дрібні деталі. З цієї причини вже в системах аналогового ТБ число рядків розкладання обрано дещо меншим, ніж впливає з кутової роздільної здатності зору (575 видимих замість 720).

1.2 Внутрішньокадрове стиснення відео

Під внутрішньокадровим стисненням розуміють алгоритм, який стискає дані кожного кадру зображення без урахування попередніх та (або) наступних кадрів відеопослідовності. Внутрішньокадрові алгоритми аналогічні стиску нерухомих зображень. У зв'язку з цим розглянемо алгоритм JPEG, який входить як складова частина алгоритми стиснення відеозображень MJPEG і MPEG.

Основна ідея алгоритму JPEG полягає у розкладанні зображення за просторовими частотами та передачі інформації про амплітуди цих просторових частот з різною точністю квантування. Більшій точності передачі вимагають низькі просторові частоти, що несуть інформацію про великі деталі, менша точність потрібна для верхніх просторових частот, в яких міститься інформація про дрібні деталі зображення. Завдяки недосконалості людського зору, складові ВЧ можна квантувати грубіше без помітної втрати якості зображення.

Алгоритм стиску JPEG [16]:

- Переведення зображення з колірному простору RGB в колірний простір YCrCb (YUV).

За рахунок того, що людське око менш чутливе до кольору, ніж до яскравості,

з'являється можливість архівувати масиви для Cr і Cb компонент з більшими втратами і, відповідно, більшими коефіцієнтами стиснення. Таке перетворення використовується і в аналоговому телебаченні. На кольоророзносні сигнали там виділяється в 4 рази вужча смуга частот, ніж для сигналу яскравості.

Переведення з колірного простору RGB (3 канали по 8 біт) в колірний простір YCrCb можна представити за допомогою матриці переходу:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Зворотне перетворення здійснюється множенням вектора YCrCb на зворотню матрицю.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{pmatrix} * \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix}$$

- Субдискретизація (проріджування) компонентів кольоровості. Проводиться відповідно до форматів 4:2:2, 4:2:0, 4:1:1, 4:1:0. Без обчислювальних витрат стискає цифровий потік без помітних втрат якості (колірна чіткість погіршується, але очі цього майже помічають).

При стисканні можна вибрати режим субдискретизації (Рис.1.7)

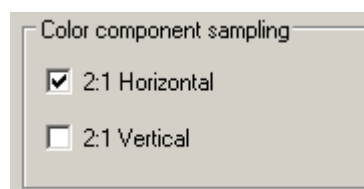


Рисунок 1.7 – Вибір режиму субдискретизації

- Перетворення зображення на область просторових частот.

Одновимірний сигнал $s(t)$, що змінюється у часі, йому однозначно та взаємно відповідає комплексний спектр $S(f) \stackrel{F}{\leftrightarrow} s(t)$, який обчислюється через перетворення Фур'є. Частоти f складових спектра $S(f)$ показують, скільки періодів коливань цих складових вміщується в 1 секунду сигналу.

Окремо взятий кадр зображення є двовимірним сигналом $s(x, y)$, де x, y – просторові координати, у випадку цифрового зображення – номери пікселів, а s – один із сигналів YCrCb. Йому так само, як і одновимірному сигналу, можна поставити у відповідність спектр, але вже двовимірний:

$$S(k_x, k_y) \leftrightarrow s(x, y),$$

де k_x, k_y - просторові частоти, які показують, скільки періодів складових зображення припадають на одиницю ширини або висоти кадру.

Перетворення кадру зображення на область просторових частот, в принципі, також можна здійснювати через перетворення Фур'є. Недолік ДПФ у цьому випадку полягає в тому, що відліки спектру носять комплексний характер, що складають на дзеркальних частотах комплексно сполучені (Рис.1.8). Таким чином, має місце деяка надмірність уявлення спектра зображення за допомогою ДПФ.

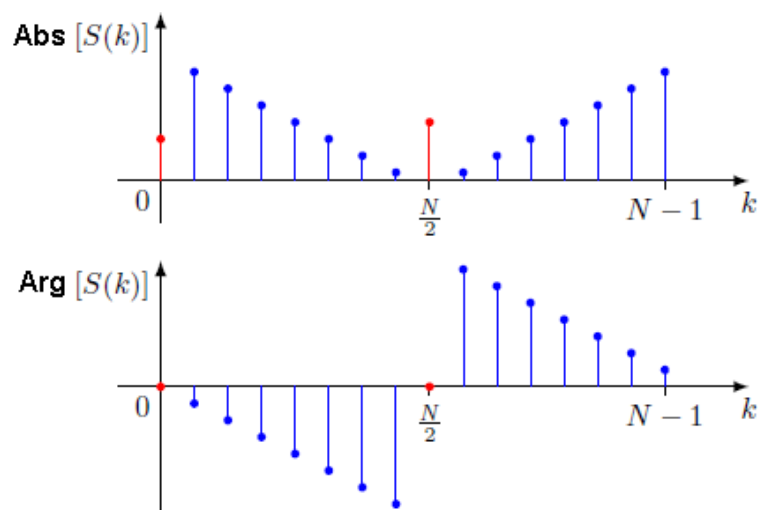


Рисунок 1.8 – Амплітудний та фазовий спектр ДПФ[16]

Враховуючи сказане, доцільніше здійснювати перетворення, в якому використовується тільки члени \cos або \sin , також, необхідно забезпечити повноту ортогональних базових функцій C_k .

У стандарті JPEG замість ДПФ використовується ДКП – дискретне косинусне перетворення з базовими функціями виду

$$C_k = \cos \frac{2n+1}{2N} k\pi, k, n = 0, 1, \dots, N - 1,$$

k – номер дискретної просторової частоти,

n – номер дискретного просторового відліку сигналу.

У JPEG та MPEG ДКП застосовується до блоків зображення розміром 8×8 . Такий розмір є компромісом між складністю обчислень та представністю вибірки.

Для обчислення прямого та зворотного одновимірного ДПФ потрібно N^2 операцій складання та множення, для двовимірного – $N^2 \cdot M^2$ операцій, де N і M – розміри масиву. Наприклад, обчислення ДПФ для складності яскравості цілого кадру 720×576 вимагатиме $1,7 \cdot 10^{11}$ операцій, а у разі розбиття його на квадратні блоки 8×8 – $\sim 2,7 \cdot 10^7$ операцій. Подальше зменшення розмірів блоку недоцільно, оскільки у меншому розмірі неможливо виявити характерні періодичні структури зображення, тобто, вибірка перестає бути представницькою.

Для подальшого зменшення обсягу обчислень застосовують алгоритми швидкого перетворення (ШПФ), у яких розміри масиву мають бути кратними 2^n , а кількість необхідних операцій дорівнює $NM \log_2 NM$. Для наведеного прикладу, коли розмір кадру 720×576 , а блоки мають розмір 8×8 , загальна кількість операцій складе $2,5 \cdot 10^6$ операцій.

Застосовуючи ДКП до кожної робочої матриці 8×8 , отримуємо матрицю також розміром 8×8 , у якій коефіцієнти у лівому верхньому куті відповідають низькочастотній складовій зображення, а правому нижньому – високочастотній.

Перетворення ДКП можна представити так:

$$S(k_x, k_y) = \frac{1}{4} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(x, k_x) \cdot C(y, k_y) \cdot s(x, y),$$

де $C(i, k) = A(k) \cdot \cos \left[\frac{(2i+1) \cdot k\pi}{2N} \right]$ - базисні функції ДКП,

$A(k) = 1/\sqrt{2}$ при $k = 0$, і $A(k) = 1$ при $k \neq 0$.

$s(x, y)$ – відліки блоків яскравості чи кольоровості, а $S(k_x, k_y)$ – коефіцієнти їх ДКП, просторовий спектр. Для більш ясного розуміння фізичного сенсу ДКП розглянемо вигляд базисних функцій $C(i, k)$ (Рис. 1.9).

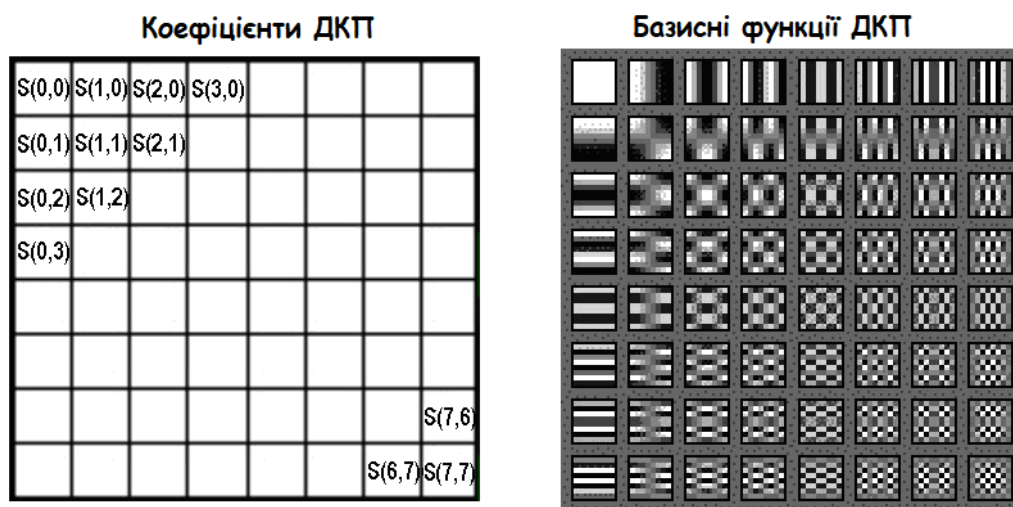


Рисунок 1.9 – Фізичний зміст коефіцієнтів ДКП[5]

Коефіцієнти ДКП – це вагові коефіцієнти, із якими потрібно підсумовувати відповідні базисні функції, щоб отримати зображення межах блоку 8×8 .

Базова функція $C(0,0)$ представляє постійну складову, тобто, середню яскравість або колір у межах блоку. Функції $C(1,0)$, $C(2,0)$, $C(3,0)$ і т.д. мають вигляд періодичних структур, що змінюються по горизонталі, $C(0,1)$, $C(0,2)$, $C(0,3)$ – періодичні структури, що змінюються по вертикалі, $C(x, x)$ – по діагоналі. Сума всіх базисних функцій $C(*,*)$ з ваговими коефіцієнтами $S(*,*)$ дає зображення в межах блоку. Перехід від подання у вигляді зображення до коефіцієнтів ДКП не змінює кількості інформації і не вносить втрат.

- Квантування коефіцієнтів ДКП.

Квантування здійснюється шляхом поділу робочої матриці $S(k_x, k_y)$ на матрицю квантування $q(k_x, k_y)$ поелементно.

$$S_q(k_x, k_y) = \text{int} \left\{ \frac{S(k_x, k_y)}{g \cdot q(k_x, k_y)} \right\},$$

де int - означає округлення числа до найближчого цілого,
 g - параметр стиснення.

На етапі квантування здійснюється управління ступенем стиснення, і відбуваються найбільші втрати. У найпростішому разі, операція поділу на 2^n – це арифметичний побітовий зсув праворуч на n біт (Рис.1.10).

В результаті квантування зменшується кількість біт, але втрачається частина інформації. Зворотна операція (деквантування) здійснюється шляхом множення на 2^n шляхом побітового зсуву вправо на n біт. При цьому в молодші біти записуються нулі, хоча у вихідному коді в молодших бітах могли бути одиниці. Ця різниця і є втратою інформації при квантуванні.

Квантування (ділення)	Деквантування (множення)
$1011011\overline{1} / 2 = 1011011$	$1011011 \cdot 2 = 1011011\overline{0}$
$101101\overline{11} / 2^2 = 101101$	$101101 \cdot 2^2 = 101101\overline{00}$
$10110\overline{111} / 2^4 = 10110$	$10110 \cdot 2^4 = 10110\overline{0000}$

Рисунок 1.10 – Поділ двійкового числа на 2^n [5]

Задаючи МК з великими коефіцієнтами, маємо велику ступінь стиснення. У стандарти JPEG та MPEG включені таблиці, побудовані досвідченим шляхом. Управління коефіцієнтом стиснення одержують шляхом множення вихідної матриці $q(k_x, k_y)$ на деяке число g , яке називається параметром стиснення. Налаштування ступеня стиснення доступне користувачеві в більшості кодеків (Рис.

1.11).



Рисунок 1.11 - Регулювання параметру g

Для кожної компоненти YCrCb у загальному випадку задається своя матриця квантування (Quantization Table). Приклад матриць квантування для компонент яскравості та кольоровості показаний на (Рис. 1.12).

Яскравість								Колірність							
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	58	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

Рисунок 1.12 – Рекомендовані матриці квантування [5]

Коефіцієнти квантування і втрати збільшуються від низькочастотних коефіцієнтів до високочастотних. Квантування компонентів кольоровості відбувається з більшими коефіцієнтами, ніж квантування коефіцієнтів ДКП яскравості.

Необхідно також зазначити, що при передачі постійних складових використовується ДКМ із передбаченням. Замість постійної складової $S_q(0,0)$ для блоку з порядковим номером використовується передача різниці

$$\Delta S_{qn}(0,0) = S_{qn}(0,0) - S_{q(n-1)}(0,0),$$

пропускаються, а "число" - число відмінне від нуля, наступне за групою нулів. Наприклад, вектор 42 3 0 0 0 -2 0 0 0 0 1 0 0 0 0 0 буде згорнутий у пари (0,42) (0,3) (3,-2) (4,1) (EOB), де EOB – символ закінчення блоку (End of Block).

Кількість інформації після групового кодування помітно менша, ніж до нього. Групове кодування ефективно при утворенні довгих ланцюжків нулів, що найбільш ймовірно у високочастотній частині спектра, де самі ВЧ коефіцієнти ДКП мають невелику величину і в процесі квантування діляться на великі числа. Тому без попереднього етапу, зигзаг-сканування, алгоритм групового кодування не дав би суттєвого результату.

Кодування за Хаффманом.

Згортаємо пари, що вийшло, кодуванням по Хаффману з фіксованою таблицею. При цьому малі серії нулів і малі величини ненульових коефіцієнтів вірогідніші, тому їм ставляться у відповідність короткі кодові слова. Символ EOB кодується найкоротшою кодовою комбінацією.

Алгоритм Хаффмана це класичний алгоритм що аналізує лише частоту появи однакових байт у зображенні. Порівнює символи вхідного потоку, які зустрічаються більше разів, ланцюжок бітів меншої довжини і, навпаки, що зустрічаються рідко – ланцюжок більшої довжини. Для збору статистики потрібен окремий прохід зображення.

Класичний алгоритм Хаффмана вимагає запису у файл таблиці відповідності кодованих символів і ланцюжків, що кодують.

Насправді використовуються різновиди класичного алгоритму Хаффмана. Так в деяких випадках краще використовувати постійну таблицю, або будувати її “адаптивно”, тобто, у процесі архівації/розархівації. Ці прийоми позбавляють необхідності зберігання таблиці разом із файлом. Можливість однопроходового (1-pass) кодування особливо важлива при стисканні в реальному часі.

Кодування з фіксованою таблицею застосовується як останній етап архівації в JPEG і як одна з налаштувань кодеків MPEG (Рис. 1.14).

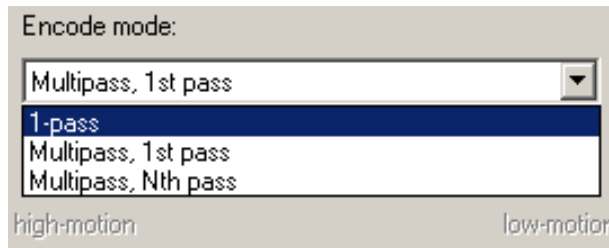


Рисунок 1.14 – Налаштування кодеку DivX 5 (вибір числа проходів) [5]

Конвеєр операцій при внутрішньокадровому кодуванні показано на (Рис.1.15).

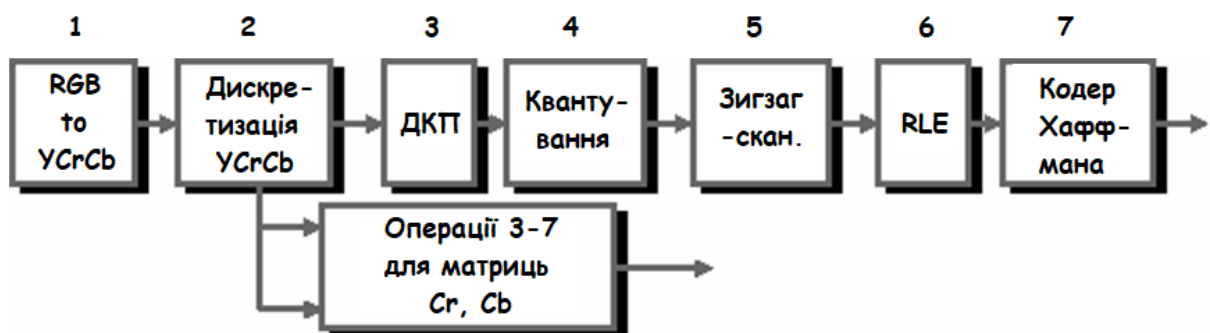


Рисунок 1.15 – Конвеєр операцій при внутрішньокадровому кодуванні [5]

Характеристики алгоритму JPEG

Переваги алгоритму:

- Регулюється ступінь стиснення (параметром g);
- Вихідне кольорове зображення може мати 24 біти на піксель.

Недоліки алгоритму:

- Розмиття зображення (втрата чіткості);
- При підвищенні ступеня стиснення зображення розпадається окремі квадрати (8x8), відбуваються великі втрати при квантуванні як на високих, так і на низьких частотах. Тому на межах блоків помітні невідповідності в яскравості та кольорі великих деталей (Рис. 1.16).



Рисунок 1.16 – Розпад зображення на квадрати (8x8) [5]

- Виявляється ефект Гіббса - ореоли по межах різких переходів яскравості та кольору. Виявляється внаслідок різкого обмеження спектра внаслідок обнулення коефіцієнтів ДКП під час квантування. На (Рис. 1.17) показаний прямокутний сигнал і результат його подання обмеженою кількістю гармонік, що призводить до коливальних процесів на фронтах.

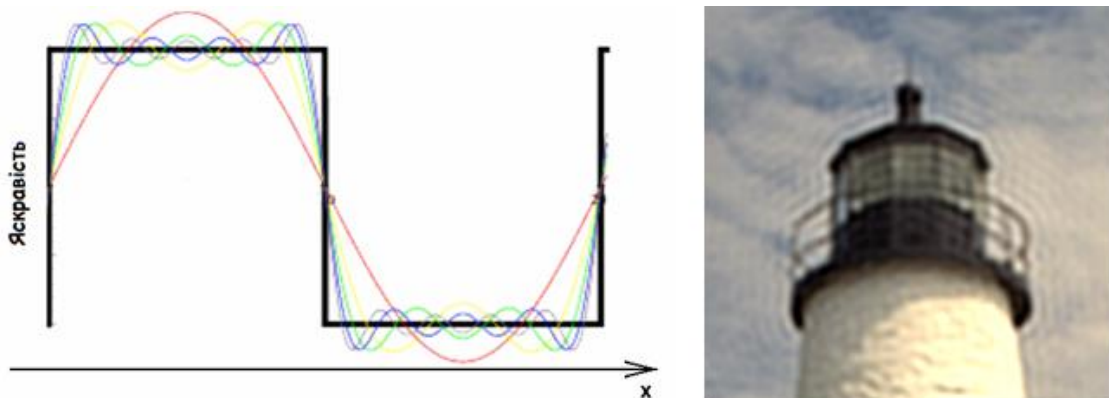


Рисунок 1.17 – Ефект Гіббса [5]

Особливо помітним є ефект Гіббса в комп'ютерній графіці – логотипи, титри, діаграми, графіки тощо.

Інші характеристики алгоритму JPEG:

- Коефіцієнти компресії: 2-200 (задається користувачем);
- Клас зображень: Повнокольорові 24-бітові зображення або зображення в градаціях сірого без різких переходів кольорів (фотографії);
- Симетричність (час архівації/час розархівування) 1;
- Характерні особливості: у деяких випадках алгоритм створює “ореол”

навколо різких кордонів у зображенні (ефект Гіббса). При високій мірі стиснення зображення розпадається на блоки 8x8 пікселів.

1.3 Дослідження внутрішньокадрового стиснення

Обчислення розміру досліджуваних файлів

$$B = N_v \times N_h \times M / 1024 / 8 = 512 \times 512 \times 24 / 1024 / 8 = 768 \text{ kB}$$

Таблиця 1.1 – Дослідження методів стиснення

Метод	Lena	Town	Text
Zip	До: 768 kB Після: 708 kB К= 1,08	До: 768 kB Після: 587 kB К= 1,31	До: 768 kB Після: 19,5 kB К= 39,4
Rar	До: 768 kB Після: 402 kB К= 1,91	До: 768 kB Після: 482 kB К= 1,59	До: 768 kB Після: 18,8 kB К= 40,9
Jpeg-0	20,4 kB	31,3 kB	37,4 kB
Jpeg-20	23,7 kB	36,1 kB	41,8 kB
Jpeg-40	28,6 kB	42,9 kB	48,0 kB
Jpeg-60	37,1 kB	54,4 kB	57,5 kB
Jpeg-80	57,9 kB	80,9 kB	76,6 kB
Jpeg-100	203 kB	217 kB	153 kB
	Коефіцієнти стиснення		
К Jpeg-0	37,65	24,54	20,53
К Jpeg-20	32,41	21,27	18,37
К Jpeg-40	26,85	17,90	16,00
К Jpeg-60	20,70	14,12	13,36
К Jpeg-80	13,26	9,49	10,03
К Jpeg-100	3,78	3,54	5,02

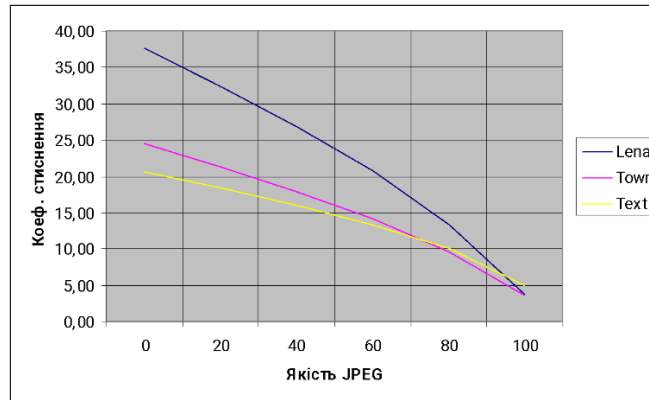


Рисунок 1.18 – Залежність коефіцієнта стиснення від налаштувань якості кодека JPEG для різних зображень [5]

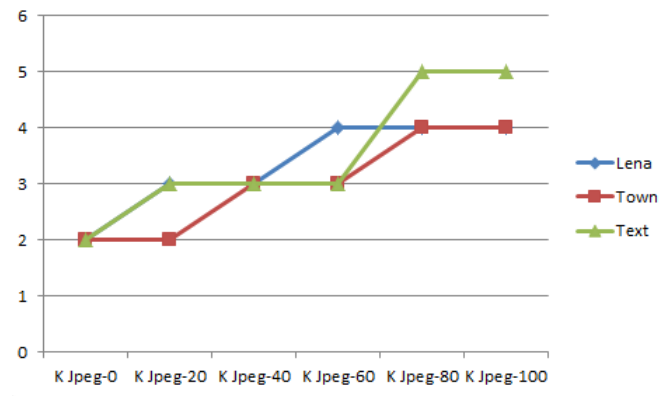


Рисунок 1.19 – Залежність візуальної якості зображення від налаштувань якості кодека JPEG для різних зображень [5]

Висновки по розділу: у цьому розділі були розглянуті декілька видів надмірності інформації, такі як: статистична, психовізуальна та структурна. Їх методи зменшення надмірності. Статистична передбачає наступний кадр на основі попереднього та теперішнього. Психовізуальна зменшує обсяг переданих даних шляхом зменшення кольорової смуги. Далі було розглянуто внутрішньокадрове стиснення, його алгоритм, переваги та недоліки. На основі цих даних було проведено дослідження стиснення даних. Напівтонові зображення погано стискаються методами Rar і Zip, тому що містять мало повторюваної інформації. Сусідні пікселі в них, хоч трохи, але відрізняються. Це зменшує ефективність стиснення без втрат. Зображення тексту містить великі однорідні області, тому ефективність стиснення висока.

2 ЕНКОДЕРИ І ЇХ ВИДИ

Енкодер - найважливіший компонент в процесі перетворення аудіо та відеосигналів у цифровий формат, придатний для зберігання, потокової передачі. Вони відіграють важливу роль у різних сценаріях - від створення мультимедійного контенту до відеоконференцій і прямих трансляцій. Кодувальники використовують алгоритми для стиснення і кодування необроблених аудіо та відеоданих, зменшуючи розмір файлів при збереженні прийнятної якості [4].

Існує два основних типи енкодерів:

- Програмні кодувальники - це програми, які виконують процес кодування на центральних процесорах. Вони використовують обчислювальну потужність центрального процесора для перетворення необроблених аудіо та відеоданих у стислі цифрові формати.

Приклади програмних кодерів:

-x264 (H.264): Широко розповсюджений та ефективний алгоритм кодування відео. Це програмний кодер, що використовує ресурси центрального процесора для стиснення відео в реальному часі.

-VP9: Розроблений компанією Google стандарт стиснення відео VP9 використовується в програмних кодерах, забезпечуючи ефективне стиснення потокового та онлайн-відео.

- Апаратні кодувальники - це спеціалізовані процесори, які розроблені для кодування відео та аудіо. Вони розвантажують основний процесор від процесу кодування, забезпечуючи підвищену ефективність і продуктивність.

Приклади апаратних кодувальників:

- NVENC (Nvidia Encoder): Вбудований у графічні процесори Nvidia апаратний кодер, який розвантажує задачі кодування відео на відеокарту, знижуючи

навантаження на центральний процесор і підвищуючи загальну продуктивність системи.

- Quick Sync Video: Розроблений Intel, Quick Sync Video апаратний кодер, інтегрований у процесори Intel, призначений для прискорення завдань кодування відео та підвищення продуктивності мультимедіа.

-AMF/VCE (Advanced Media Framework/Video Coding Engine): Інтегровані в графічні процесори AMD, AMF і VCE апаратні кодери, які забезпечують ефективне кодування відео.

Висновки по розділу: кодувальники, як програмні, так і апаратні, є невід'ємними компонентами цифрового мультимедійного ландшафту. Вибір між програмним і апаратним кодуванням часто залежить від таких факторів, як специфікації системи, вимоги до продуктивності та конкретний випадок використання. Кожен тип кодувальника має свої переваги та недоліки, і вибір відповідного кодувальника має вирішальне значення для досягнення оптимальних результатів у різних мультимедійних додатках.

3 КОМПЛЕКТУЮЧІ КОМП'ЮТЕРА ДЛЯ ПРЯМОЇ ТРАНСЛЯЦІЇ

3.1 Вибір комплектуючих

Під час складання конфігурації комп'ютера для стрімінгу, важливо зосередитися на виборі процесора та відеокарти, оскільки вони відіграють ключову роль у процесі кодування відео, залежно від обраного енкодера.

Для досягнення мінімально прийнятної якості зображення при кодуванні з процесора, варто вибрати процесор із 6 ядрами, не старший за Ryzen покоління Zen 2 або Intel 10-го покоління.

Універсальним вибором відеокарти для кодування є Nvidia з кодувальником не старіше 6-го покоління [6], наприклад, GTX 1060 і новіше. Це обґрунтовано їхньою здатністю зберігати високу якість зображення як у статичних, так і у динамічних сценах. Енкодери від AMD та Intel, хоч і менш ефективні, але вони можуть використовуватися для менш динамічних сцен, наприклад, трансляції лекції.

Окрему увагу слід приділити вибору накопичувача, на якому розташовуватиметься система, відео, зображення, програми та ігри, що використовуватимуться під час трансляції. SSD є найкращим вибором для уникнення затримок при завантаженні цих файлів. Також не варто зберігати такі файли на віддаленому сервері.

Щодо оперативної пам'яті, рекомендується мати щонайменше 16 ГБ, оскільки під час трансляції будуть використовуватися ігри, додаткові плагіни та браузер, а всі додаткові джерела браузера в OBS також займають своє місце в оперативній пам'яті.

Інші комплектуючі вибираєте залежно від потреб процесора, відеокарти та бюджету. Важливо обирати компоненти, які відповідають потребам системи для забезпечення стабільності та високої якості стрімінгу.

3.2 Комплектуючі комп'ютера на якому проводилось дослідження

CPU: Ryzen 5 5600x

GPU: Nvidia RTX 3070 (7 покоління кодувальника)

RAM: 32GB

Storage: SSD

Висновки по розділу: у цьому розділі було наголошено на ключових аспектах, що визначають успішність процесу стрімінгу, а саме виборі комп'ютерних компонентів для прямої трансляції необхідно знайти баланс між CPU, GPU і оперативною пам'яттю, щоб створити гарну систему для прямих трансляцій. Віддайте перевагу компонентам з підтримкою апаратного кодування, достатньою обчислювальною потужністю та достатнім обсягом пам'яті, щоб забезпечити безперебійну та якісну роботу з прямими трансляціями. Крім того, постійне оновлення новітніх технологій і врахування майбутніх можливостей масштабування можуть сприяти створенню більш надійної системи потокового мовлення.

4 НАЛАШТУВАННЯ OBS

4.1 Підключення акаунту

Спочатку необхідно налаштувати підключення облікового запису Twitch або YouTube. Для цього перейдіть до вікна налаштувань та оберіть розділ "Stream" (Рис. 4.1). У випадку Twitch введіть свій логін та пароль або вкажіть сервер та ключ трансляції. Для YouTube треба вказати сервер та ключ трансляції.

Після успішного підключення облікового запису програма надасть можливість відкрити додаткові вікна, такі як чат та налаштування трансляції. Тут можна задати назву трансляції, вибрати його категорію, а також додаткові теги для опису контенту.

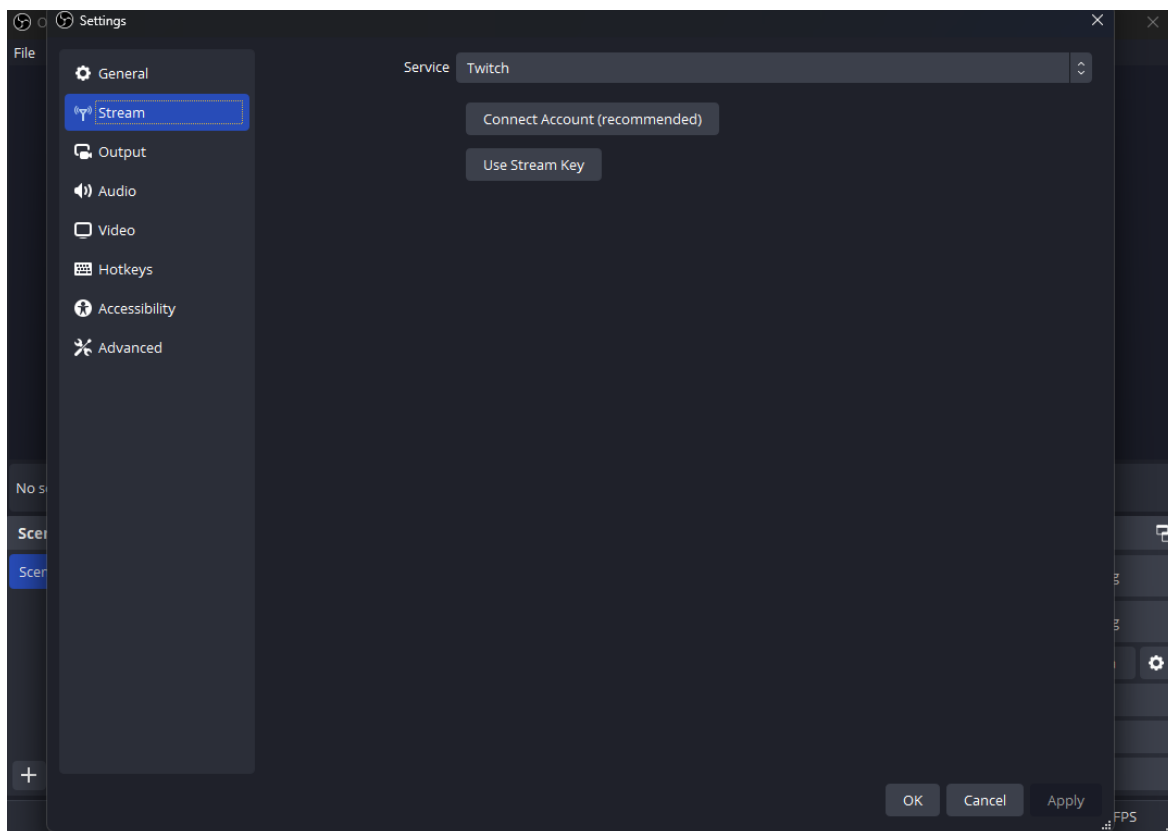


Рисунок 4.1 – Налаштування акаунту

4.2 Роздільна здатність та частота оновлення

Роздільна здатність та частота кадрів (Рис. 4.2) в контексті трансляцій відео є суттєвими аспектами, визначаючими якість та плавність відтворення. Роздільна здатність визначає кількість пікселів на екрані, в той час як частота кадрів (FPS) вказує на кількість кадрів, які відправляються за секунду. Високі значення роздільної здатності сприяють чіткому відображенню на екрані, тоді як збільшення FPS призводить до більшої плавності відтворення. Однак збільшення роздільної здатності та частоти кадрів збільшує обсяг даних, які потрібно передати, що може вимагати збільшення бітрейту.

Рекомендується використовувати роздільну здатність 1920x1080 для досягнення високої якості відображення. Однак, за відсутності достатньо потужного ПК, прийнятним варіантом може бути вибір меншої роздільної здатності, такої як 1280x720 або 1664x936. Особливо це стосується динамічного контенту, який транслюється на платформі Twitch із бітрейтом 6000 кб/с або менше [1].

Важливо зазначити, що при виборі високої роздільної здатності та FPS необхідно дотримуватись відповідного бітрейту, оскільки недостатній бітрейт може спричинити розмиття та фрагментацію зображення. Оптимальний баланс між роздільною здатністю, FPS та бітрейтом дозволяє досягти оптимальної якості.

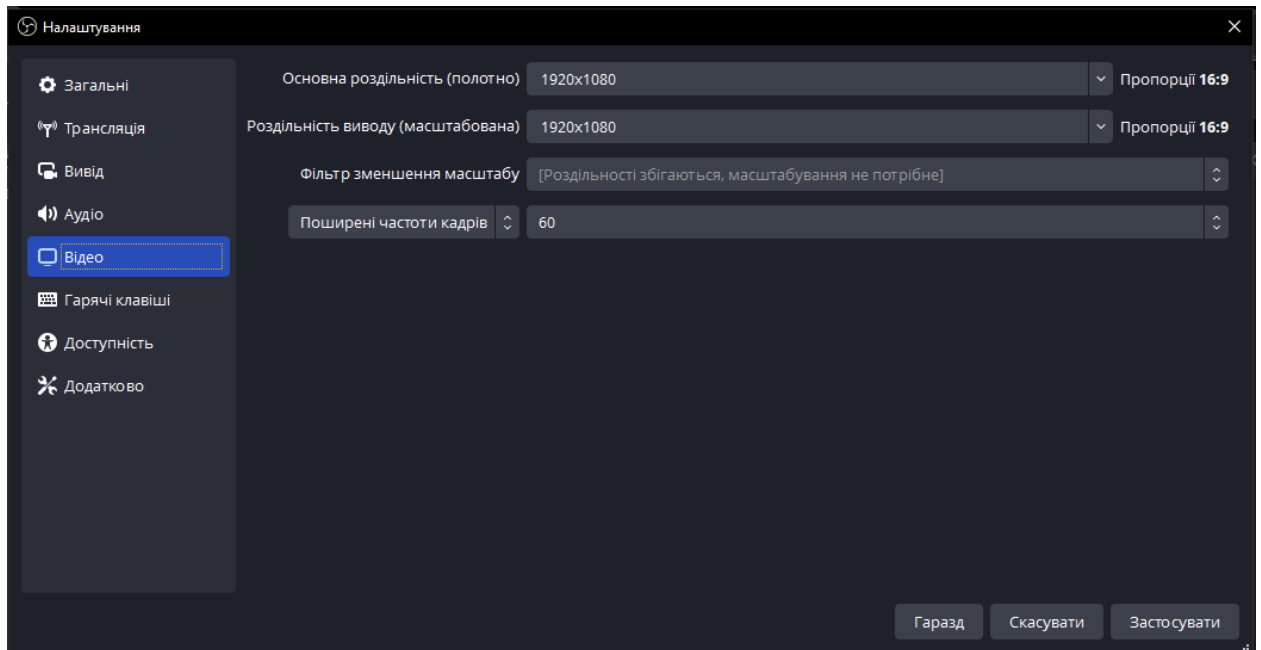


Рисунок 4.2 – Налаштування роздільної здатності та частоти оновлення

4.3 Вхідний бітрейт і їх види

Бітрейт це кількість бітів, які передаються за одиницю часу, і визначає обсяг інформації, який несе кожен кадр відео. Чим вищий бітрейт, тим більше пропускної спроможності мережі вимагається для безперервної передачі цих даних.

У випадку недостатньо низького бітрейту для обраної роздільної здатності та частоти кадрів може виникнути деградація якості зображення. З іншого боку, надто високий бітрейт може породжувати труднощі, такі як перепустка кадрів та проблеми з буферизацією. Це може впливати на стійкість трансляції, спричиняючи розриви в передачі даних та перевантаження буферів, що може призвести до нестабільного відтворення контенту.

Отже, у виборі оптимального бітрейту слід дотримуватися балансу між якістю відео та вимогами до пропускної здатності мережі, зокрема враховуючи роздільну здатність та частоту кадрів.

Види бітрейта:

- Постійний бітрейт (CBR)

При постійному бітрейті виділяється задана кількість біт для кодування незалежно від складності відеофрагменту. Автоматичне регулювання ступеня стиснення використовується для того, щоб відповідати зазначеному постійному бітрейту. Це забезпечує сталу якість, але може бути неефективним у випадках, коли складність відеофрагментів значно варіюється.

- Змінний бітрейт (VBR)

Принципова відмінність змінного бітрейту від постійного полягає в динамічному розподілі біт кодування в залежності від складності відеофрагментів. При цьому більшу частину часу бітрейт відповідатиме заданому цільовому значенню. Проблема VBR у тому, що коли анімації стають менш динамічними, наприклад, якщо ставите гру на паузу, VBR значно знижує бітрейт. У динамічних сценах, наприклад, під час бою бітрейт знову різко зростає, що може спричинити проблеми на стороні кінцевого користувача, таких як пропуск кадрів або буферизацію трансляції. Для будь яких трансляцій слід використовувати CBR бітрейт.

Для оптимальної якості на платформі Twitch, рекомендовані значення бітрейту починаються з 1500 кб/с для формату 360p30fps і можуть зростати до 8000 кб/с для формату 1080p60fps, якщо маєте статус партнера Twitch. У випадках інших користувачів існує обмеження в 6000 кб/с [1].

Для YouTube, рекомендовані значення бітрейту становлять не менше 1000 кб/с для формату 360p30fps і можуть бути збільшені до 51000 кб/с для формату 4320p [2].

Такий підхід допомагає забезпечити оптимальний баланс між якістю відео та вимогами платформи щодо обмеження бітрейту.

4.4 Енкодери і їх налаштування

Еncoder – це програмне забезпечення, що виконує функцію кодування цифрового відеопотоку, що передбачає стиснення з втратами інформації. Процес кодування аудіо та відео вимагає використання спеціалізованих методів стиснення, які враховують різні аспекти, такі як якість відтворення відео, обсяг інформації (бітрейт), складність алгоритмів кодування, стійкість до втрат даних та помилок, зручність редагування, довільний доступ до вмісту, якість використовуваних алгоритмів, затримка в каналах зв'язку і інші фактори.

Типи еncoderів та їх налаштування:

- x264 використовує обчислювальні потужності процесора для виконання кодування відеопотоку. У даного кодувальника існує декілька шаблонів налаштувань якості кодування, відомих як пресети, які варіюються від "ultrafast" до "placebo". Простими словами, чим повільніший пресет, тим більше витрат на обчислювальні ресурси ЦП.

У випадку повільних пресетів ЦП витрачає більше часу на кодування відео, щоб досягти вищого рівня якості. Однак це також означає, що у процесора залишається менше ресурсів для виконання інших завдань, оскільки вся потужність спрямовується на кодування відео. У випадку швидких пресетів ЦП витрачає менше часу та ресурсів, але це веде до зниження якості зображення. Такий баланс між швидкістю кодування та якістю визначається вибором конкретного пресету в залежності від вимог до якості та доступних ресурсів процесора.

Налаштування еncoderу x264 (Рис. 4.3):

Тип бітрейту:

- CBR;
- VBR.

Бітрейт:

- Від 0кб\с до 51000кб\с.

Інтервал ключових кадрів:

- Від 0 до 10 секунд.

Пресет:

- Ultrafast;

- Superfast;

- Veryfast;

- Faster;

- Fast;

- Medium;

- Slow;

- Slower;

- Veryslow;

- Placebo.

Профіль:

- Baseline;

- Main;

- High.

Налаштування:

- film;

- animation;

- grain;

- stillimage;

- psnr;

- ssim;

- fastdecode;

- zerolatency.

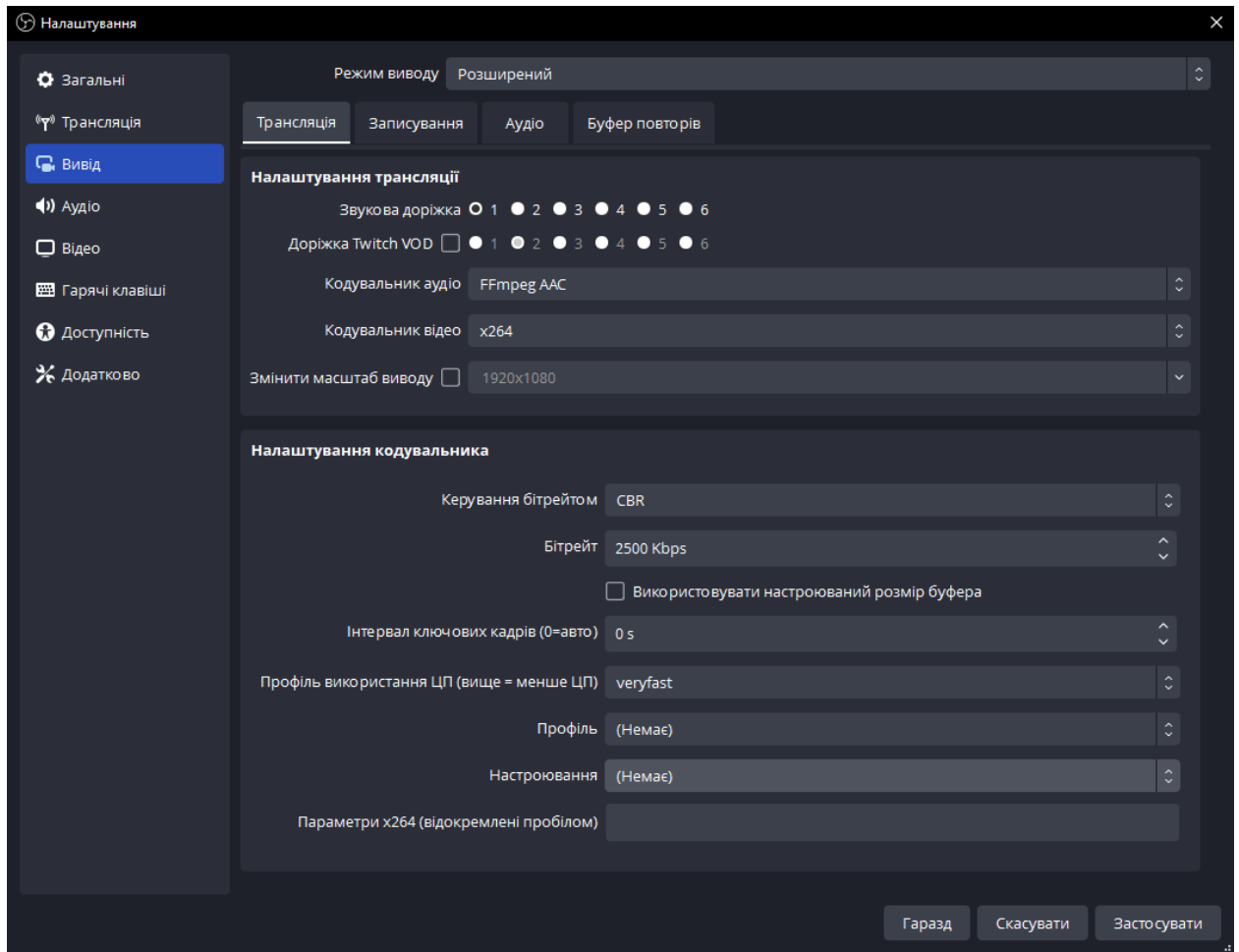


Рисунок 4.3 – Налаштування енкодеру X264

- H.264 це енкодер, який використовує вбудований чіп на відеокарті, що дозволяє кодувати якісне зображення порівнянне за якістю кодування на процесорі з пресетом x264 slow. Така можливість існує для відеокарт, таких як Nvidia, AMD або Intel. Цей енкодер має кілька передвстановлених налаштувань, які можуть варіюватися від надшвидкого до надповільного. Ці налаштування визначають ступінь використання цього чіпу та графічного процесора впливаючи на якість зображення.

Налаштування енкодеру NVENC (Рис. 4.4):

Тип бітрейту:

- CBR;
- VBR.

Бітрейт:

- Від 0кб\с до 51000кб\с.

Інтервал ключових кадрів:

- Від 0 до 10 секунд.

Пресет:

- Fastest;

- Faster;

- Fast;

- Medium;

- Slow;

- Slower;

- Slowest.

Налаштування:

- Висока якість;

- Низька затримка;

- Наднизька затримка.

Багатопрохідний режим:

- Один прохід;

- Два проходи (чверть роздільної здатності);

- Два проходи (повна роздільна здатність).

Профіль:

- High;

- Main;

- Baseline.

Передбачення автоматично налаштовує кількість В кадрів (Nvidia) (Рис. 4.4):

- Увімк\Вимкн.

Психовізуальне настроювання підвищує якість зображення особливо в швидких сценах (Nvidia):

- Увімк\Вимкн.

GPU дозволяє обрати потрібну відеокарту якщо у вас їх декілька.

Максимальна кількість В-кадрів задає максимально можливу кількість використаних кадрів для передбачення зображення. Від 0 до 4. Чим більша кількість тим менша завантаженість та гірша якість зображення.

Бітрейт аудіо від 64 кб\с до 320 кб\с.

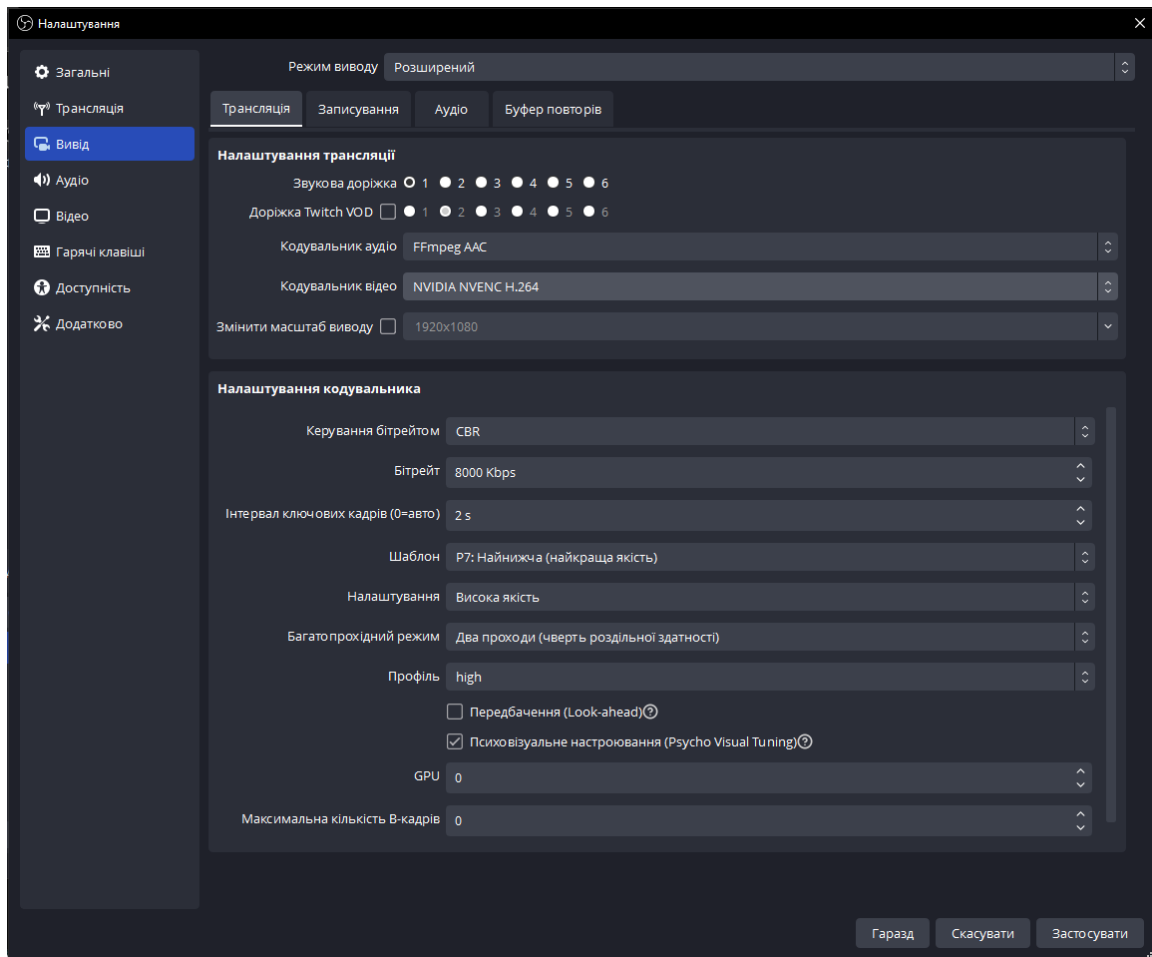


Рисунок 4.4 – Налаштування енкодера NVENC H.264

4.5 Налаштування сцени

На головній сторінці в програмі додаємо джерела (Рис. 4.10) захоплення зображення та звуку для подальшої використання в трансляції (Рис. 4.5). Тут користувач може додати різні джерела, такі як звук, браузерні вікна, веб-камеру, вікно конкретної програми, вміст монітора, зображення, відео та інші.

У конкретному випадку можливість додавання різних джерел включає захоплення звуку з гри, відтворення музики з платформи Spotify, звуку з платформи

Discord та захоплення зображення гри (Рис. 4.6). Додатково, можна встановити плагін для відображення повідомлень про донати.

Для додаткової гнучкості встановлено можливість захоплення всього звуку з комп'ютера та захоплення зображення зі всього монітору але вони будуть вимкнені.

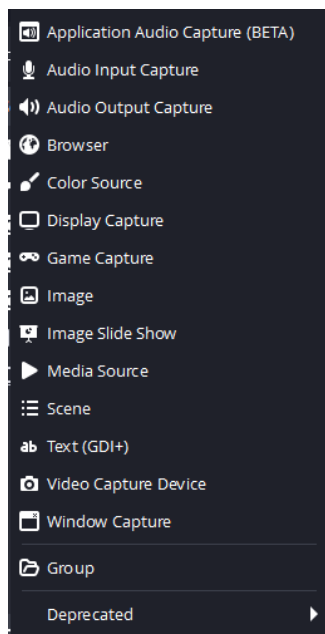


Рисунок. 4.5 – Перелік можливих джерел

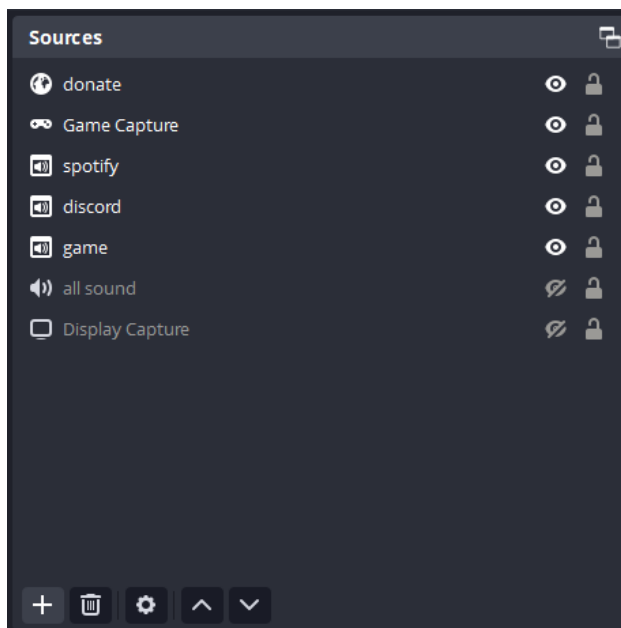


Рисунок. 4.6 – Додані джерела

У розділі аудіо мікшера (Рис. 4.7) продовжимо налаштування звуку. Тут користувач має можливість ретельно налаштувати аудіо відтворення під час трансляції. Серед доступних опцій:

Гучність: Загальний рівень гучності.

Моно або стерео канал: Можна обрати потрібний канал для цієї доріжки. За замовчуванням стерео.

Зміщення синхронізації: Можна обрати зміщення доріжки у мілісекундах, це потрібно якщо звук з якогось пристрою має затримку і потрібно підлаштувати увесь інший звук під нього.

Моніторинг: Тут можна обрати тільки прослуховування, прослуховування та виведення доріжки, або вимкнути моніторинг.

Вибір доріжки: Тут можна вибрати конкретну аудіодоріжку, що транслюється, що дозволяє ефективно розділяти аудіо для подальшого монтажу, або вимкнення певної доріжки на записі трансляції, хоча на самій трансляції звук з неї буде.

Фільтри для мікрофону: Додавання фільтрів дозволяє поліпшити якість аудіосигналу з мікрофону, наприклад, шумопониження, еквайзери або підсилення діапазону частот.

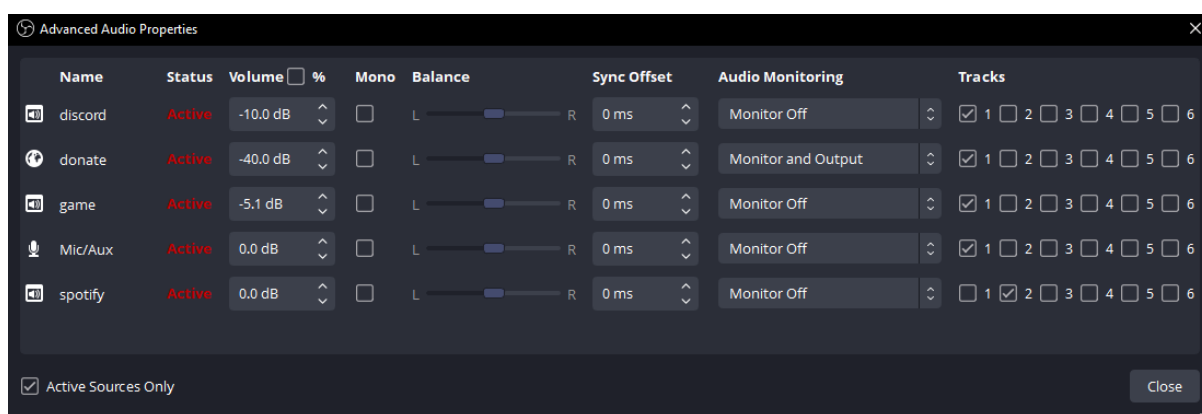


Рисунок. 4.7 – Налаштування звуку

Для себе зміню доріжку для музики, щоб її не було чути на записі трансляції, але було чути на самій трансляції.

Також використаю декілька фільтрів (Рис. 4.8) для мікрофону, а саме: Noise Suppression, Noise Gate, Compressor щоб прибрати зайвий шум та покращити звучання (Рис. 4.9).

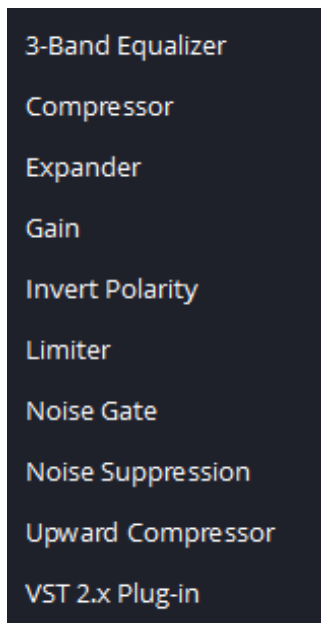


Рисунок. 4.8 – Перелік фільтрів

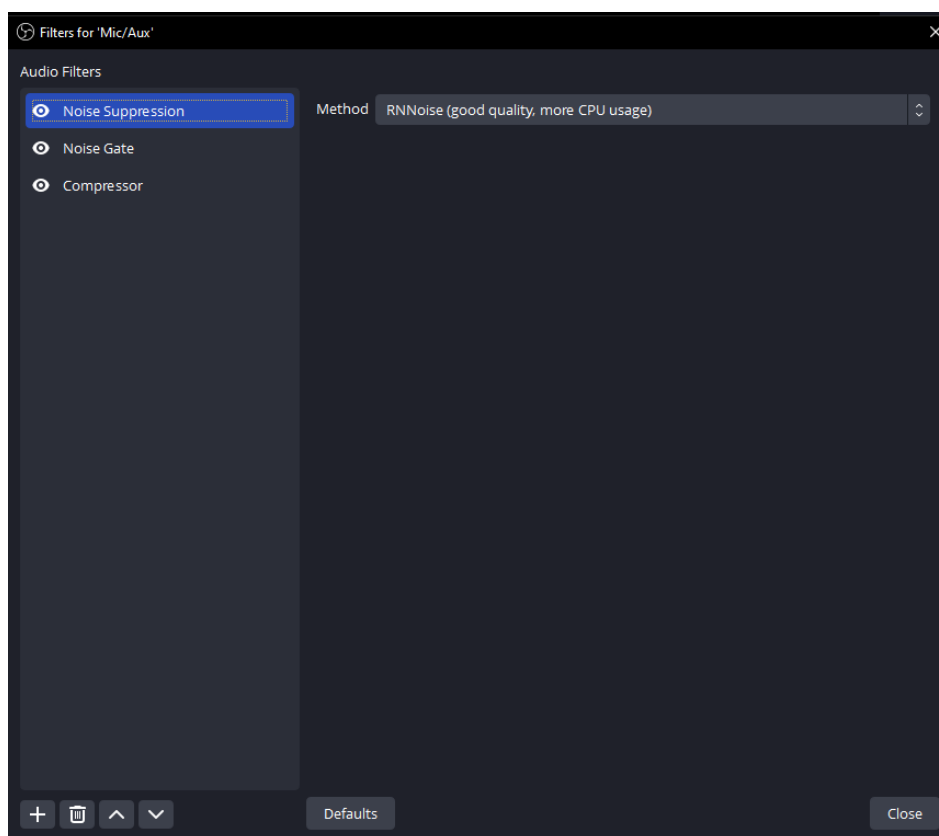


Рисунок. 4.9 – Перелік фільтрів для мікрофону

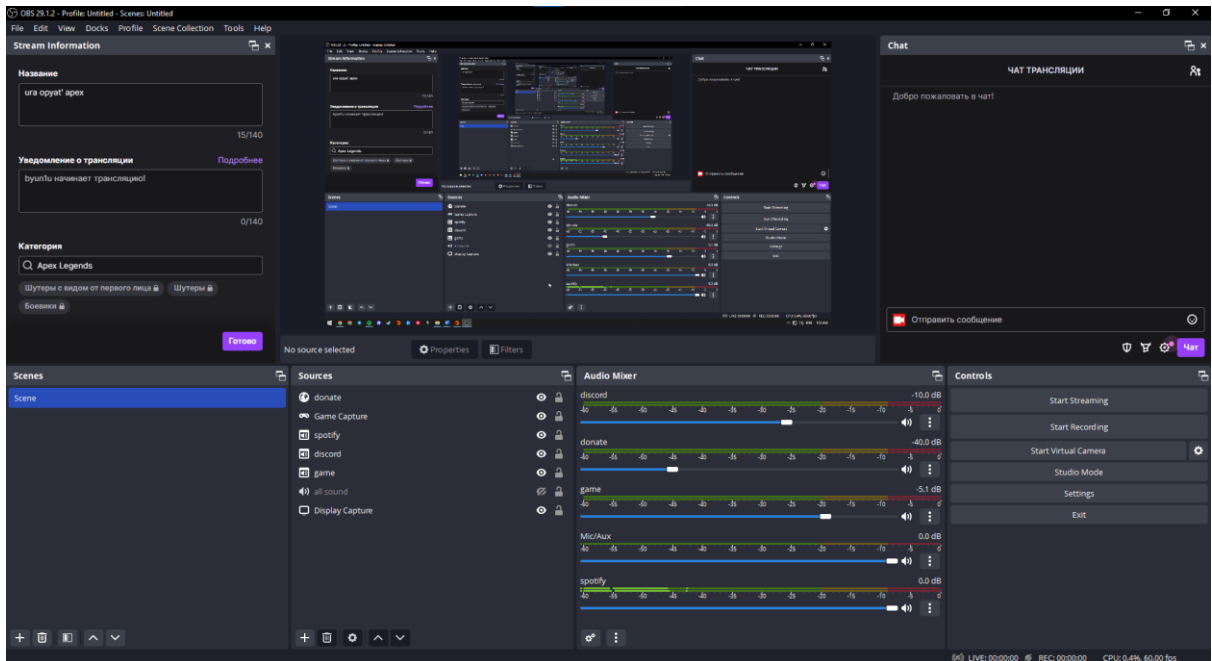


Рисунок 4.10 – Готова програма для проведення прямої трансляції, або запису

Висновки по розділу: у цьому розділі була налаштована програма для проведення прямих трансляцій OBS. А саме були під'єднанні облікові записи Twitch та Youtube, налаштовані енкодера, додані аудіо та відео джерела.

Підсумовуючи всі налаштування можна сказати що це динамічний процес, що вимагає коригування залежно від вашого обладнання та ваших конкретних потреб. Регулярно тестуйте свої конфігурації, оптимізуйте налаштування для підвищення продуктивності та використовуйте додаткові функції для поліпшення загальної якості прямої трансляції. Добре налаштована система OBS дасть змогу надавати аудиторії цікавий і професійний контент у прямому ефірі.

5 ТРАНСЛЯЦІЯ НА ДЕКІЛЬКА СТІМІНГОВИХ ПЛАТФОРМ ОДРАЗУ

Мультиплатформний стрімінг передбачає трансляцію одного прямого потоку на кілька платформ одночасно. Це досягається за допомогою програмного забезпечення для кодування, яке отримує необроблене відео та аудіо, кодує його, а потім надсилає за кількома напрямками в режимі реального часу.

Плюси багатоплатформного стрімінгу:

- Збільшення охоплення аудиторії

Трансляція на кількох платформах одразу дає змогу охопити різноманітну аудиторію на різних потокових сервісах, розширюючи охоплення і потенційно залучаючи нових глядачів.

- Резервування та надійність

Якщо на одній платформі виникнуть технічні проблеми, ваша трансляція залишиться доступною на інших платформах, забезпечуючи стабільне сприйняття глядачами.

Мінуси багатоплатформного стрімінгу:

- Ресурсоємність

Потокове передавання на кілька платформ одночасно може бути ресурсоємним, вимагати надійного інтернет з'єднання, потужного ПК та ефективних налаштувань кодування для підтримання трансляції.

- Обмеження, пов'язані з конкретною платформою

Кожна платформа прямих трансляцій може мати свої унікальні обмеження, такі як максимальний бітрейт, роздільна здатність або особливі вимоги до кодування. Адаптація трансляції для відповідності цим специфікаціям може бути складною.

Існує декілька методів для запуску трансляцій на декілька платформ одразу:

5.1 Сторонні сервіси

Не вимагають будь яких додаткових можливостей від вашого ПК або інтернету. Один із прикладів такого сервісу – ReStream (Рис. 5.1), який вирізняється зручністю та широким функціоналом. Даний сервіс дозволяє безкоштовно підключити два канали для трансляції і розширити цей перелік до 15 каналів, охоплюючи такі платформи як Twitch, YouTube, Telegram, Instagram та інші. Крім того, ReStream забезпечує можливість запису трансляцій безпосередньо у хмарне сховище, об'єднує чат з різних платформ у єдиний, надає статистику та інші корисні функції.

Для запуску трансляції на Twitch і YouTube одночасно за допомогою ReStream, користувачу потрібно увійти в облікові записи Twitch та YouTube, ввести сервер та ключ трансляції у програму OBS, а після цього можна розпочати трансляцію на обрані платформи.

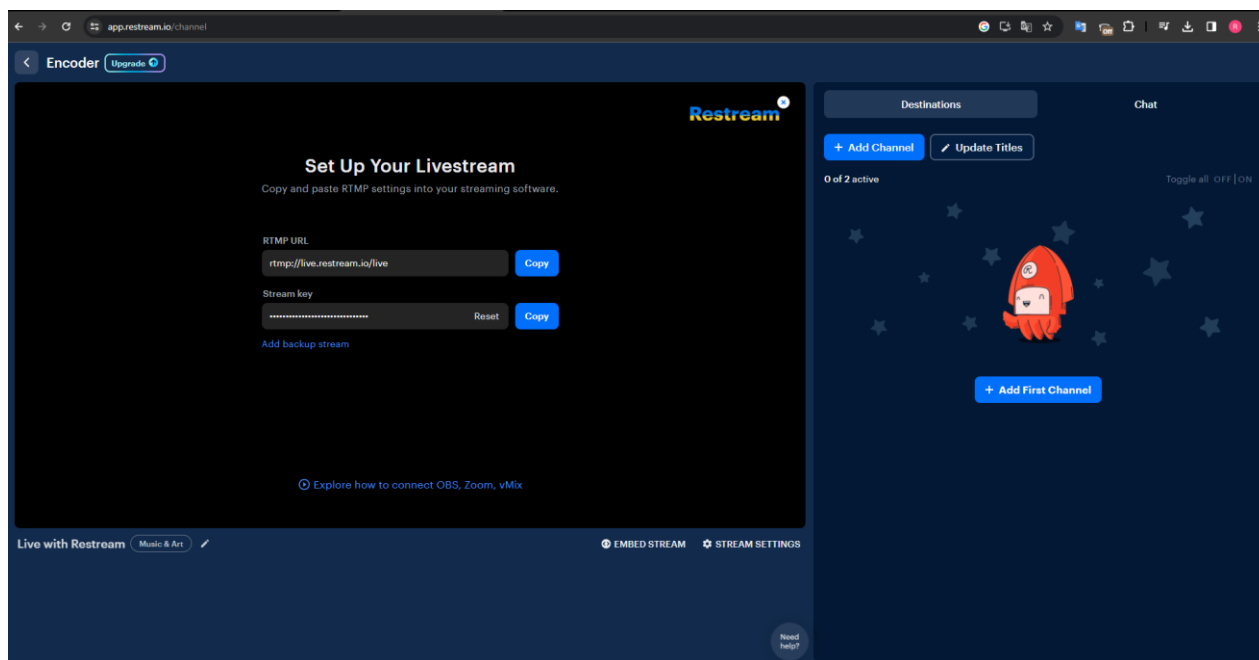


Рисунок 5.1 – Сторінка під'єднання каналів та інформації щодо ключа трансляції

5.2 Плагіни OBS

Потребують збільшену пропускну здатність інтернету. Розглянемо приклад плагіну Multiple RTMP (Рис. 5.2). Цей сервіс безкоштовно дозволяє додавати значну кількість каналів для одночасної трансляції. Для налаштування трансляції на Twitch і YouTube (Рис. 5.3) одночасно за допомогою плагіну Multiple RTMP, користувачу слід ввести сервери і ключі трансляції для кожної платформи окремо. Також доступна можливість налаштування енкодера, сцени для трансляції, звуку для кожної платформи окремо, або використовувати налаштування, які вже були зазначені в OBS.

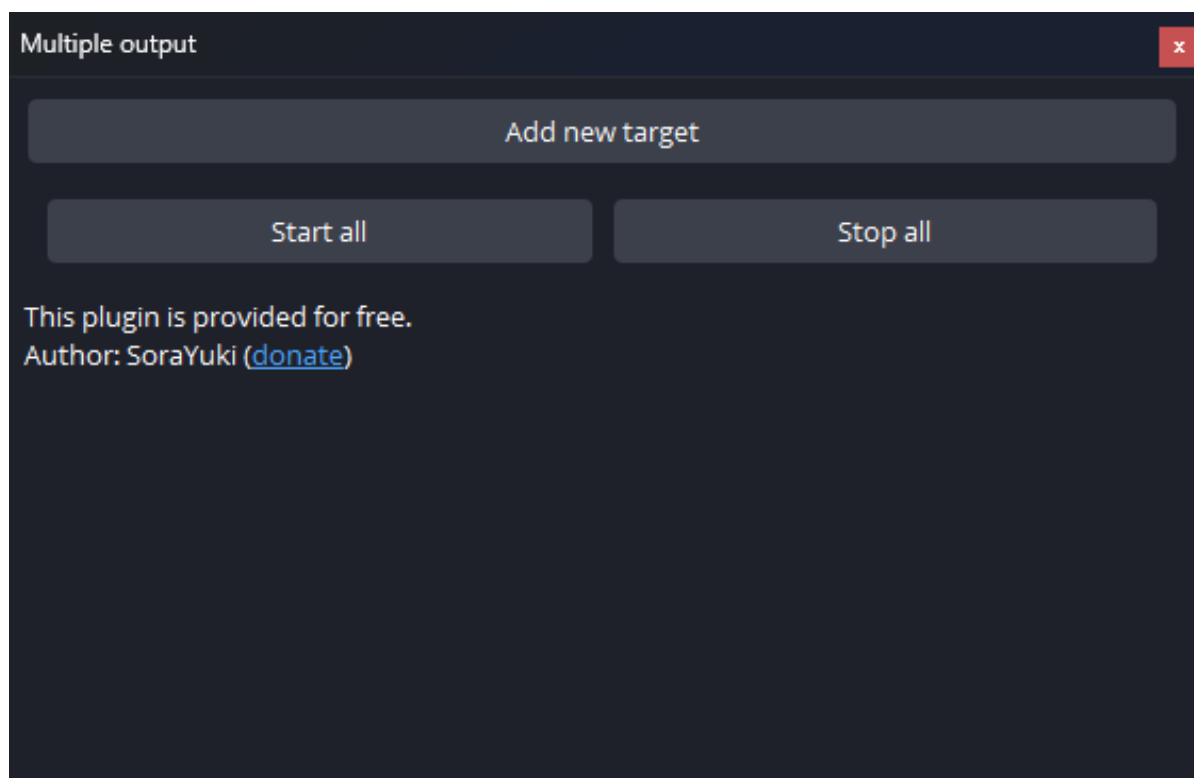


Рисунок 5.2 – Головне вікно плагіну Multiple RTMP

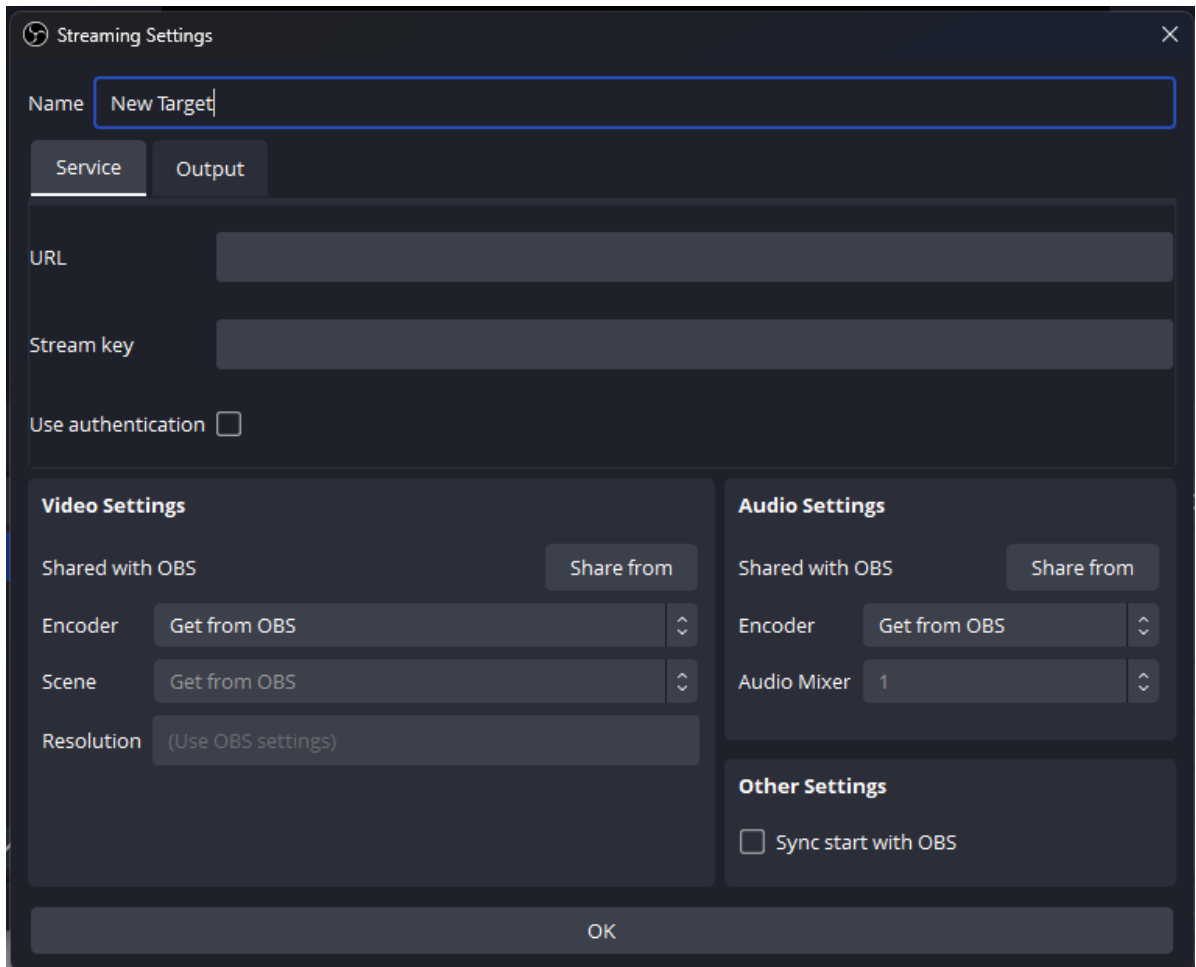


Рисунок 5.3 – Вікно додавання платформи та налаштування трансляції

Висновки по розділу: у цьому розділі було розглянуто декілька методів трансляції одразу на декілька платформ. Треба зазначити що вони відрізняються налаштуваннями, споживанням пропускної здатності інтернету. Також слід зазначити, що багатоплатформна потокова трансляція відкриває широкі можливості для творців контенту, але вимагає уважного ставлення до налаштувань. Використовуючи правильні інструменти та сервіси, можна розкрити потенціал для охоплення ширшої аудиторії та підвищити загальний успіх ваших починань у сфері стримінгу.

6 МЕТОДИКА ТЕСТУВАННЯ ЕНКОДЕРІВ

Головною метою тестування енкодерів є якість зображення при однакових налаштуваннях, оцінка фрагментації зображення, втрати кадрів.

При тестуванні використовувались деякі сталі налаштування, такі як:

Роздільна здатність:

- 1920x1080

Є стандартом для всіх пристроїв.

- 1664x936

Тестується як метод зниження навантаження на комп'ютер, інтернет та зменшення фрагментації зображення в динамічних сценах за рахунок трохи гіршої якості.

- 2560x1440

Вже теж стає стандартом, але це тільки погіршить фрагментацію зображення в динамічних сценах при існуючих можливостях енкодерів. Але воно було протестовано для того, щоб дізнатись чи дійсно YouTube використовує більш ефективний енкодер для цієї роздільної здатності.

Частота кадрів: завжди 60 кадрів на секунду.

Бітрейт:

Для Twitch було обрано два значення, 6000кб\с та 8000кб\с. 6000кб\с вказано як максимальне значення для якості відео 1080р60. 8000кб\с можуть використовувати партнери Twitch для такої ж самої якості.

Для YouTube було обране одне значення бітрейту – 25000кб\с, так як він має обмеження в 50000кб\с що є значно більше ніж у Twitch. 25000кб\с є зайвим для якості 1080р60, але в самий раз для 1440р60 і якихось додаткових переваг для 1080р60 немає.

Енкодери:

-x264 (кодування процесором)

Пресети: Ultrafast як найшвидший, Fast як проміжний, який має непогану якість зображення та Medium який вважається самим оптимальним пресетом.

- NVENC (кодування відеокартою)

Пресети: P1 як найшвидший з самою низькою якістю зображення, P3 як швидкий пресет, але вже з менш вираженою фрагментацією зображення, P5 як майже самий повільний пресет, з гарною якістю та P7 самий повільний пресет з найкращою якістю.

Для тестування було обрано дві платформи та три гри з різною динамічністю, кількістю деталей. По черзі були запуснені тестові відрізки. Потім продивлялися всі відрізки і оцінювалось як той чи інший енкодер з відповідним пресетом впорався з динамічною сценою, з великою кількістю деталей, з поганим освітленням.

Висновки по розділу: У цьому розділі було визначено методику тестування, а саме за якої роздільної здатності, частоти кадрів, бітрейту та яких енкодерах проводилося дослідження кінцевої якості відео. Для дослідження було обрано три гри, тестові відрізки яких були запуснені на різних платформах з зазначеними налаштуваннями.

7 ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ H264

H264 - це програмний кодер, який в покладається на ресурси процесора для кодування відео. Він пропонує високий рівень контролю над налаштуваннями кодування і відомий відмінною якістю відео. Він відмінно зберігає якість сприйняття, особливо в сценаріях зі складними деталями, швидким рухом або швидкими змінами складності сцени.

Мінуси:

- Високе використання процесора: енкодер H264 значною мірою залежить від ресурсів процесора. Потокове передавання з більш високою роздільною здатністю або частотою кадрів може дуже навантажити центральний процесор, що може вплинути на продуктивність системи, випадання кадрів та зниження якості.

Для досягнення задовільної якості зображення рекомендується використовувати хоча б Fast (Рис. 7.2, 7.5, 7.8) пресет. Однак це супроводжується значним навантаженням на процесор. У моєму конкретному випадку використання пресету Fast майже призводило до досягнення максимального рівня завантаженості процесора. Якщо врахувати навантаження при роботі з пресетом Medium (Рис. 7.3, 7.6, 7.9), вже спостерігалася перепустка кадрів.

Цей енкодер рекомендується використовувати у випадку, коли відсутня зовнішня відеокарта, а також в разі трансляції вмісту з невисокою динамікою. Також цей варіант виправданий, якщо в наявності є другий комп'ютер, який може бути використаний виключно для кодування трансляції, розвантажуючи основний комп'ютер від зайвих завдань.



Рисунок 7.1 – Приклад зображення з енкодера x264 ultrafast

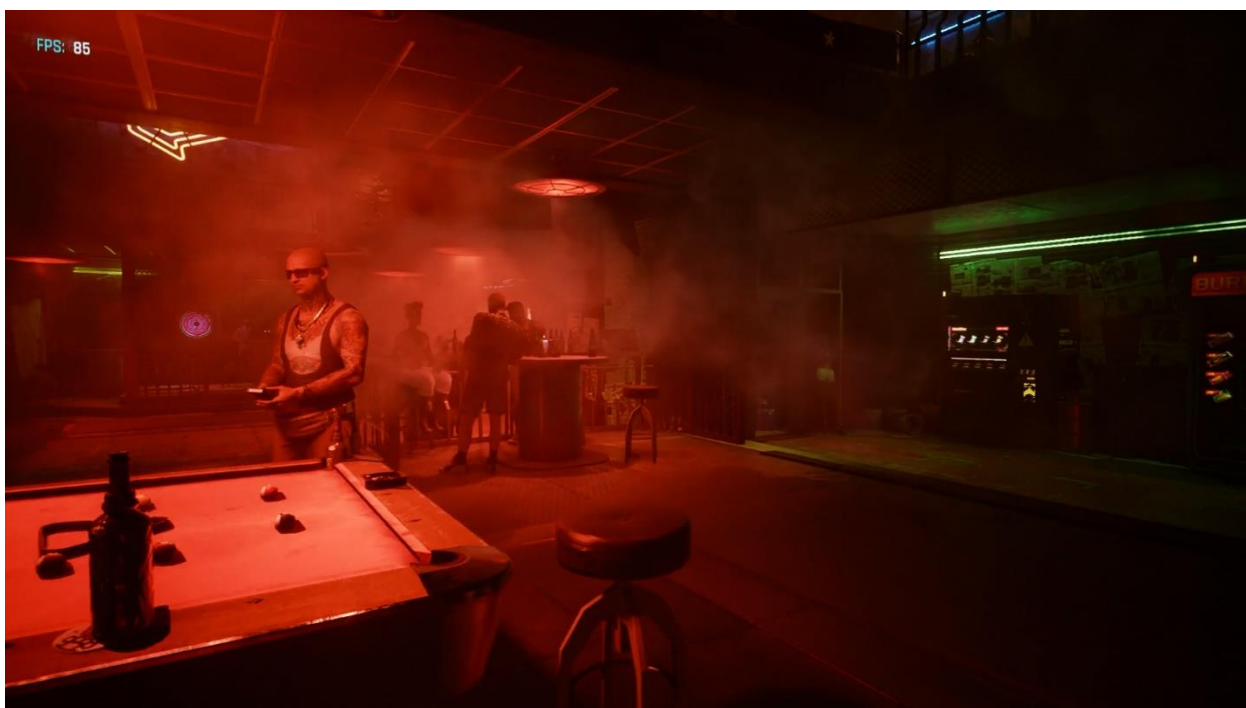


Рисунок 7.2 – Приклад зображення з енкодера X264 fast



Рисунок 7.3 – Приклад зображення з енкодера X264 medium

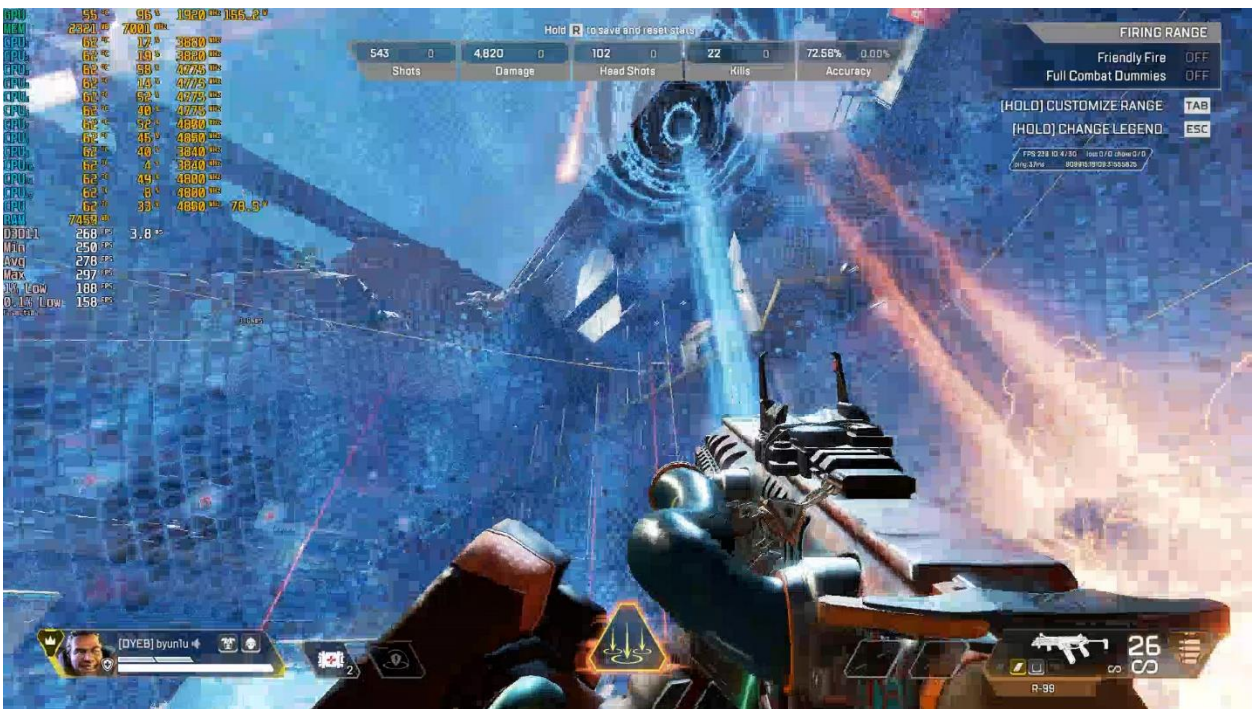


Рисунок 7.4 – Приклад зображення з енкодера x264 ultrafast



Рисунок 7.7 – Пример изображения с энкодера x264 ultrafast



Рисунок 7.8 – Пример изображения с энкодера x264 fast



Рисунок 7.9 – Приклад зображення з енкодера x264 medium

Висновки по розділу: у цьому розділі було дослідження енкодера x264 та порівняння його пресетів. Він відзначається певними особливостями, які слід враховувати при налаштуванні трансляцій. Переважаючою характеристикою є висока вимогливість до процесору, що призводить до потреби ретельно обирати пресети для досягнення оптимальної якості та продуктивності.

Рекомендується використовувати x264 у випадках, коли відсутня зовнішня відеокарта, і контент має невелику динаміку, також якщо доступний другий комп'ютер, що може використовуватися для кодування, тим самим розвантажуючи основний комп'ютер.

8 ПОРІВНЯННЯ ЯКОСТІ ПРЕСЕТІВ ЕНКОДЕРУ NVENC H.264

NVENC - це апаратний кодер, вбудований у графічні процесори Nvidia. Він перекладає завдання кодування відео з CPU на GPU, що забезпечує ефективне кодування з мінімальним впливом на продуктивність CPU. Він забезпечує низьку затримку кодування, що робить його добре придатним прямих трансляцій. Також він чудово справляється з кодуванням декількох відеопотоків одночасно, забезпечуючи чудову продуктивність у багатопотокових сценаріях.

NVENC є високоефективним кодувальником, який демонструє ефективність навіть на найшвидшому пресеті P1 (Рис. 8.1, 8.7, 8.13), але хоч у даному пресеті спостерігається висока якість зображення, можлива наявність фрагментації зображення. Починаючи з пресету P3 (Рис. 8.2, 8.8, 8.14) є деяка невизначеність у якості зображення. P5-P7 (Рис. 8.3, 8.4, 8.5, 8.6, 8.9, 8.10, 8.11, 8.12, 8.15, 8.16, 8.17, 8.18) вже мають вищу якість зображення, фрагментація може бути тільки у динамічних сценах з високою кількістю деталей.

Для користувачів із слабкими відеокартами рекомендується використовувати пресети P1-P3, у той час якщо є більш потужна відеокарта можна вибирати між пресетами P4-P5. У випадку флагманських відеокарт можна з комфортом застосовувати пресети P6-P7. Важливо відзначити, що пресет P7 практично не відрізняється за якістю зображення від пресету P6, тож його використання може бути необов'язковим.

Додатково слід підкреслити, що кодування за допомогою відеокарти мінімально впливає на продуктивність самої відеокарти, втрати продуктивності на рівні 5-15%, що залежить від обраного пресету та складності сцени. Це робить його оптимальним вибором для трансляцій при гарній якості зображення та обмежених можливостях.



Рисунок 8.1 – Приклад зображення з енкодера NVENC P1



Рисунок 8.2 – Приклад зображення з енкодера NVENC P3



Рисунок 8.3 – Приклад зображення з енкодера NVENC P5

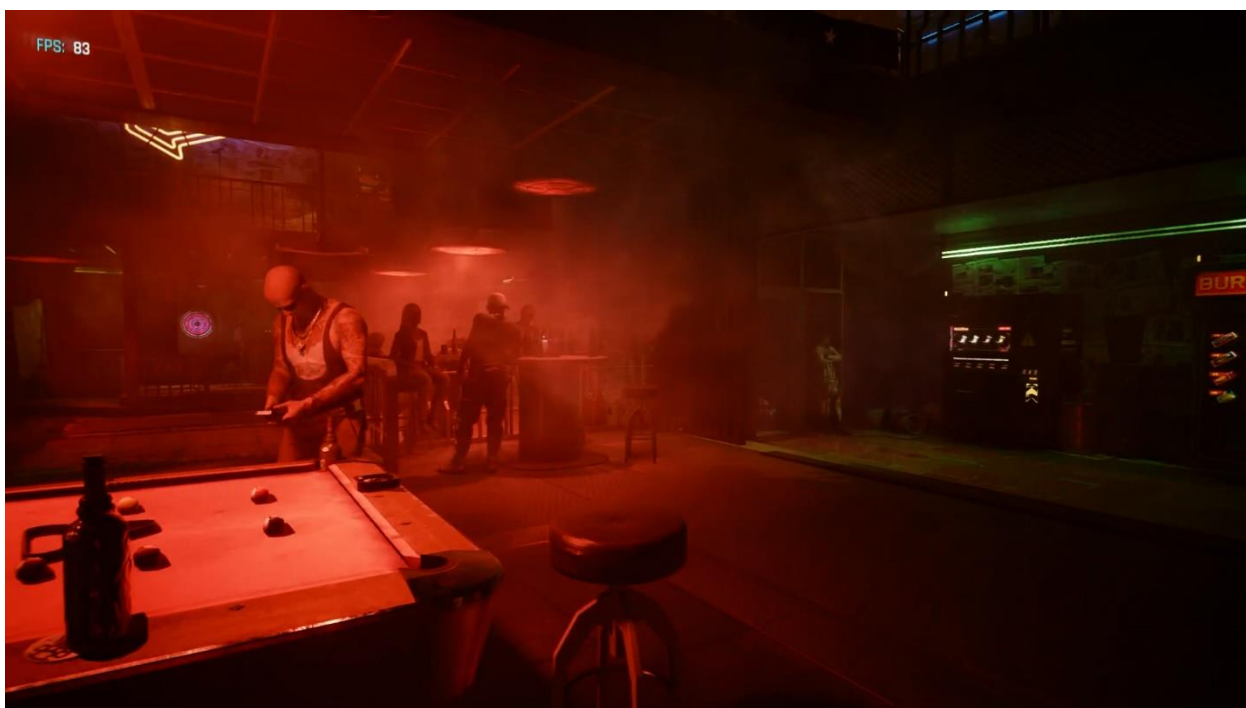


Рисунок 8.4 – Приклад зображення з енкодера NVENC P7



Рисунок 8.5 – Приклад зображення з енкодера NVENC P7 8000кб\с



Рисунок 8.6 – Приклад зображення з енкодера NVENC P7 8000кб\с 936р



Рисунок 8.7 – Приклад зображення з енкодера NVENC P1

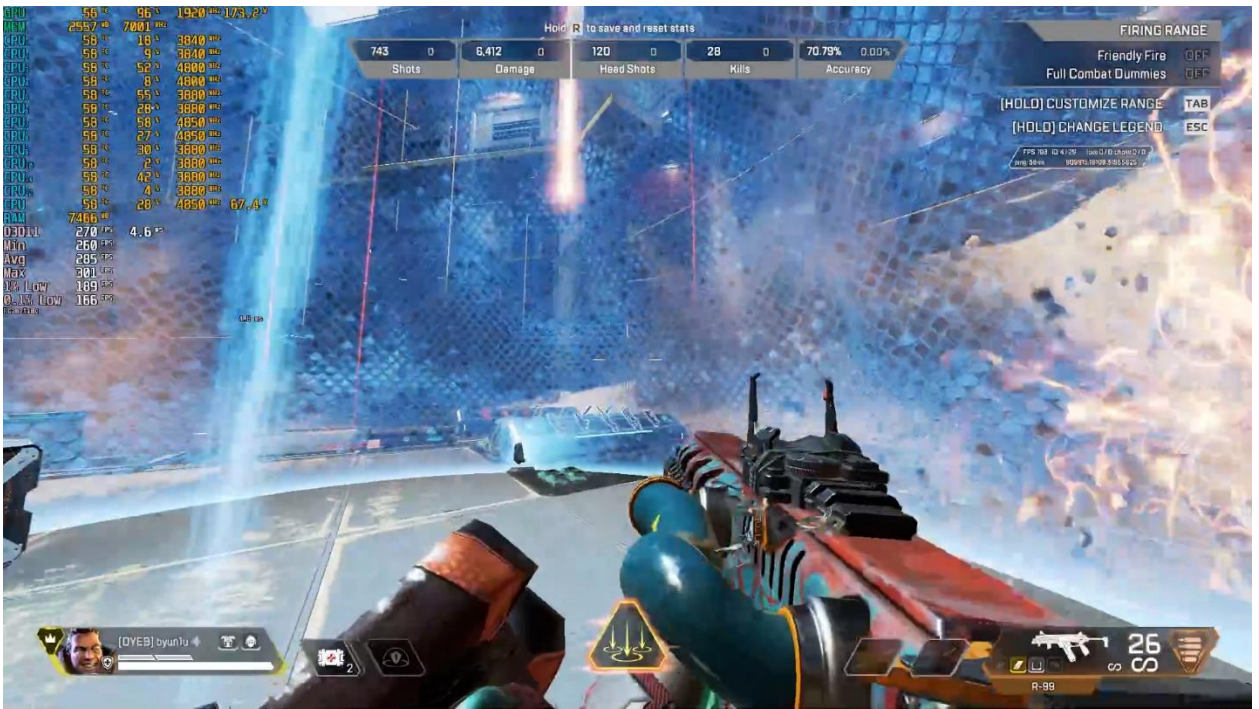


Рисунок 8.8 – Приклад зображення з енкодера NVENC P3

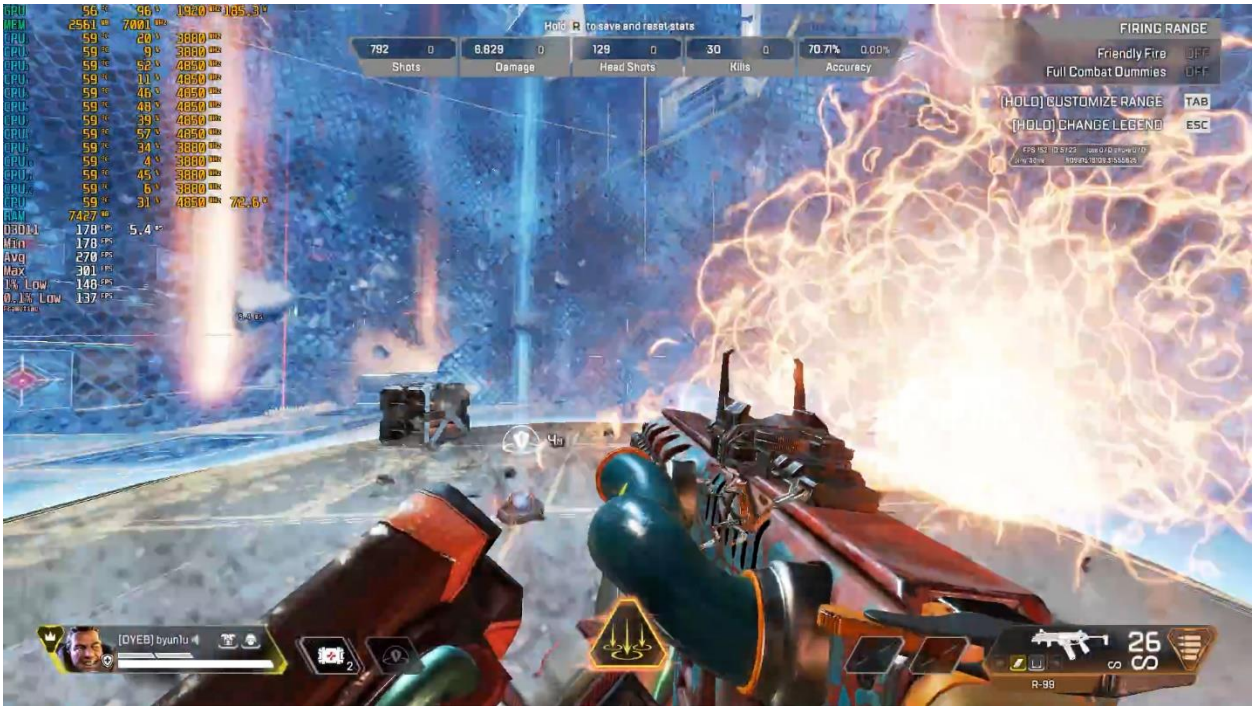


Рисунок 8.9 – Приклад зображення з енкодера NVENC P5

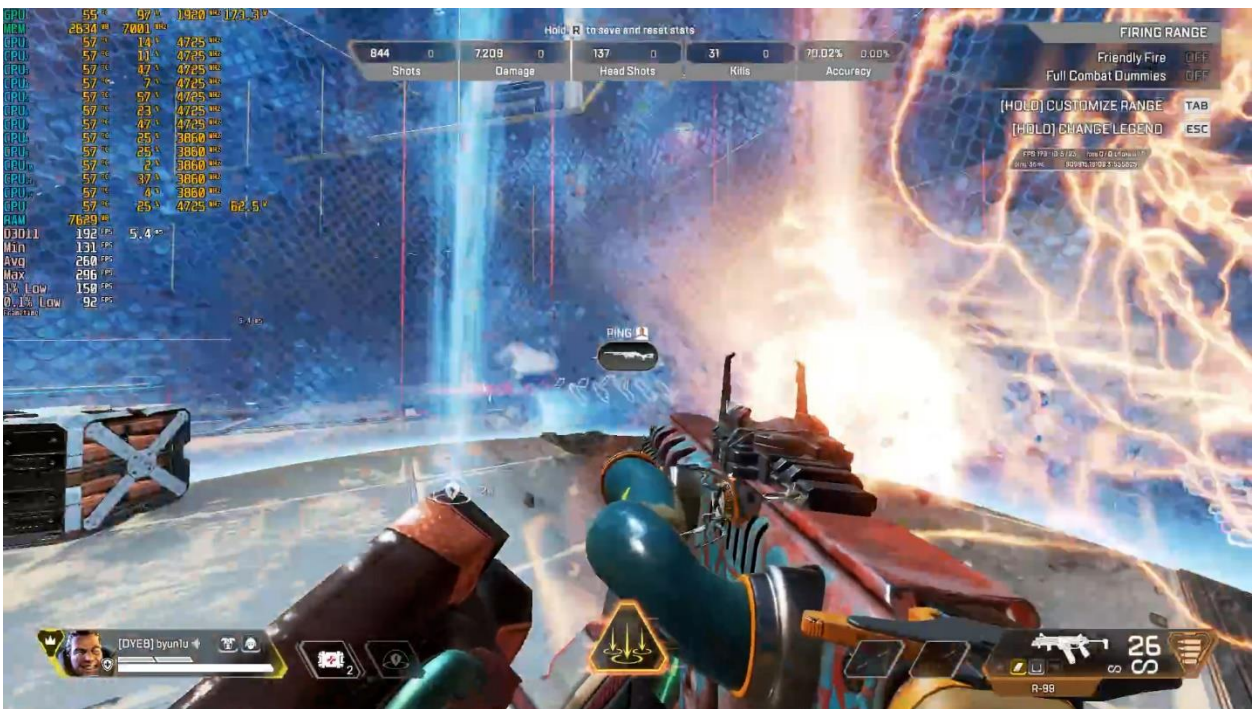


Рисунок 8.10 – Приклад зображення з енкодера NVENC P7



Рисунок 8.13 – Пример изображения с кодировщика NVENC P1



Рисунок 8.14 – Пример изображения с кодировщика NVENC P3



Рисунок 8.15 – Пример изображения с энкодера NVENC P5



Рисунок 8.16 – Пример изображения с энкодера NVENC P7



Рисунок 8.17 – Пример изображения с энкодера NVENC P7 8000kb/s



Рисунок 8.18 – Приклад зображення з енкодера NVENC P7 8000кб/с 936р

Висновки по розділу: у цьому розділі було дослідження енкодера NVENC та порівняння його пресетів. Він виступає як високоефективний кодувальник, проявляючи ефективність навіть на найшвидшому пресеті P1, але зображення може супроводжуватися виявленою фрагментацією. Для користувачів із слабкими відеокартами рекомендується використовувати пресети P1-P3. У разі наявності більш потужної відеокарти можна вибрати між пресетами P4-P5. Флагманські відеокарти дозволяють комфортно використовувати пресети P6-P7. Важливо відзначити, що пресет P7 майже не відрізняється за якістю зображення від пресету P6. Також NVENC мінімально впливає на продуктивність відеокарти, що робить його оптимальним вибором для трансляцій при високій якості зображення та обмежених можливостях.

9 ПОРІВНЯННЯ ДЕКОДЕРІВ ПЛАТФОРМ TWITCH ТА YOUTUBE

Порівнюючи якість зображення між Twitch та YouTube, можна відзначити, що при однаковій роздільній здатності Twitch здатний відтворити більш високоякісне зображення (Рис. 9.1-9.9), навіть при використанні пресету P1, порівняно із YouTube, де використання, навіть, пресету P7 може призвести до менш задовільного результату. Однак, варто враховувати особливі нюанси, пов'язані з декодерами платформ.

YouTube використовує різні декодери для різних роздільних здатностей, зокрема AVC1 для роздільності від 144p до 1080p та VP9 для роздільності від 1440p до 4320p. Декодер для роздільності 1440p і вище на YouTube виявляється значно вищої якості, наближаючись до рівня декодера Twitch AVC1.

Важливо врахувати, що при наявності потужної відеокарти можна застосовувати апскейлінг зображення для трансляції на YouTube, щоб досягти високоякісного зображення. Однак, якщо ваша відеокарта не дуже потужна, і транслюєте в роздільності 720p-1080p, Twitch може забезпечити більш якісним зображення при меншій роздільній здатності та бітрейту. Таким чином, вибір між цими платформами може залежати від ваших потреб та ресурсів.



Рисунок 9.1 – Приклад зображення Twitch NVENC P7 8000кб\с 1080р

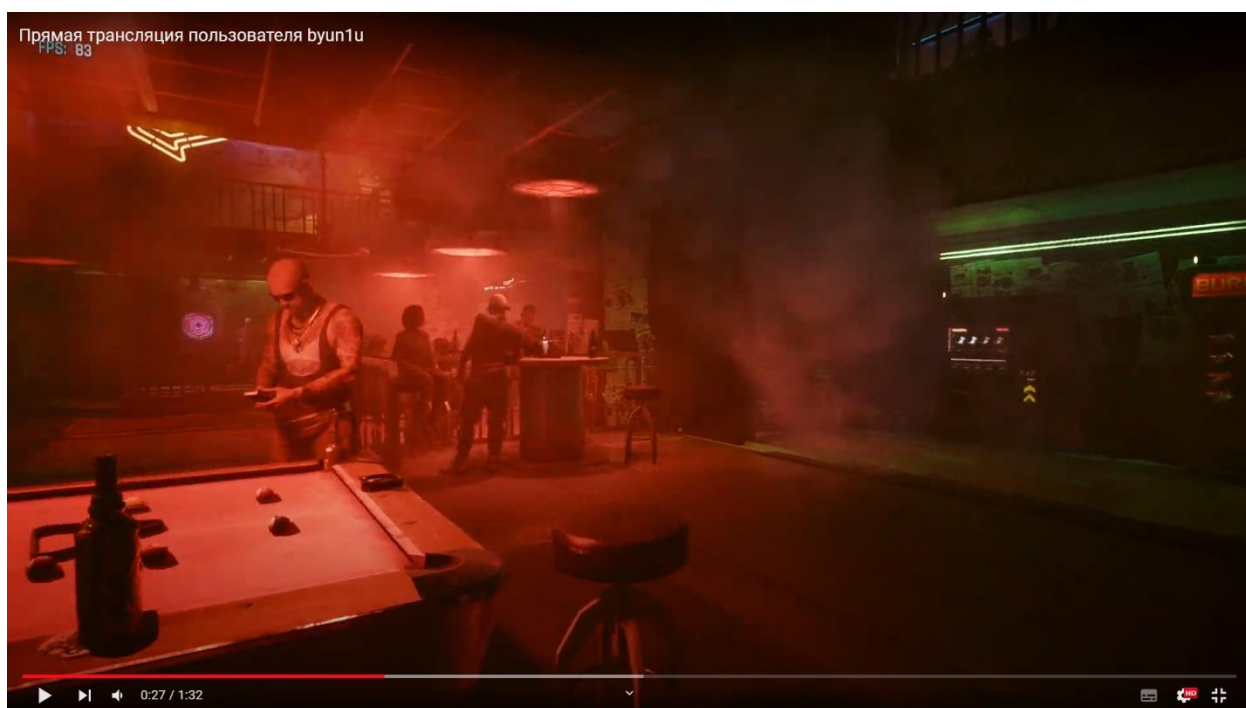


Рисунок 9.2 – Приклад зображення YouTube NVENC P7 25000кб\с 1080р

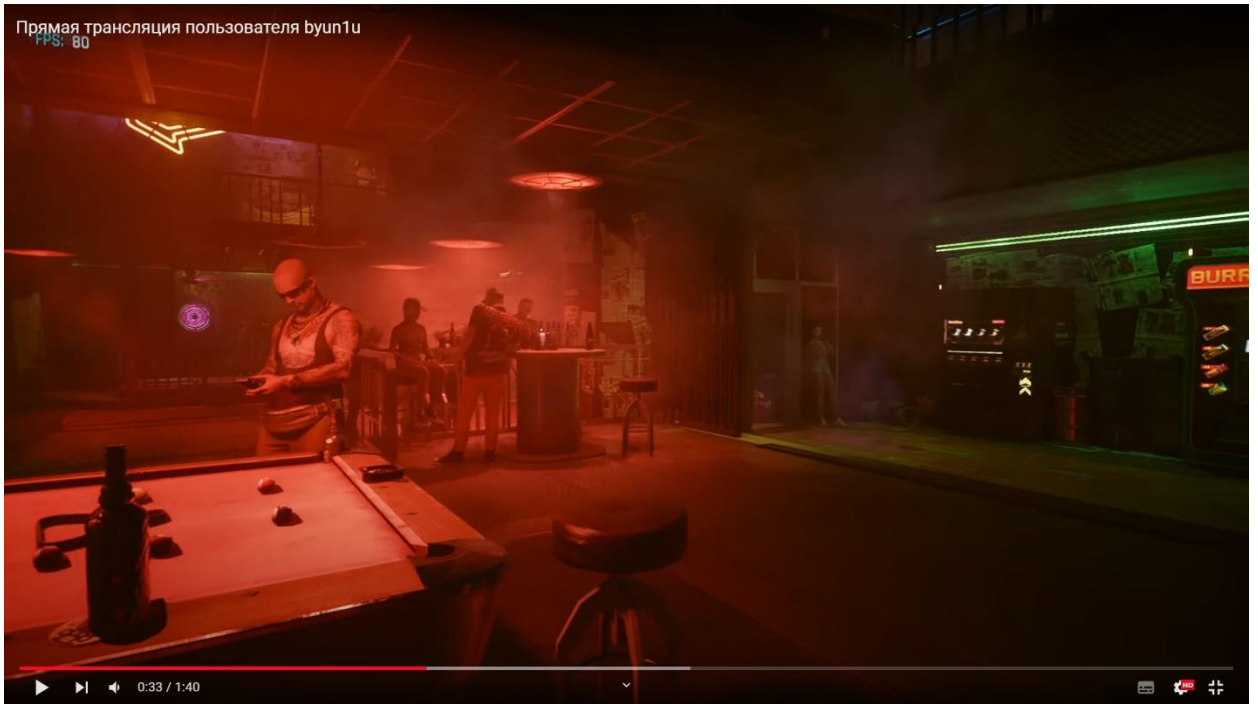


Рисунок 9.3 – Приклад зображення YouTube NVENC P7 25000кб\с upscale
1440р

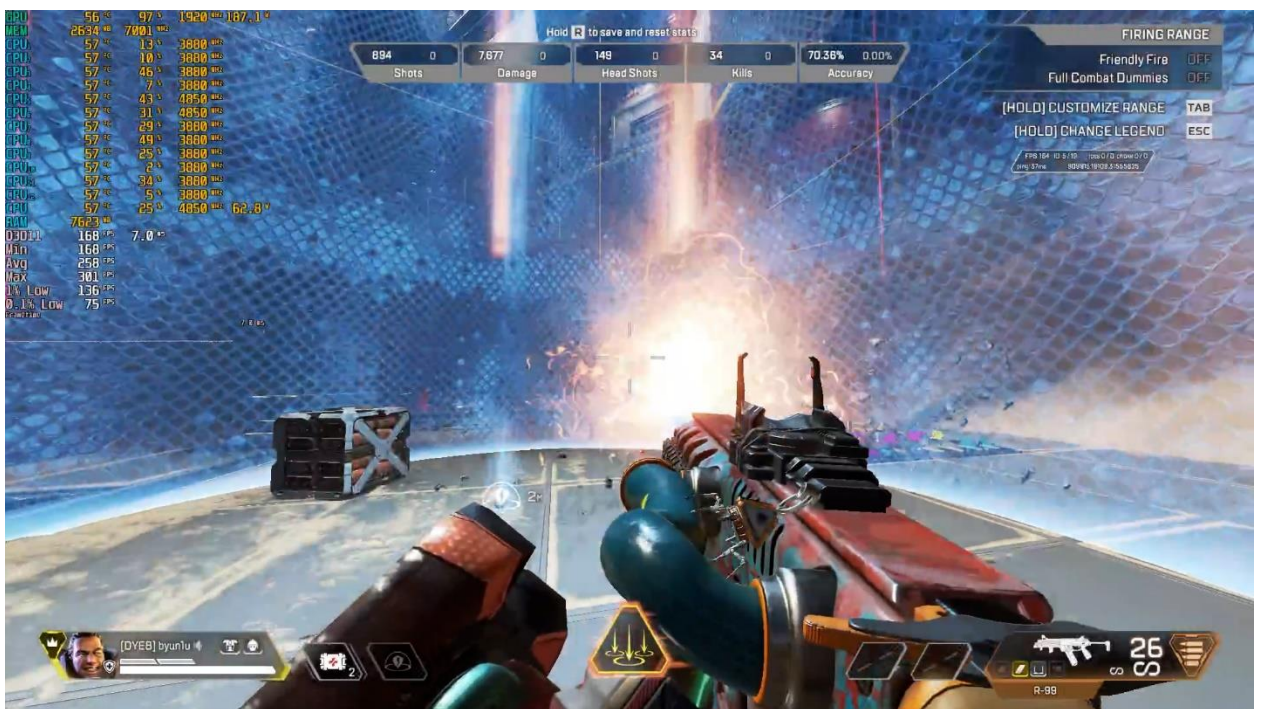


Рисунок 9.4 – Приклад зображення Twitch NVENC P7 8000кб\с 1080р

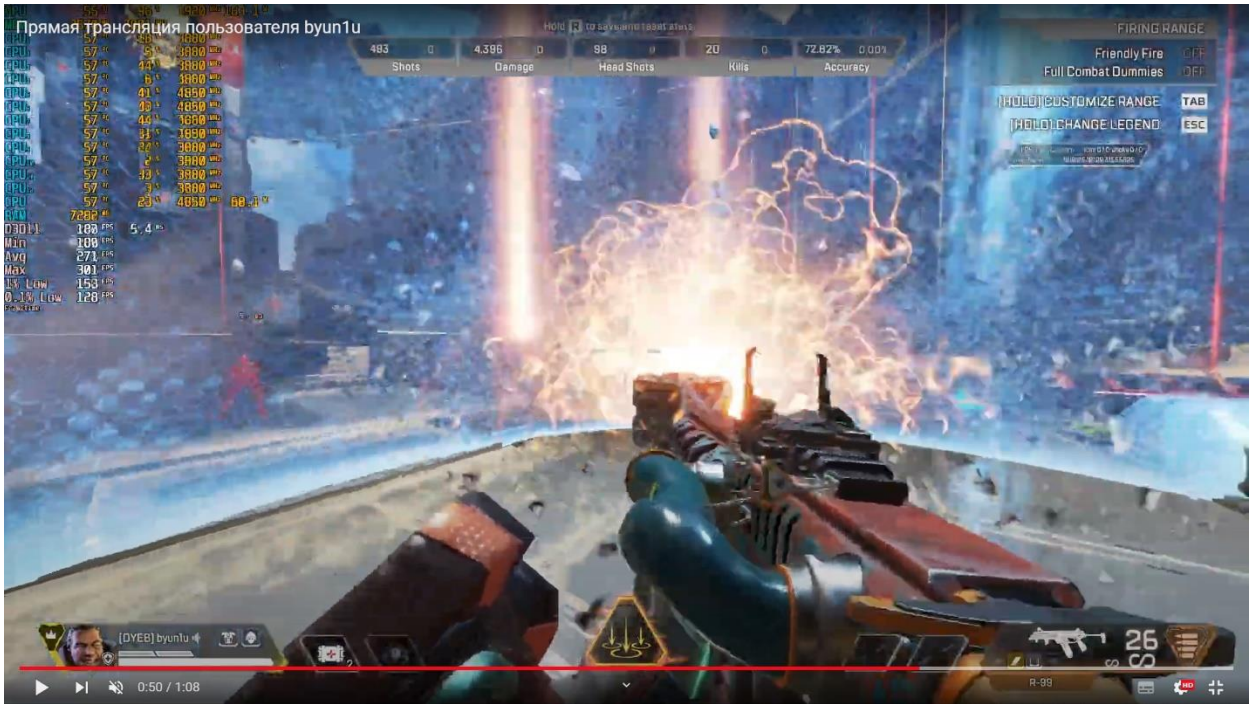


Рисунок 9.5 – Приклад зображення YouTube NVENC P7 25000кб\с 1080р

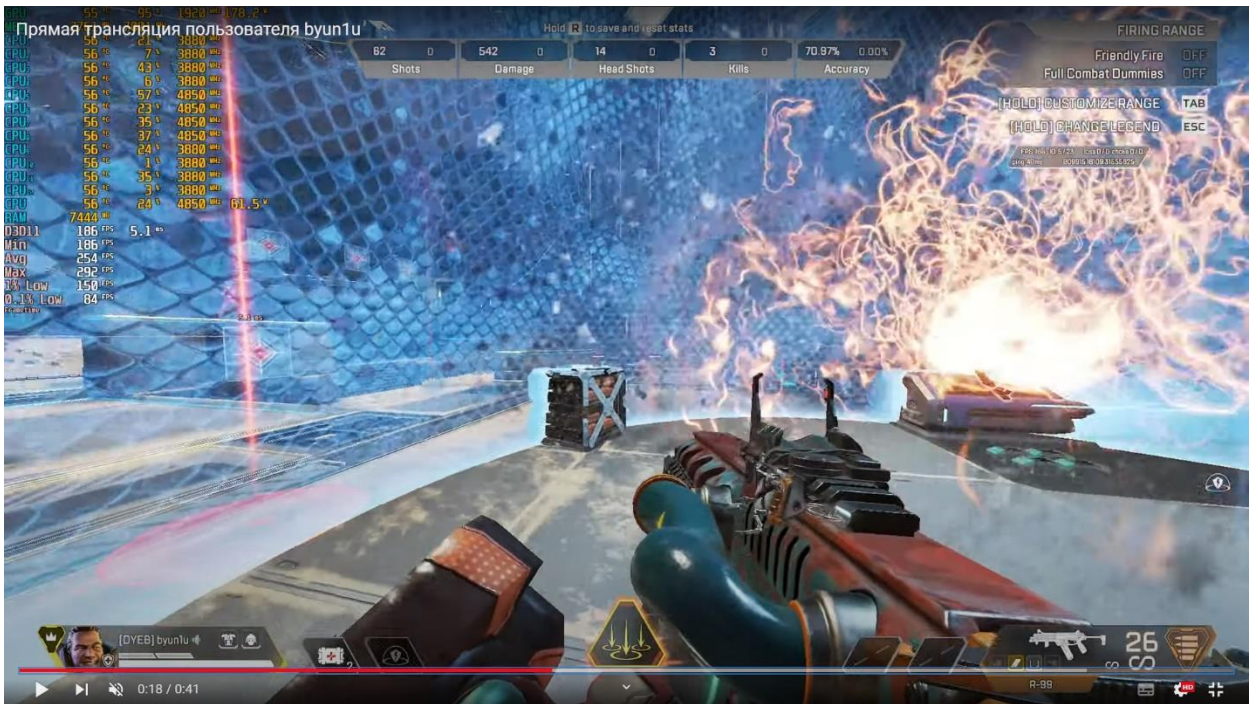


Рисунок 9.6 – Приклад зображення YouTube NVENC P7 25000кб\с upscale
1440р



Рисунок 9.7 – Пример изображения Twitch NVENC P7 8000кб\с 1080p



Рисунок 9.8 – Пример изображения YouTube NVENC P7 25000кб\с 1080p

ВИСНОВКИ

Під час виконання кваліфікаційної роботи був проведений аналіз засобів, технологій та методів, які використовуються під час прямих трансляцій.

Були розглянуті декілька видів надмірності інформації, такі як: статистична, психовізуальна та структурна, їх методи зменшення надмірності. Далі було розглянуто внутрішньокадрове стиснення, його алгоритм, переваги та недоліки. На основі цих даних було проведено дослідження стиснення даних. Напівтонові зображення погано стискаються методами Rar і Zip, тому що містять мало повторюваної інформації. Сусідні пікселі в них досить мало відрізняються. Це зменшує ефективність стиснення без втрат. Зображення тексту містить великі однорідні області, тому ефективність стиснення висока.

Далі були надані рекомендації до комплектуючих комп'ютера для успішного проведення прямої трансляції, а саме вибір процесора: це повинен бути сучасний процесор з 6 ядрами і більше, бажано не старший за Ryzen покоління Zen 2 або Intel 10-го покоління. Відеокарта: Nvidia з кодувальником не старіше 6-го покоління, наприклад, GTX 1060 і новіше, тому що AMD та Intel набагато гірше кодують динамічний контент з дуже вираженою фрагментацією зображення. Оперативна пам'ять: рекомендується мати щонайменше 16 ГБ, оскільки окрім самої трансляції обов'язково будуть запущені фонові процеси які теж потребують пам'яті. Накопичувач: SSD є найкращим вибором для уникнення затримок при завантаженні програм та файлів, та не слід використовувати віддалений сервер для зберігання потрібних даних під час трансляції.

Отже на основі цих рекомендацій комп'ютер на якому проводилося дослідження мав процесор 6 ядерний Ryzen покоління Zen 3, відеокарта Nvidia RTX 3070 7 покоління кодувальника. оперативна пам'ять 32GB, накопичувач SSD.

Було налаштовану програму OBS для проведення прямої трансляції, а саме були налаштовані: роздільна здатність, частота кадрів, додані відео та аудіо

джерела, енкодері. Також була досліджена можливість трансляції на декілька стрімінгових платформ одразу за допомогою сторонніх сервісів та плагінів.

Було розібрано що таке кодувальники (енкодері), їх види та налаштування, які вони мають вимоги до комп'ютера та інтернет мережі. Також були приведені сценарії використання для кожного з енкодерів, їх переваги та недоліки, а саме NVENC на сьогодні є найбільш ефективним кодувальником для трансляції будь-якого контенту, лекцій, ігор, концертів з мінімальним навантаженням на комп'ютер і водночас зі збереженням гарної якості відео, яка сприяє гарному досвіду глядачів. Переважно обирати найповільніший пресет P7 з найліпшою якістю кодування, але також можна обирати пресети починаючи з P3, вони також мають гарну якість кодування, але можливе деяка фрагментація зображення.

В свою чергу енкодер x264 є менш універсальним кодувальником, він вимагає дуже значну обчислювальну спроможність від центрального процесора, що стане проблемою під час прямої трансляції динамічного контенту або важких робочих програм для роботи. Для цього кодувальника переважно обирати пресети починаючи від Medium, тому що на більш швидких пресетах дуже чітко виражена фрагментація зображення. Для використання цього енкодера з пресетом Medium щонайменше знадобиться сучасний процесор з 8 ядрами. Його слід використовувати тільки якщо немає зовнішньої відеокарти, або є другий комп'ютер який буде виключно займатися кодуванням прямої трансляції.

Було проведено дослідження якості відео на платформах Twitch та YouTube. Twitch надає більш гарну якість зображення при менших вимогах до інтернету та комп'ютеру, а саме для FULL HD потрібен бітрейт у 6000кб\с, коли у YouTube 12000кб\с і при цьому якість на YouTube буде значно гіршою ніж на Twitch. Для досягнення такої ж самої якості як на Twitch потрібно підвищувати роздільну здатність до 2K та відповідно підвищувати бітрейт до 24000кб\с, що створить додаткове навантаження на комп'ютер та інтернет. Тому якщо ви не володієте безмежними ресурсами у вигляді потужності процесору, відеокарти, пропускної спроможності інтернету слід обрати Twitch як платформу для проведення прямих

трансляцій з найкращою якістю зображення при менших витратах ресурсів ПК та інтернету, що буде сприяти досвіду глядачів.

ПЕРЕЛІК ДЖЕРЕЛ, ПОСИЛАНЬ

1. Help Twitch [Електронний ресурс] – Режим доступу до ресурсу:
<https://help.twitch.tv/s/article/broadcast-guidelines?language=ru> (дата звернення 18.10.2023)
2. Support Google [Електронний ресурс] – Режим доступу до ресурсу:
<https://support.google.com/youtube/answer/2853702?hl=ru> (дата звернення 18.10.2023)
3. YouTube [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.youtube.com> (дата звернення 04.12.2023)
4. NetflixTechBlog [Електронний ресурс] – Режим доступу до ресурсу:
<https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652> (дата звернення 23.10.2023)
5. MMIP [Електронний ресурс] – Режим доступу до ресурсу:
<https://imaging.cs.msu.com/eng/research/ringing> (дата звернення 20.10.2023)
6. NVIDIA [Електронний ресурс] – Режим доступу до ресурсу:
<https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new>
(дата звернення 21.10.2023)
7. Cesar Sepulveda-Valdez ; Oleg Sergiyenko ; Vera Tyrsa ; Wendy Flores-Fuentes Julio César Rodríguez-Quñonez ; Fabian Natanael Murrienta-Rico ; Jesús Elías Miranda-Vega ; Paolo Mercorelli ; Marina Kolendovska. "Geometric analysis of a laser scanner functioning based on dynamic triangulation," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1398-1403, doi: 10.1109/ISIE45063.2020.9152268.
8. Cuauhtémoc Mariscal-García; Wendy Flores-Fuentes; Daniel Hernández Balbuena; Julio C. Rodríguez-Quñonez ; Oleg Sergiyenko. "Classification of Vehicle Images through Deep Neural Networks for Camera View Position Selection," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1376-1380, doi: 10.1109/ISIE45063.2020.9152440.

9. Developing and Applying Optoelectronics in Machine Vision/ O. Sergiyenko, J.C. Rodriguez-Quiñonez, IGI Global, 2016; 341p.

10. Experimental estimation of direction finding to unmanned air vehicles algorithms efficiency by their acoustic emission, /Oleynikov, V., Zubkov, O., Kartashov, V., ...Sheiko, S., Babkin, S.//2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 - Proceedings, 2019, стр. 175-178, 9061337

11. Features of acoustic noise of small unmanned aerial vehicles / Semenets, V.V., Kartashov, V.M., Leonidov, V.I. //Telecommunications and Radio Engineering (English translation of *Elektrosvyaz and Radiotekhnika*), 2020, 79(11), стр. 985-995

12. Geometric Analysis Of A Laser Scanner Functioning Based On Dynamic Triangulation /Sepulveda-Valdez, C., Sergiyenko, O., Tyrsa, V, Mercorelli, P., Kolendovska, M.// IEEE International Symposium on Industrial Electronics, 29th IEEE International Symposium on Industrial Electronics, ISIE 2020; Delft; Netherlands; 17 June 2020 до 19 June 2020; Volume 2020-June, June 2020, № 9152268, Pages 1398-1403

13. I. Y. A. Corpus, L.Lindner, O.Sergiyenko. "Transimpedance Amplifier for Laser Scanning System Range Extension," 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1421-1426, doi: 10.1109/ISIE45063.2020.9152487.

14. Ivanov, M., Sergiyenko, O., Mercorelli, P., Hernandez, W.c, Rodriguez Quinonez, J.C.d, Katashov V., Kolendovska, M., Iryna, T. Effective informational entropy reduction in multi-robot systems based on real-time TVS. IEEE International Symposium on Industrial Electronics, 2019-June,8781209, c. 1162-1167.

15. Jonathan J. Sanchez-Castro ; Julio C. Rodríguez-Quiñonez ; Luis R. Ramírez-Hernández ; Guillermo Galaviz ; Daniel Hernández-Balbuena ; Gabriel Trujillo-Hernández ; Wendy Flores-Fuentes ; Paolo Mercorelli ; Wilmar Hernández-Perdomo ; Oleg Sergiyenko ; Félix Fernando González-Navarro. "A Lean Convolutional Neural Network for Vehicle Classification," 2020 IEEE 29th International Symposium on

Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 1365-1369, doi: 10.1109/ISIE45063.2020.9152274.

16. Lindner, L., Sergiyenko, O., Rivas-López, M., (...), Gurko, A., Kartashov, V.M. Machine vision system for UAV navigation; IEEE, 2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference, ESARS-ITEC, 2016; pp.1–6. DOI: 10.1109/ESARS-ITEC.2016.7841356.

17. M. Ivanov, O. Sergiyenko, V. Tyrsa, P. Mercorelli, V. Kartashov, W. Hernandez, S. Sheiko, M. Kolendovska. Individual scans fusion in virtual knowledge base for navigation of mobile robotic group with 3D TVS // Proceedings of 44th Annual Conference of IEEE Industrial Electronics Society (IECON).. -2018. – Washington DC, USA. -S. 3187-3192. . ISBN 978-1-5090-6683-4/18/.

18. Murrieta-Rico, F.N., Petranovskii, V., Galvan, D.H., Sergiyenko, O., Yocupicio-Gaxiola, R.I., De Dios Sanchez-Lopez, J. Phase effect in frequency measurements of a quartz crystal using the pulse coincidence principle. 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), Delft, Netherlands, 17-19 of June 2020, pp. 185-190, 9152255, DOI: 10.1109/ISIE45063.2020.9152255

19. Oleksandr Sotnikov, Vladimir Kartashov, Oleksandr Tymochko, Oleg Sergiyenko, Vera Tyrsa, Paolo Mercorelli, Wendy Flores-Fuentes. Methods for Ensuring the Accuracy of Radiometric and Optoelectronic Navigation Systems of Flying Robots in a Developed Infrastructure. Chapter 16// Machine Vision and Navigation; Springer, Cham. pp.537–578. Editors: Sergiyenko, Oleg, Flores-Fuentes, Wendy, Mercorelli, Paolo. DOI: 10.1007/978-3-030-22587-2_16.