

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет радіоелектроніки  
Факультет Комп'ютерних наук  
Кафедра Програмної інженерії

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

\_\_\_\_\_ другий (магістерський) \_\_\_\_\_

(рівень вищої освіти)

Дослідження методів стиснення відеоматеріалів для клієнт-серверної взаємодії у  
реальному часі

Виконав:

студент 2 курсу групи ІПЗм-20-3

Брестовицький Р.М.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного  
забезпечення

Тип програми Освітньо-наукова

Керівник доц. Афанасьєва І.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри

3.В. Дудар

\_\_\_\_\_ 2022 р. \_\_\_\_\_

# Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
Кафедра \_\_\_\_\_ Програмної інженерії \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 121– Інженерія програмного забезпечення \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-наукова програма \_\_\_\_\_  
Освітня програма \_\_\_\_\_ Інженерія програмного забезпечення \_\_\_\_\_

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_» \_\_\_\_\_ 202\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента \_\_\_\_\_ Брестовицького Романа Миколайовича \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів стиснення відеоматеріалів для клієнт-серверної взаємодії у реальному часі»  
затверджена наказом університету від «24» березня 2022р. № 412 Ст
2. Термін подання студентом роботи до екзаменаційної комісії «21» травня 2022р.
3. Вихідні дані до роботи методи стиснення відеоматеріалів, методи клієнт-серверної взаємодії у реальному часі, відеокодеків, які використовують різні методи стиснення, результати оцінки якості відеокодеків, пояснювальна записка.
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної галузі та постановка задачі, дослідження методів стиснення відеоматеріалів, дослідження технологій клієнт-серверної взаємодії, вивчення методів стиснення у сучасних відеокодеках.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі та постановка задачі	17.01.2022	виконано
2	Аналіз існуючих методів та алгоритмів	31.01.2022	виконано
3	Виконання експериментальної частини	21.02.2022	виконано
4	Підготовка пояснювальної записки	03.04.2022	виконано
5	Підготовка презентації та доповіді	03.04.2022	виконано
6	Перевірка на академічний плагіат	11.05.2022	виконано
7	Нормоконтроль	14.05.2022	виконано
8	Рецензування	15.05.2022	виконано
9	Занесення диплома в електронний архів	16.05.2022	виконано
10	Попередній захист	17.05.2022	виконано
11	Допуск до захисту у зав. кафедри	18.05.2022	виконано
12	Захист кваліфікаційної роботи	21.05.2022	

Дата видачі завдання \_\_\_\_\_ 17 січня \_\_\_\_\_ 2022 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Афанасьєва І.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 73 с., 15 рис., 4 табл., 30 джерел, 4 додатків.

БІТОВА ШВИДКІСТЬ, ВЗАЄМОДІЯ У РЕАЛЬНОМУ ЧАСІ, ВІДЕОМАТЕРІАЛИ, КЛІЄНТ, КОДЕКИ, СТИСНЕННЯ, СЕРВЕР.

Актуальність роботи полягає у тому, що передача відеоматеріалів є важкою задачею, з точки зору використання вільного простору для збереження, якості відтворення відеоматеріалів та швидкості передачі.

Об'єктом дослідження є методи стиснення відеоматеріалів у контексті взаємодії клієнта та сервера у реальному часі.

Предметом дослідження є вплив стиснення відеоматеріалу на якісні показники та загальний обсяг даних.

Метою роботи є знаходження залежностей якості та розміру від ступеня стиснення відеоматеріалів у контексті взаємодії клієнту та серверу у реальному часі.

У роботі було проведено експериментальне порівняння сучасних відеокодеків, та знайдена залежність якості та розміру відеоматеріалів від їх бітової швидкості. Розглянуті методи оцінки якості та технології для клієнт-серверної взаємодії у реальному часі. Для експериментальної частини було використано програмні засоби: Movavi Video Converter 19 Premium, VideoMaster 11.0, Microsoft Office Excel. Також було розроблено власний код, який реалізує суб'єктивний алгоритм оцінки якості SSIM. Для написання коду було використано мову програмування Python.

BIT RATE, REAL-TIME INTERACTION, VIDEO, CLIENT, CODECS, COMPRESSION, SERVER.

The relevance of the work is that the transfer of video is a difficult task, in terms of the use of free space for storage, the quality of video playback and transmission speed.

The object of research is the methods of video compression for real time client-server interaction.

The subject of the study is the impact of video compression on quality indicators and total data.

The aim of the work is to find the dependences of quality and size on the degree of compression of video materials in the context of interaction between client and server in real time.

An experimental comparison of modern video codecs was performed, and the dependence of the quality and size of video materials on their bit rate was found. Methods of quality assessment and technologies for real-time client-server interaction are considered. For the experimental part the following software tools were used: Movavi Video Converter 19 Premium, VideoMaster 11.0, Microsoft Office Excel. Own code was developed, which implements a subjective algorithm for assessing the quality of MS-SSIN. The Python programming language was used to implement the code.

Я, Брестовицький Роман Миколайович, студент групи ПЗм-20-3, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів стиснення відеоматеріалів для клієнт-серверної взаємодії у реальному часі», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Зміст.....	6
Вступ.....	7
1 Аналіз предметної галузі.....	9
1.1 Аналіз існуючих рішень.....	9
1.2 Постановка задачі.....	13
2 Аналіз існуючих методів і алгоритмів.....	15
2.1 Методи і алгоритми стиснення.....	15
2.2 Відеокодеки.....	21
2.3 Методи клієнт-серверної взаємодії у реальному часі.....	25
3 Методи оцінювання якості відеоматеріалу.....	31
4 Експериментальні дослідження.....	37
4.1 Програмне забезпечення.....	37
4.2 Реалізація коду для оцінки якості відеоматеріалів.....	40
4.3 Опис та підготовка експерименту.....	41
4.4 Результати експерименту.....	48
Висновки.....	51
Перелік джерел посилання.....	53
Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії.....	56
Додаток А Звіт результатів перевірки кваліфікаційної роботи на унікальність тексту.....	57
Додаток Б Слайди презентації.....	58
Додаток В Апробація результатів роботи.....	70
Додаток Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015.....	73

## ВСТУП

Уже існує велика кількість застосунків для онлайн конференцій, ці системи надають можливість запуску відео у режимі реального часу, але дані потребляють велику кількість простору, а у свою чергу напружують трафік інтернету. Наприклад, година Full HD запису може займати 64 ГБ пам'яті, у свою чергу цей запис має буди надісланий до співбесідника. При великій швидкості поліпшення сучасних технологій, збільшується потужність відеокамер та їх можливість у зйомці відео з високоякісною картинкою. Що безумовно дає переваги у сферах використання відеозаписів (охоронна зйомка, аматорська зйомка), але потребує великих баз для збереження матеріалів.

Також взаємодія у реальному часі має великі переваги для сучасників, адже з'являється можливість спілкуватися на великій дистанції один від одного. Більшість людей перебувають в постійному русі та частіше за все використовують мобільний інтернет, який зазвичай має обмеження в кількості використаного інтернету за період часу. Гонка за оптимізацією продовжується серед передових розробників. Всі борються за можливість поліпшити якість відеоматеріалу при однаковому обсязі, або навпаки суттєво зменшити розмір файлу без зміни якості.

Високоякісна передача відеоматеріалів має стратегічно важливу функцію, адже вона необхідна як у військовому або охоронному мистецтві, так і в розважальному або робочому.

Була розроблена велика кількість підходів та методів для стиснення відеоматеріалів, але якщо сильно стиснути потік, то можуть з'явитися помилки та артефакти відео, які шкодять нормальному сприйняттю картинки [1].

Під час експлуатації може виникнути декілька проблем: використовується статичне відео (матеріал з веб-камери), у цьому випадку можна використовувати сильне стиснення, адже відео має статичний фон; використовується динамічне відео

(доповідь у великому залі), у цьому випадку необхідно понизити ступінь стиснення, адже у іншому випадку відео матиме велику кількість артефактів.

Для підвищення ефективності та якості стиснення необхідно розраховувати показники збереження відео та контролювати параметри стиснення. Під час повільних сцен на відеозаписі, немає необхідності зберігати усі дрібні деталі фону, легше за все зберегти одну картину та додатково змінювати її коли необхідно.

Для оцінки якості існують об'єктивні та суб'єктивні методи оцінки. Сучасні методи використовують математичні моделі, якими можна розрахувати коефіцієнт якості. Цей коефіцієнт наближений до того, як оцінює відеоматеріал людське око. Більш застарілі методи використовують математичні моделі, які націлені на виявленні надлишкових елементів та/або артефактів.

Таким чином, виникає потреба у налагодженні параметрів стиснення відео для різних сцен під час взаємодії у реальному часі.

Отримані результати також можуть бути використані для подальших досліджень у сфері кодування відеоматеріалів, також досліджена інформація може допомогти у створенні нових систем та застосунків для відеоконференцій або трансляцій потоку відео у реальному часі.

Під час дослідження методів стиснення відеоматеріалів та клієнт-серверної взаємодії у реальному часі було використано теоретичний підхід. Для цього були проаналізовані існуючі методи суб'єктивної оцінки якості відео, методи стиснення відеоматеріалу та технології для взаємодії клієнта та сервера у реальному часі.

Результати досліджень були опубліковані в збірнику матеріалів XII Міжнародній науково-практичній конференції «СУЧАСНІ НАПРЯМКИ РОЗВИТКУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЗАСОБІВ УПРАВЛІННЯ», що відбулася 27-28 квітня 2022 року. Сертифікат учасника представлено в додатку В.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз існуючих рішень

Основними представниками, які постачають відеоматеріали у реальному часі, є камери для охорони, телебачення, прямі трансляції, відеоконференції. Усі сфери потребують високоякісного потоку відео у реальному часі.

Великою сферою для розважання, роботи, навчання є платформи для показу потоку відео у реальному часі. Більшість спортивних та ігрових турнірів проводяться з їх допомогою.

Стрімінг – це потокове мовлення у реальному часі. До стрімінгових мультимедіа відносять музику, відео та інформацію, яку користувач отримує безперервно від потокового провайдер. Більшість ринку стрімінгу належить до сегменту кіберспорту, тут геймери показують у режимі онлайн, як вони проходять ігри.

Twitch – відеострімінговий сервіс, що спеціалізується на тематиці комп'ютерних ігор, у тому числі на трансляціях геймплею та кіберспортивних турнірів [2]. Відео на платформі Twitch можна переглядати як у реальному часі, так і на запит користувача. Twitch часто використовується для навчання відеоігор. Природа Twitch дозволяє великій кількості учнів взаємодіяти один з одним і з викладачем у режимі реального часу. Twitch також використовується для навчання розробці програмного забезпечення, коли спільноти користувачів транслюють проекти програмування та розповідають про свою роботу.

Restream – це хмарна потокова служба, яка дозволяє одночасно транслювати відео на більш ніж 30 платформах, включаючи Twitch, YouTube, Facebook тощо [3]. Це також одна з небагатьох поточкових платформ, які дозволяють перемикання платформ і каналів у реальному часі. Це означає, що ви можете редагувати та додавати більше каналів, поки ви на зв'язку, без необхідності виходити з трансляції або

перезапускати кодер. Також служба має вбудований підсилювач для потокової передачі, який доступний у будь-який час. Restream дозволяє глядачам з різних каналів взаємодіяти між собою. Служба дозволяє зміну метаданих в одному місці для усіх запущених каналів.

Wirecast – це платформа для виробництва потокового відео в прямому ефірі, розроблена компанією Telestream [4]. Платформа допомагає створювати прямі трансляції, схожі на телевізійні, і трансляції для публікації в Інтернеті. Wirecast забезпечує прийом багатоканального звуку та підтримку пристроїв, сумісних з WASAPI, ASIO та CoreAudio. Також забезпечує використання таких функцій, як повторення, годинник, табло та таймери для спортивних подій у прямому ефірі. Платформа має більше 100 вбудованих заголовків та графіків та підтримку QR-кодів, Chroma Key та списків відтворення, що робить презентації прямої трансляції більш професійною.

Відеоконференція – це технологія, яка дозволяє людям бачити і чути один одного, обмінюватися даними та спільно обробляти їх в інтерактивному режимі, використовуючи можливості звичного для всіх комп'ютера, максимально наближаючи спілкування на відстані до реального живого спілкування. Области застосування відеоконференції величезні. На сьогоднішній день практично не залишилося сфери життєдіяльності, в якій не використовують зв'язок по відео.

Сервіси для відеоконференцій створюють всі умови для віддалених зустрічей. Якщо раніше збори могли бути лише очними, зустріч відбувалося в кабінетах відповідних установ, то сьогодні можливі варіанти. Конференційний зв'язок дозволяє підтримувати контакт з співробітниками, що знаходяться в різних куточках земної кулі, у тому числі з тими, хто використовує для цього своїх смартфонів. Це виводить ділові та неформальні комунікації на новий рівень.

Сучасні технології зв'язку з використанням відео є невід'ємною складовою роботи багатьох сфер. З початком епідемії більшість навчальних закладів перейшли

до дистанційної форми навчання, що потребує взаємодію з викладачем у реальному часі. Застосунки для відеоконференцій краще за все підходять для цього.

Google Meet – сервіс для обміну повідомленнями у реальному часі, а також для проведення відео дзвінків [5]. Ця система була анонсована компанією Google у 2017 році. У браузерях, які засновані на технології Chromium (Google Chrome та інші), використовуються приховані можливості WebRTC, які не описані в документації та періодично з'являються першими у рішеннях для Meet та Hangouts. Прикладами є захоплення екрану, розмиття фону, підтримка апаратного кодування на деяких платформах. Сервіс має інтеграцію з інструментами G Suite, до яких входять пошта, календар. Google Meet має функцію автоматичного запису та збереження матеріалу на власний диск, які також додаються у події календаря.

Jitsi Meet – система з відкритим кодом, яку випустила компанія 8x8 [6]. Технологія Jitsi заснована на архітектурі Simulcast, що означає нестабільну роботу на слабких каналах зв'язку та високі вимоги швидкості підключення на стороні сервера. Програма дозволяє використання лише у браузері та не має клієнтських застосунків для спільної роботи. Для налагодження роботи у корпоративному середовищі, з використанням Jitsi, необхідна самостійна розробка та встановлення додаткового програмного забезпечення. Для використання системи рекомендовано утриматися від використання бездротових навушників, а також утриматися від запуску інших процесів на пристрої, який використовується для зв'язку.

Discord – застосунок, який створювався як безкоштовний месенджер з підтримкою VoIP та відеоконференцій для геймерів [7]. Зараз цей сервіс став набагато популярнішим, навіть серед ділової аудиторії. Величезною перевагою Discord є його безкоштовність. Система при низькому навантаженні на ресурси пристрою забезпечує високу якість звуку, завдяки використанню аудіокодека Opus. Як запевняють розробники, програма надійно захищена від DDoS-атак, що додає впевненості, що відеоконференція пройде без проблем. При включенні режиму демонстрації, ховається вся особиста інформація, відключаються звуки та сповіщення на робочому

столі. Вмикається режим автоматично (при запуску трансляцій), також режим можна ввімкнути або вимкнути вручну

Zoom – застосунок для конференцій [8]. Ця система не використовує технологію WebRTC для передачі та декодування медіа-даних. Зроблено це задля економії серверних ресурсів. На стороні браузеру задіяні інші веб-технології – низькорівневий WebAssembly і WebSocket. При використанні таких нестандартних підходів передачі потоку можуть виникати проблеми з якістю картинки у деяких учасників. Zoom працює майже на всіх пристроях. Застосунок може відправляти запрошення, використовуючи телефон, пошту або контакти компанії. Також система має інтеграцію з Google диском та сервісом Dropbox. Під час дзвінка користувач має змогу підключити демонстрацію власного екрану або підключити білу дошку для малювання. Під час конференції усіх користувачів можна поділити на різні кімнати, що дає змогу окремо обговорити деталі конференції.

Microsoft Teams – корпоративна платформа, що об'єднує у робочому просторі чат, зустрічі, нотатки та вкладення [9]. Розроблена компанією Microsoft як конкурент найпопулярнішого корпоративного рішення Slack. Microsoft Teams є частиною пакету Office 365 і поширюється використовуючи корпоративну передплату. Крім Office 365 також інтегрований зі Skype, крім того, є можливість інтеграції з програмами сторонніх розробників. Дозволяє створювати групи/кімнати під проекти або департаменти, спілкуватися, обмінюватися файлами з колегами. Є боти для сповіщень та відповідей на запити. Microsoft Teams – повністю хмарний сервіс, що має низку переваг. Можливість доступу до інформації з будь-якої точки та будь-якого девайсу – значна бізнес-перевага. Переміщення усієї документації у цифровий формат сприяє гнучкості роботи співробітників.

BigBlueButton – це вільне програмне забезпечення для відеоконференції [10]. Особливий акцент розробники роблять на дистанційну освіту (присутні такі функції, як інтерактивна дошка, показ контенту, підтримка опитувань тощо). Підтримує веб-конференцію до 100 учасників. Користувач може увійти до конференції або як глядач,

або як модератор. Як глядач, користувач може приєднатися до голосової конференції, використовувати камеру, підняти руку (попросити слово) та спілкуватися з іншими людьми. Як модератор користувач має можливість відключити та включити мікрофон будь-якого глядача, видалити будь-якого глядача з конференції, а також передати слово будь-якому глядачеві для виступу (зробити будь-якого користувача ведучим). Ведучий може завантажувати презентації, документи, використовувати дошку.

TrueConf – програмна система уніфікованих комунікацій з підтримкою дозволу 4K Ultra HD, яка працює по інтернету або в закритій локальній мережі [11]. Архітектура програмного забезпечення TrueConf базується на принципах масштабованого кодування відео. Протягом багатосторонньої відеоконференції кожен її учасник отримує індивідуальний набір відеопотоків, що відповідає ширині пропускового каналу, типу мережного з'єднання та властивостям робочої станції.

## 1.2 Постановка задачі

Основним завданням є пошук універсальної комбінації методів, алгоритмів та підходів для вирішення задачі стиснення відеоматеріалів під час клієнт-серверної взаємодії у реальному часі.

Для того, щоб досягти поставленої мети, необхідно дослідити різні алгоритми та моделі, які використовуються для стиснення відеоматеріалів, та знайти оптимальну комбінацію. У якості оптимальної комбінації будемо враховувати мінімально можливі втрати якості картинки, при максимальній швидкості передачі кадрів та мінімально можливому розмірі кадрів.

Використовуючи методи оцінювання якості відеоматеріалу, необхідно знайти можливі оптимальні налагодження для передачі різних потоків відео (статичних, динамічних) для мінімізації затраченої пам'яті при збереженні якості картинки.

Для вирішення завдання необхідно:

- дослідити наявні методи стиснення;
- зробити порівняльний аналіз методів стиснення;
- дослідити методи передачі даних у реальному часі;
- зробити порівняльний аналіз методів передачі даних у реальному часі;
- знайти залежність якості та розміру відеоматеріалів від типу відеокодека та його бітової швидкості;
- експериментальним шляхом знайти найкращу комбінацію методів та налагоджень для них.

У результаті потрібно знайти спосіб, завдяки якому можна оптимізувати налагодження потоку відео.

## 2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І АЛГОРИТМІВ

### 2.1 Методи і алгоритми стиснення

Стиснення відеозображення – це зменшення і видалення надлишкових відеоданих з метою оптимізації зберігання і передачі файлів цифрового відео.

Стиснутий файл створюється за допомогою обробки вихідного відеосигналу, після чого цей файл можна надсилати та обробляти. Для того, щоб відтворити стиснутий файл використовують інверсний алгоритм, який відтворює те ж саме відеозображення, що і оригінал.

Затримка – час, який потрібен на стиснення, надсилання, розпакування та відображення файлу.

Відеокодек – це спільна робота декількох алгоритмів. Це комп'ютерний код, який виконує свою функцію щоразу, коли файл викликається програмним забезпеченням. Відеокодеки також можна використовувати у фізичному обладнанні, наприклад у камері, для перетворення вхідного відео та звуку у цифровий формат. Відеокодеки, які використовують різні стандарти є несумісними один з одним, тому відеодані, які були стиснуті з використанням одного стандарту, неможливо відтворити із застосуванням іншого стандарту. Причиною цього є той факт, що один алгоритм не має можливості вірно декодувати результат кодування, який був отриманий за допомогою роботи іншого алгоритму.

Різні стандарти стиснення використовують різні методи зменшення розмірів даних, таким чином, результати кодування можуть відрізнятися по швидкості передачі даних, якості та рівню затримки [12].

У різних розробників, використовуючи один і той самий стандарт, можуть відрізнятися результати стиснення, адже саме розробник обирає необхідний набір методів та засобів обраного стандарту [13]. Це необхідно, адже різні методи мають використовуватися для різних цілей та з урахуванням різного бюджету розробки.

Професійні програмні розробники для роботи з носіями, які працюють не в режимі реального часу, повинні мати можливість забезпечувати краще кодоване відображення, коли розробники, які працюють з потоком у реальному часі, мають враховувати також швидкість передачі. Таким чином, обраний стандарт не може гарантувати визначену швидкість передачі даних або її якість.

Декодер на відміну від кодера, повинен реалізовувати в собі всі необхідні елементи стандарту з тим, щоб декодувати відповідний потік бітів. Тому стандарт чітко вказує, як саме алгоритм розпакування повинен відновлювати кожен біт стиснутого відеозображення.

Методи стиснення ділять на дві категорії: стиснення без втрат та стиснення з втратами [14]. Під час використання методу без втрат – зображення, отримане після декодування, повністю збігається з початковим. Під час використання методу з втратами – частина інформації втрачається у процесі стиснення. Принцип стиснення з втратами ґрунтується на обмежених можливостях людського зору.

Стиснення без втрат – клас алгоритмів стиснення даних, при використанні яких закодовані дані однозначно можуть бути відновлені. При цьому оригінальні дані повністю відновлюються зі стисненого стану. Цей тип стиснення принципово відрізняється від стиснення даних із втратами. Для кожного з типів цифрової інформації зазвичай існують свої оптимальні алгоритми стиснення без втрат.

У загальних рисах ідея стиснення без втрат полягає у тому, що у вихідних даних знаходять якусь закономірність і з урахуванням цієї закономірності генерують другу послідовність, яка повністю визначає вихідну.

У свою чергу методи стиснення з втратами поділяються на три додаткові категорії: стиснення з процентними втратами, стиснення з природними втратами та стиснення з неприродними втратами.

Стиснення з процентними втратами – усі втрати знаходяться у межах, коли людське око не бачить різницю між початковим зображенням та декодованим зображенням, що має помилки стиснення з втратами.

Стиснення з природними втратами – втрата великої деталізації зображення є достатньою, тому що люди сприймають об’єкти в природному світі з різними рівнями деталізації, залежно від відстані та місця знаходження. Також, люди звикли до природніх дефектів таких як дощ або туман. Людське око відрізняє грані та лінії зображення. Таким чином, людина сприймає об’єкт, незважаючи на зміни в рівні деталізації.

Стиснення з неприродними втратами – низька якість стиснення, з’являються різні візуальні артефакти, які різко помічаються людським оком.

Більшість відеокодеків, які використовуються, працюють з алгоритмами стиснення з втратами. Під час високого рівня стиснення з’являються проблеми с відображенням граней. Стиснення яке відбувається на основі векторної квантизації, дискретного косинус перетворення або на перетворенні елементарної хвилі працює з растровим зображенням, у якому відсутні поняття краю або лінії.

Простішим методом, який використовують відеокодеки є послідовне кодування довжини.

Послідовне кодування довжини – кодує послідовність пікселів однакового кольору (наприклад, чорного або білого) як одиночне ключове слово. У основі алгоритму є перетворення однієї послідовності на більш малу, не втрачаючи інформації [15]. Цей метод добре працює із зображеннями двійкового рівня (наприклад, чорно-білий текст або графіка) та 8 бітними образами, особливо кадрами мультиплікації, що містять великі послідовності однакового кольору. Послідовне кодування довжини практично не працює з 24-розрядними природними зображеннями, оскільки в цих зображеннях послідовності пікселів одного кольору практично відсутні. Відеокодек, який використовує послідовне кодування довжини: Microsoft RLE. Також схожий алгоритм використовується під час кодування коефіцієнтів у блоках дискретного косинус перетворення.

Векторна квантизація має інтенсивний обчислювальний процес кодування, цей спосіб кодування неможливо використовувати для кодування у реальному часі без

спеціальних апаратних засобів. У основі алгоритму, при векторній квантизації, є зображення, яке ділиться на блоки. Як правило, деякі блоки подібні до інших блоків, хоча зазвичай, вони не ідентичні. Кодер ідентифікує клас подібних блоків і замінює їх на "універсальний" блоковий представник, що становить пошукову таблицю коротких двійкових кодів до "універсальних" блоків. Як правило, найкоротші двійкові коди представляють найбільш загальні класи блоків у зображенні. При векторній квантизації (VQ) декодер використовує пошукову таблицю, щоб транслювати приблизне зображення, складене з "універсальних" блоків згідно з пошуковою таблицею. Відеокодеки, які використовують векторну квантизацію: Indeo 3.2, Сінерак.

Дискретне косинус перетворення (ДКП) широко використовується для стиснення зображень. Двовимірне ДКП застосовується до блоків 8 на 8 пікселів. Дискретне косинус перетворення має великі потреби до обчислювальної потужності, при надвеликому стисненні з'являються блочні артефакти, може виникати розмиття гострих граней. Логіка роботи алгоритму тісно пов'язане з дискретним перетворенням Фур'є і є гомоморфізмом його векторного простору. Це перетворення є лінійним, тому його результат можна обчислити за допомогою множення матриці перетворення та вектору. Відеокодеки, які використовують дискретне косинус перетворення: Motion JPEG, Editable MPEG, MPEG-(1,2,4), H.261, H.263.

Дискретне перетворення елементарної хвилі (ДПЕХ) – проходження потоку через два фільтри (високих на низьких частот). Перед введенням на фільтри, сигнал розбивається на два однакових. Далі ці сигнали зменшуються вдвічі. Параметри фільтрів вибрані так, щоб при складанні сигналів високих та низьких частот відтворювався початковий сигнал. При стисненні відеозображення ДПЕХ може повторюватися кілька разів. Алгоритм на виході має таку ж кількість біт, яка подається на його вхід. Результати на виході називаються коефіцієнтами перетворення. У порівнянні з ДКП, ДПЕХ має менший шанс появи блочних артефактів, сам алгоритм перемагає у швидкості, суб'єктивне порівняння якості при

одному рівні стиснення також має більш високий рівень якості. Відеокодеки: VDOWAVE VDONET'S, VxTreme, Intel Indeo 5.x.

Контур – лінія, що обмежує об'єкт. Текстура – представлення структури поверхні. Контур-засноване кодування зображення представляє зображення як контури, що обмежують області, що є текстурованими [16]. Оскільки контури часто відповідають межах об'єктів у сцені, є близькі відносини між контур-заснованим кодуванням зображення та заснованим на об'єктах (представляє зображення як сукупність об'єктів). Наприклад, як тільки текстури та контури витягуються із зображення, останні можуть бути закодовані як контрольні точки сплайну (поліноміальна функція уявлення кривих). Текстури, у свою чергу, можуть бути закодовані за допомогою ДКП або ДПЕХ.

Зазвичай зміни між сусідніми кадрами є незначними (наприклад, у разі кулі, що летить перед статичним фоном, більша частина зображення не змінюється між кадрами). На цьому заснований алгоритм кодування різниці кадрів. Різниця кадрів кодуються з допомогою алгоритмів стиснення нерухомих зображень (ДПЕХ, ДКП). Ключові кадри стискаються незалежно від попередніх, що обмежує накопичені помилки та уможлиблює пошук у межах потоку відео. Наприклад, у широко використовуваному відеокодеку Сінерак ключовий кадр встановлюється кожні 15 кадрів.

Особливості методу компенсації руху: рівень стиснення вищий аніж у методу різниці кадрів, стадія кодування є інтенсивною з точки зору обчислювальної складової. Розпізнавання об'єктів – невирішена проблема обробки зображення. При стисненні зображення ділиться на блоки (16x16 пікселів в MPEG-1). Для кожного блоку кодується вектор руху, що вказує на блок у попередньому або наступному кадрі, який схожий з кодованим блоком. Блок посилення може співпадати з вихідним (відсутність руху) або відрізнятись від нього (рух). Відеокодеку не потрібно розпізнавати наявність кулі чи іншого об'єкта. Він лише порівнює блоки пікселів у декодованому кадрі та кадрі посилення.

Стиснення досягається шляхом пересилання або збереження вектору руху замість значень пікселів для повного блоку, як показано на рисунку 2.1. Кодовані ("передбачені") блоки формують кадр, що декодується. Блоком посилення може бути будь-який 16x16 блок пікселів у кадрі посилення, який найбільше схожий на кодований блок. Кадр посилення має бути декодований до початку декодування поточного кадру. Однак, кадр посилення не обов'язково повинен йти перед поточним кадром, що декодується. Фактично, кадром посилення може бути майбутній кадр. MPEG враховує це через так званий В (двосторонній передбачуваний) кадр.

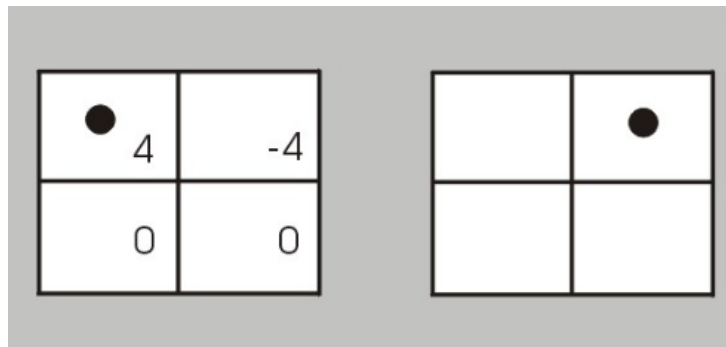


Рисунок 2.1 – Запис вектору під час руху блоку пікселів

Group of Pictures (GOP, група зображень) – упорядкований ланцюжок зображень один за одним у кодованому потоці відео [17]. Кадри об'єднуються в групи з метою міжкадрової компресії, без якої передача потоку відео через мережу витрчала б величезний час і трафік. Стиснутий потік являє собою набір груп зображень, які розташовані один за одним. На стороні одержувача декодер складає видимі кадри з кадрів, об'єднаних у групи зображень. Група зображень складається з І-кадру та наступних за ним Р- та В-кадрів.

І-кадр – перший кадр у групі. Має повне зображення, яке стискається незалежно від інших кадрів;

Р-кадр – має різницю між зображенням на попередньому Р-кадрі та зображенні на поточному кадрі;

В-кадр – має посилання на сусідній кадр (на один або декілька І та Р кадрів), що дозволяє прискорити звернення потоку при перемотуванні.

Розмір GOP (відстань у кадрах між сусідніми ключовими кадрами) – кількість кадрів в одному GOP. Розмір GOP у потоку буває змінним та постійним. У нових проектах, представлених в H.264, MPEG-4, кодери мають набагато більшу гнучкість при зверненні до структур. Вони можуть використовувати ті ж структури посилань, які раніше використовувалися в більш старих проектах, або вони можуть використовувати більше зображень як посилання, і вони можуть використовувати більш гнучкий порядок черги кодування щодо порядку відображення.

## 2.2 Відеокодеки

Усі розглянуті методи та алгоритми для стиснення відеоматеріалів можна використовувати окремо, але їх поєднання дає змогу перекрити недоліки одних та підсилити переваги інших. Якщо використовувати алгоритми разом необхідно розумно підходити до умов їх застосування та обирати момент коли це більш доцільно. Для вирішення цієї задачі була створена велика кількість готових відеокодеків, які використовують наведені методи та алгоритми для кодування та декодування відеоматеріалів.

MPEG-1 (Moving Picture Experts Group) – дуже популярний стандарт у всьому світі, з основою, яка була взята від відеокодека JPG. Стиснення проводиться серіями по три кадри. Це один із найстаріших відеокодеків, практично на будь-яких системах можна переглянути відео зі стереозвуком у цьому форматі.

MPEG-2 – стандарт для стиснення, який є не революційним, а скоріше еволюційним форматом, виникнувши в результаті переробки MPEG-1 під потреби замовників.

Стандарт MPEG-2 навмисно не визначає способи компресії зображення, він лише вказує, як має бути оформлене стиснене зображення. Стандарт не визначає, як повинен бути реалізований кодер або декодер MPEG-2, він визначає лише структуру даних. Це дає можливість учасникам ринку конкурувати один з одним за створення якісніших пристроїв та алгоритмів. MPEG-2 – це потоковий стандарт, який спочатку призначався для пересилання супутниковими каналами або кабельними мережами. У основі алгоритму кодування лежать групи зображень GOP.

MPEG-4 – міжнародний стандарт, який використовується переважно для стиснення цифрового аудіо та відео [18]. MPEG-4 включає багато функцій MPEG-1, MPEG-2 та інших схожих стандартів, додатковими функціями є: підтримка мови віртуальної розмітки VRML для відображення 3D-об'єктів, об'єктно-орієнтовані файли, підтримка управління правами та різні типи інтерактивного медіа.

Ефективне та оптимальне уявлення об'єктів означає не лише досягнення високого ступеня компресії, а й відповідність іншим вимогам, наприклад, захищеність до перешкод, довільний доступ, простота редагування.

Дані MPEG-4 можуть упаковуватися в пакети транспортного потоку MPEG-2, які передаються в інфраструктурі мереж телебачення. Дані MPEG-4 можуть бути упаковані в пакети протоколу реального часу.

Існує багато різних додатків, у яких доцільно та вигідно використовувати об'єктно-орієнтовані засоби MPEG-4. Ці програми: мовлення, дистанційне спостереження, персональні комунікації, ігри, мобільні мультимедійні системи зв'язку, системи віртуальної реальності.

У основі алгоритму стиснення MPEG-4 є модель сприйняття людським оком відеозображення та особливості будови людського ока: здатність сприймати варіації кольору та градації яскравості. Відомо, що людське око здатне краще сприймати градацію яскравості, ніж кольоровості.

Завдання зводиться до визначення нерухомого фону та рухомих об'єктів на екрані, на підставі цих даних можна виділити і передати інформацію о базових кадрах,

а далі передати кадри з інформацією о рухомих об'єктах. У процесі передачі даних відбувається видалення малозначущої інформації, аналогічної принципам, які використовує графічний формат JPEG.

H.263 – це стандарт стиснення відео, розроблений Групами експертів з кодування відео (VCEG) у 1996 році як удосконалення в сімействі H26x. Він був призначений для функцій відеоконференцій із низьким бітрейтом.

До H.264 велика частина потокового вмісту, доступного в Інтернеті, базувалася на відеокодеку H.263. H.263 також використовується в IP-мультимедійній підсистемі (IMS), службі мультимедійних повідомлень (MMS) та прозорій наскрізній пакетній комутаційній службі потокової передачі. Він також використовується у файловому контейнері 3gp, призначеному для мобільних пристроїв.

H.264 – міжнародний стандарт стиснення відео, розроблений групою спеціалістів організації ITU. Стандарт кодування H.264 впроваджуватися ще з 2003 року. На той час на зміну масовому DVD формату і широко використовуваному кодеку MPEG2 необхідний був ефективний засіб стиснення відео. Адже починалася ера масового переходу на HD формат.

Маючи більш сучасні та прогресивні аналоги кодування відео, стандарт H.264 є популярним у сучасності [19]. Адже впровадження нових і несумісних з H.264 технологій кодування вимагає використання більш потужної, енергоємної та дорогої електроніки. Також при використанні нових стандартів потребується затрат на покупку патентів, перебудову виробничих процесів, що у свою чергу викликає збільшення ціни для кінцевого користувача.

Для поліпшення передбачення у стандарті H.264 реалізовані такі основні технології:

- компенсація руху на основі малих блоків адаптивно налаштованого розміру;
- точність компенсації руху до відліку;
- компенсація руху на основі одного або декількох опарних кадрів;
- можливість використання будь-якого кадру в якості опорного;

– циклічна фільтрація для ліквідації ефекту блоковості.

Алгоритм кодування складається з чотирьох основних компонентів:

- компенсація руху і віднімання поточного кадру з опорного;
- дискретне косинус перетворення різницевого кадру;
- квантування коефіцієнтів перетворення;
- ентропійне кодування квантованих коефіцієнтів.

Відеокодек нового покоління High Efficiency Video codec (HEVC), відомий також як H.265 [20]. Стандарт H.265 став значним кроком вперед в області кодування відео. Одна з його переваг в тому, що він подвоює ефективність стиснення H.264. Так що при передачі зображень аналогічної якості H.265 використовує тільки половину бітрейта попереднього відеокодека. Завдяки цьому вимоги до пропускної здатності і зберігання різко скорочуються, що дозволяє більш вигідно використовувати і апаратні, і програмні засоби. На графіку ефективності кодування HEVC, який наведено на рисунку 2.2, можна побачити, що бітрейт нижче на 25-35% при еквівалентній якості (HD).

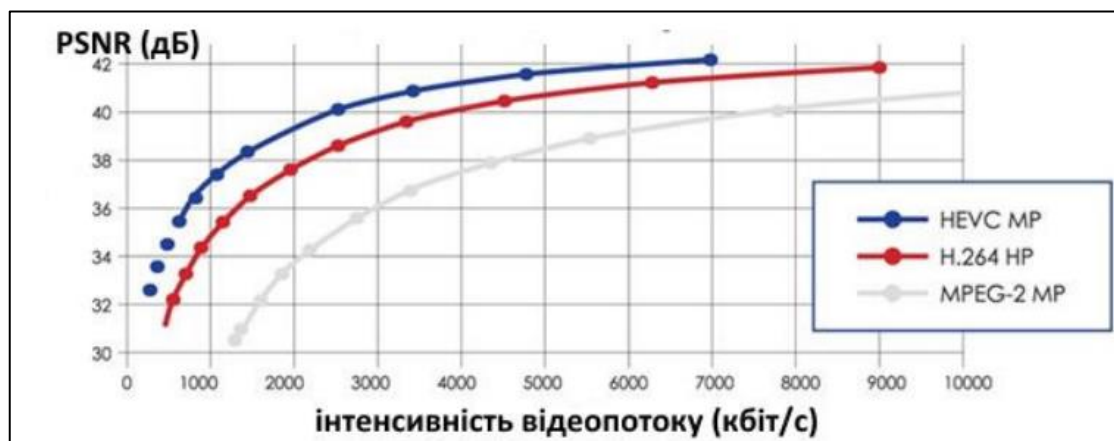


Рисунок 2.2 – Ефективність кодування H.265

Існує складний баланс між якістю отриманого відео, складністю алгоритмів кодування та декодування, кількістю інформації, необхідної для його відтворення.

## 2.3 Методи клієнт-серверної взаємодії у реальному часі

Незважаючи на очевидну перешкоду, потреба у спілкуванні в реальному часі не є новою [21]. І роками намагаються обійти блокпости HTTP цікавими способами.

RTC (real-time communication) – спілкування у режимі реального часу, є цінним для інформації, яка є чутливою до часу, або просто для підвищення швидкості передачі ідей [22]. Ось чотири конкретні способи, якими режим реального часу може допомогти бізнесу:

- вільний доступ: RTC надає особам миттєвий доступ до інших людей та додаткової інформації без затримок. Це призводить до підвищення ефективності та економії коштів, а також прискорення виконання завдань;

- швидкість: з методами зв'язку, які не пов'язані з RTC, наприклад, електронною поштою, особа повинна дочекатися відповіді, перш ніж продовжити, також їй може знадобитися надіслати кілька електронних листів, перш ніж вона отримає всю інформацію;

- віддалений доступ: у наш час, люди можуть отримати доступ до інформації з багатьох пристроїв, і до цих мереж можна додати більше людей, щоб підвищити креативність і швидкість розробки рішень;

- безпека: можна спілкуватися на великих дистанціях, без необхідності очної зустрічі.

Технологія RTC перетворила світ на глобальне суспільство. Тепер можна спілкуватися та обмінюватися інформацією швидко, на великі відстані, як ніколи раніше. Можна зробити це так, ніби інша людина знаходиться в одній кімнаті з користувачем.

Дані надсилаються безпосередньо і миттєво від відправника до одержувача і не зберігаються на шляху до місця призначення. Телефон є лише одним із класичних прикладів спілкування в реальному часі.

Це на відміну від зв'язку зі зміщенням у годину, коли ми будемо доставлені на час, і це буде певний період для їх отримання, а також інший період, перш ніж ми отримаємо відповідь. Пошта равликів і електронна пошта є чудовими прикладами такого спілкування.

Зв'язок у режимі реального часу з'єднує користувачів без будь-яких суттєвих затримок [23].

Технології, якими частіше користуються наведені у таблиці 2.1 [24].

Таблиця 2.1 – Порівняння технологій RTC

	Regular pooling	Long pooling	Server sent events	Forever frames	WebSockets	WebRTC
Постійне з'єднання	Ні	Ні	Клієнт слухає сервер	Так	Так	Так
Використовує захищені канали зв'язку	Так	Так	Ні	Ні	Так	Так
Легка інтеграція	Так	Так	Ні	Ні	Так	Ні
Витрати серверної пам'яті	Ні	Ні	Так	Так	Ні	Ні
Перенавантаження сервера запитами	Так	Так	Ні	Ні	Ні	Ні

Під час звичайного опитування (regular polling) клієнт багаторазово запитує сервер, чи є якась нова інформація, яку потрібно надати, як показано на рисунку 2.3. Інтервал запитів може змінюватися залежно від потреб програми, і сервер переважно

відповідає без нової інформації. Це явно не в реальному часі, але може мати певні переваги, якщо інтервал опитування буде коротким.

Проблема з опитуванням полягає в тому, що клієнт повинен постійно запитувати у сервера будь-які нові дані. В результаті багато відповідей порожні, створюючи накладні витрати HTTP.

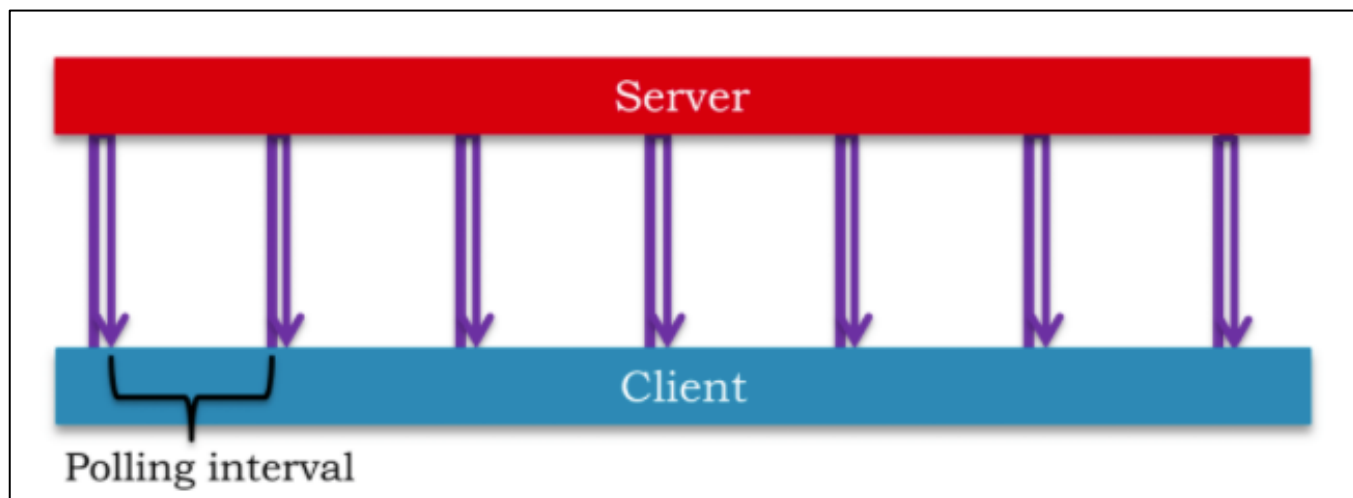


Рисунок 2.3 – Опитування сервера

Зловживання пропускною спроможністю не викликає великої проблеми, оскільки більшість запитів та відповідей не містять багато даних, але регулярні опитування включає непотрібну взаємодію між клієнтом-сервером.

Тривале опитування – це розширена форма опитування, що забезпечує спілкування в реальному часі, як показано на рисунку 2.4. Клієнт робить запит до сервера для отримання інформації. Сервер просто перебуває на запиті, доки не станеться щось важливе, або доки запит не закінчиться. Якщо сервер відповідає, щоб запобігти тайм-ауту, клієнт може бути запрограмований на те, щоб негайно запустити інший запит.

Як тільки дані стають доступними, клієнту надсилається повна відповідь. Потім клієнт негайно повторно запитує інформацію від сервера, так що сервер майже завжди

матиме доступний запит на очікування, який він може використовувати для доставки даних у відповідь на подію.

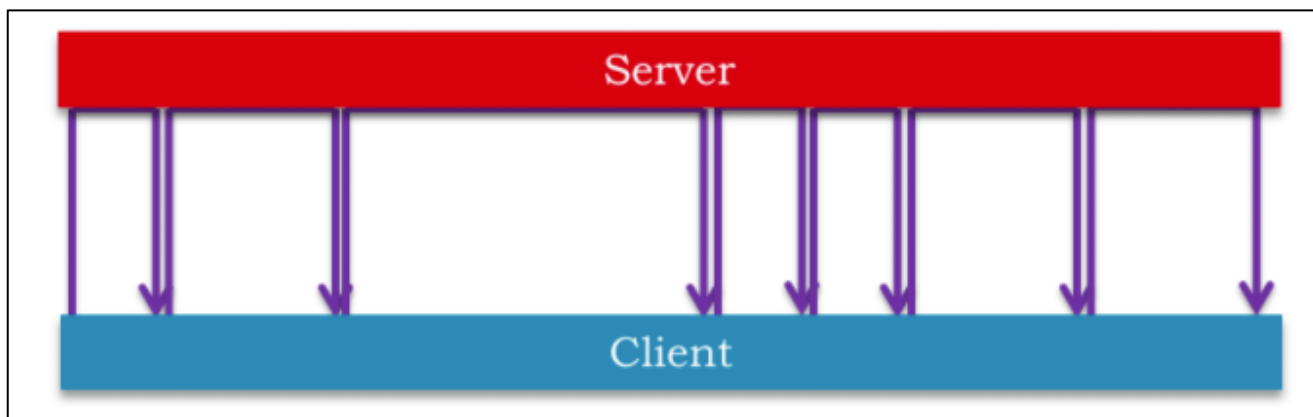


Рисунок 2.4 – Приклад тривалого опитування

Це гарантує, що сервер завжди має відкритий запит на відповідь і може передавати інформацію в режимі реального часу. Незважаючи на те, що тривале опитування залишається популярним, для його успішного впровадження часто необхідна спеціальна реалізація як на рівні сервера, так і на рівні клієнта.

Server sent events – полегшує спілкування в реальному часі, але в основному є односпрямованим поштовхом від сервера до клієнта. Об'єкт EventSource (стандартизований у HTML5) використовується на стороні клієнта для захоплення поточкових сповіщень із сервера, як показано на рисунку 2.5. Якщо клієнт хоче надіслати дані на сервер, йому знадобиться інша технологія або протокол для цього. Server sent events найкраще підходить, коли нам потрібен трафік у реальному часі від сервера до клієнта або якщо сервер генерує дані в циклі та надсилає клієнту кілька подій.

Forever Frame – це здебільшого специфічна для ІЕ техніка, яка допомагає у спілкуванні в реальному часі, і включає прихований iFrame на стороні клієнта, який підтримує з'єднання із сервером за допомогою сценаріїв. Сервер надсилає фрагменти

інформації, яка отримується в iFrame і керується спеціальним кодом на стороні клієнта.

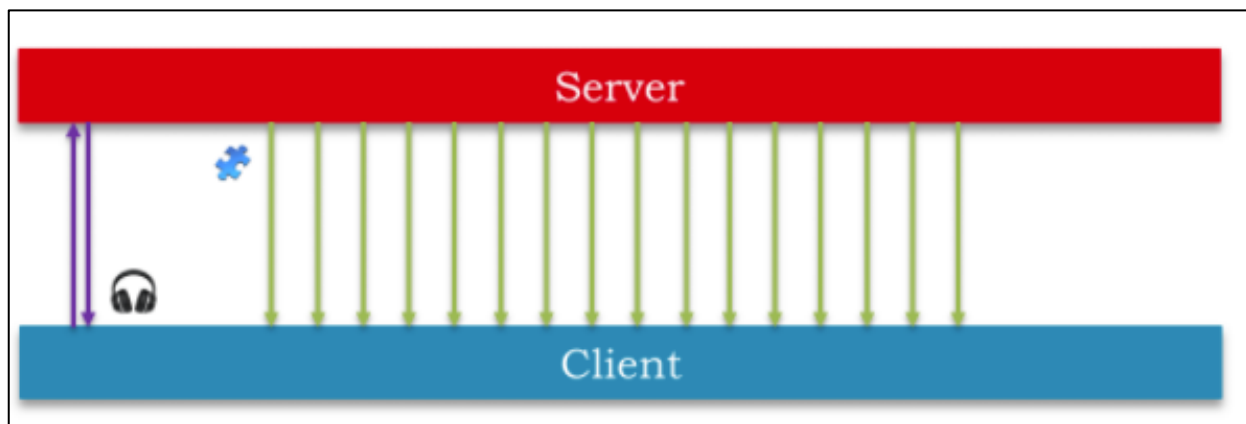


Рисунок 2.5 – Server sent events взаємодія

Завдяки стандартизації HTML5 WebSockets з'явилися як чудова техніка для полегшення спілкування в реальному часі, як показано на рисунку 4.6.

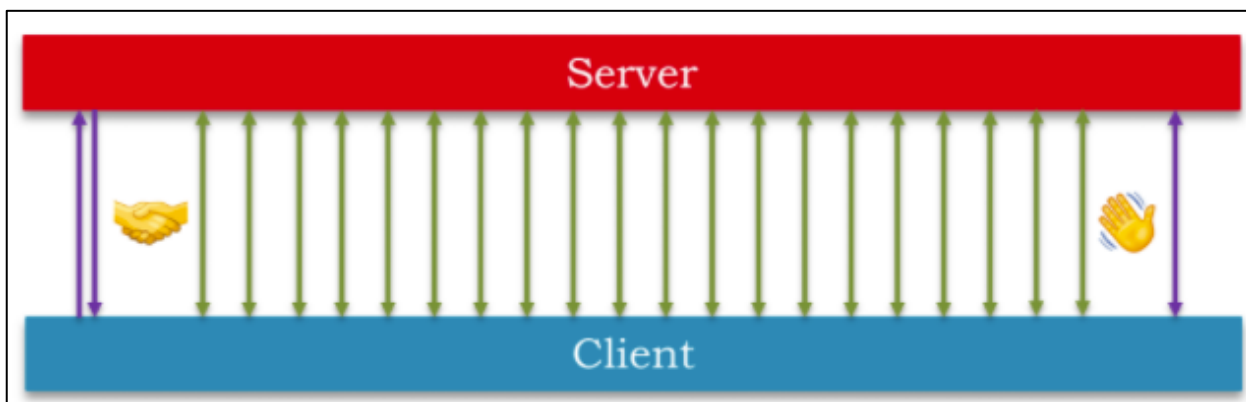


Рисунок 2.6 – WebSockets взаємодія

На початку сервер і клієнт домовляються, щоб переконатися, що вони можуть взаємодіяти через WebSockets.

Після завершення переговорів WebSockets забезпечують справжнє постійне з'єднання для двостороннього зв'язку – сервер може викликати клієнта, а клієнт може викликати сервер у будь-який час.

WebSockets тепер користуються широкою підтримкою платформ, що дозволяє обмінюватися інформацією в реальному часі з низькою затримкою. WebSocket – це найбільш кардинальне розширення протоколу HTTP з його появи. За допомогою WebSockets також можна передавати і бінарні дані.

Для вирішення поставленої задачі, краще за все використовувати WebSockets, оскільки цей метод найкраще підходить для отримання подій та не перевантажує систему зайвими перевітками, також цей метод має переваги у налагодженні передачі бінарних даних.

### 3 МЕТОДИ ОЦІНЮВАННЯ ЯКОСТІ ВІДЕОМАТЕРІАЛУ

Потік відео – це послідовність зображень. У більшості методів якість відео оцінюється як середнє значення серед всіх кадрів, тобто порівняння спотворень, які може має кожен кадр, всіх зображень. Припускається, що якість відеоматеріалу – це середнє значення показників якості усіх кадрів.

Користувач помічає частіше зміни якості, тобто відео може буди дратливим, якщо має стрибки. Більш вигідною є версія, яка має постійну якість, але з більш низьким бітрейтом.

Методи оцінювання якості поділяють на об'єктивні та суб'єктивні методи. Більшості об'єктивних методів працює за принципом вимірювання кадрової різниці між вхідним сигналом і сигналом на виході системи. Суб'єктивні методи, в свою чергу, базуються на думці експертів, що не завжди дає точні результати оцінювання якості [25].

Об'єктивні методи класифікуються згідно з корисністю вихідного відео сигналу, для якого забезпечується висока якість. Тому вони класифікуються за трьома категоріями: повні референсні методи, скорочені референсні методи і неререференсні методи.

Найтрадиційнішим методом вимірювання якості системи обробки цифрового відео є вимірювання відношення сигнал/шум та пікове відношення сигнал/шум між вхідним сигналом і сигналом на виході системи. PSNR – це одна з метрик об'єктивної якості відео [26]. Вона може бути автоматично обчислена комп'ютерною програмою. Але хороший показник PSNR не свідчить про хорошу якість, це пов'язано з тим, що зорова система людини має нелінійну характеристику.

PSNR (пікове співвідношення сигналу до шуму) – метрика, що найчастіше використовується на практиці. Використання саме цієї метрики обумовлено її популярністю, її використовують у багатьох наукових статтях і порівняннях як

запобіжний втрат якості. В цієї метрики, як в інших свої переваги і недоліки [27]. Чим більше значення метрики, тим більша різниця між порівнюваними зображеннями.

За суттю ця методика близька до середньоквадратичної помилки, але більш зручна завдяки логарифмічній шкалі, формула для розрахунку пікового співвідношення сигналу до шуму має вигляд:

$$PSNR = 10 \log_{10} \left( \frac{MAX_1^2}{MSE} \right) = 20 \log_{10} \left( \frac{MAX_1}{\sqrt{MSE}} \right)$$

де  $MAX_1$  – максимальне значення, що може бути прийняте пікселем зображення;

$MSE$  – середньоквадратичне відхилення для двох монохромних зображень  $I$  та  $K$  розміром  $m \times n$ , одне з яких вважається зображенням з шумом, яке наближено до іншого.

Середнє квадратичне відхилення – у теорії ймовірностей і статистиці один із найпоширеніших показників розсіювання (розкиду) значень випадкової величини відносно її математичного сподівання, тобто центру розподілу. Середньоквадратичне відхилення для двох монохромних зображень має наступний вигляд:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i,j) - K(i,j)|^2$$

Індекс структурної схожості (SSIM – structure similarity) SSIM оснований на ідеї, що людська зорова система пристосована для обробки структурної інформації. Тому алгоритм розроблено для вимірювання зміни в цій інформації між еталонним і спотвореним зображенням. Особливістю є те, що результат завжди лежить в проміжку від -1 до 1, причому якщо його значення дорівнює 1, це означає, що ми маємо дві

однакові картинки. Значення 0 означає відсутність кореляції. Тому такий метод є найдоцільнішим у використанні [28].

Система оцінки SSIM складається із трьох частин: структурної, оцінки яскравості та контрастності. В першу чергу порівнюється складова яскравості, яка розраховується як середня інтенсивність сигналу, та для відеозображення  $S$  розміром  $M \times N$  обчислюється за формулою:

$$\mu = \frac{1}{NM} \sum_{i=0}^M \sum_{j=0}^N S(i, j)$$

Далі необхідно середню інтенсивність відняти із сигналу для того, щоб отримати складову контрастності. Оцінкою даної складової є варіація, яка для сигналу  $(S(i, j) - \mu)$  обчислюється так:

$$\delta = \sqrt{\left( \frac{1}{(N-1)(M-1)} \sum_{i=0}^M \sum_{j=0}^N (S(i, j) - \mu)^2 \right)}$$

Таким чином кінцева оцінка схожості двох відеозображень  $x$  та  $y$  буде наступною:

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$$

де  $l(x, y)$  – функція порівняння складових яскравості  $x$  та  $y$ ;

$c(x, y)$  – функція порівняння складових контрастності  $x$  та  $y$ ;

$s(x, y)$  – функція порівняння складових структурності  $x$  та  $y$ .

Функція, яка порівнює значення складової яскравості виглядає наступним чином:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1},$$

$$c_1 = (k_1L)^2$$

де  $\mu_x$  – середнє значення  $x$ ;

$\mu_y$  – середнє значення  $y$ ;

$c_1$  – змінна, яка необхідна для стабілізації значення внаслідок незначного знаменника;

$L$  – динамічний діапазон пікселів;

$k_1$  – константа, яка дорівнює 0.01.

Функція, яка порівнює значення складової контрастності визначається наступним чином:

$$c(x, y) = \frac{2\delta_x\delta_y + c_2}{\delta_x^2 + \delta_y^2 + c_2},$$

$$c_2 = (k_2L)^2$$

де  $\delta_x$  – варіація  $x$ ;

$\delta_y$  – варіація  $y$ ;

$c_2$  – змінна, яка необхідна для стабілізації значення внаслідок незначного знаменника;

$k_2$  – константа, яка дорівнює 0.03.

Функція, яка порівнює значення складової контрастності визначається наступним чином:

$$s(x, y) = \frac{\delta_x \cdot \delta_y + c_3}{\delta_x + \delta_y + c_3},$$

$$c_3 = \frac{c_2}{2}$$

Якщо об'єднати наведені формули то можна отримати формулу для розрахунку індексу структурної схожості:

$$SSIM = \frac{(2\mu_x \cdot \mu_y + c_1) \cdot (2\delta_x \cdot \delta_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) \cdot (\delta_x^2 + \delta_y^2 + c_2)}$$

Головною метою великої кількості об'єктивних метрик оцінки якості є автоматична оцінка передбачуваного сприйняття користувачами обробленого системою відео [29]. Але найкращим способом визначити думку користувачів це просто запитати в них. Проте іноді, суб'єктивне вимірювання якості відео є важким завданням, оскільки вимагає досвідчених експертів для їх оцінки.

Суб'єктивні вимірювання передбачають оцінку якості зображень спостерігачами. Ці вимірювання завжди є прямими, оскільки думка глядачів про якість відтворення випробувальних сигналів або таблиць є недоречною. При застосуванні методів суб'єктивної оцінки, людина виступає вимірювальним приладом для визначення якості системи відображення даних. Це стосується того, як глядач сприймає відео, і визначає думку щодо певної послідовності відео. Суб'єктивні тести якості відео досить дорогі з точки зору часу (підготовки та запуску) та людських ресурсів.

Оглядові дослідження застосовуються для вивчення почуттів та реакцій глядачів з незаангажованою позицією, по відношенню до новітніх технологій, таких як 3D. Приклад дослідницького вивчення – фокус-група, де неупереджені глядачі, розділені на малі групи, в яких обговорюють свій досвід перегляду матеріалів з певної

системи формування зображень. Результатом є описані цими глядачами відчуття та «переживання» після перегляду 3D матеріалів.

У випадку з відеокодеками це дуже поширена ситуація, коли відеокодеки з подібними об'єктивними результатами показують результати з різними суб'єктивними результатами. Наприклад, «до» та «після» фільтри, які можуть підвищувати якість з боку глядача (розмиття заднього фону, для поліпшення передньої картинки), але давати погіршення з боку суб'єктивної оцінки.

Важко використовувати довгі послідовності для суб'єктивного тестування. Зазвичай використовують три чи чотири послідовності по 10 секунд, тоді як цілі фільми використовуються для об'єктивних метрик.

## 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

### 4.1 Програмне забезпечення

Для реалізації експерименту було обрано додаткові застосунки для створення заготовок відеозаписів.

Movavi Video Converter 19 Premium – застосунок, який дозволяє конвертувати відео файли з різним бітрейтом, що дозволяє керувати та налагоджувати розмір та якість відеоматеріалу. На рисунку 4.1 можна побачити приклад інтерфейсу.

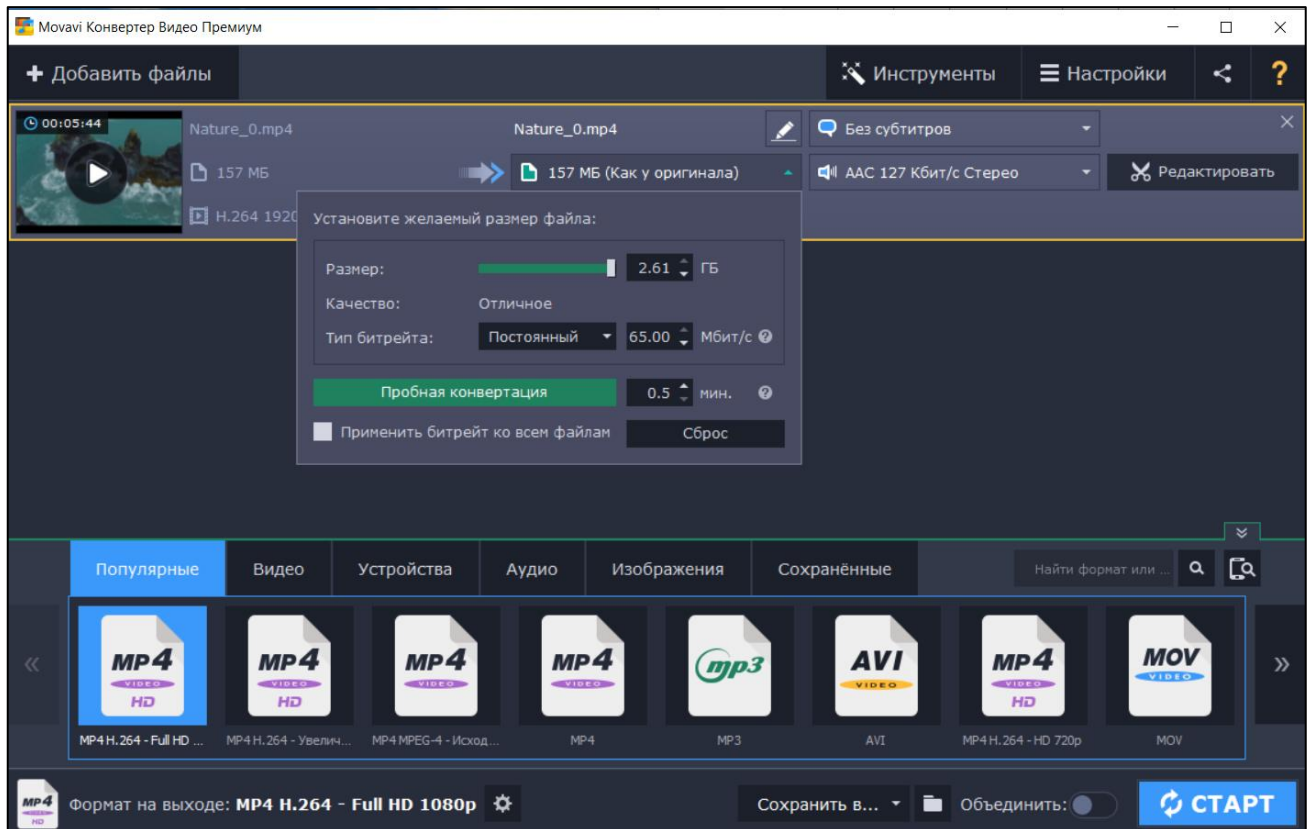


Рисунок 4.1 – Інтерфейс застосунку Movavi Video Converter 19 Premium

Застосунок може зберігати відео у якості Full HD та Ultra HD. Також застосунок має велику кількість сучасних відеокодеків та форматів відео. Застосунок має готові налагодження для мобільних пристроїв та сучасних соціальних мереж.

VideoMaster 11.0 – схожий застосунок для конвертування відео файлів.

На рисунку 4.2 зображено приклад інтерфейсу головного вікна VideoMaster 11.0.

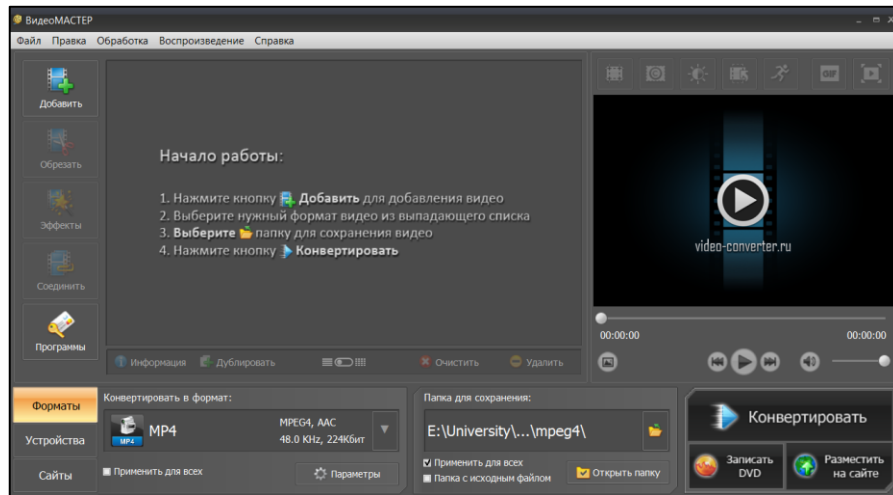


Рисунок 4.2 – Інтерфейс головного вікна VideoMaster 11.0

На рисунку 4.3 зображено вікно налагодження бітрейту, формату та вибору відеокодеку для конвертації відеозапису.

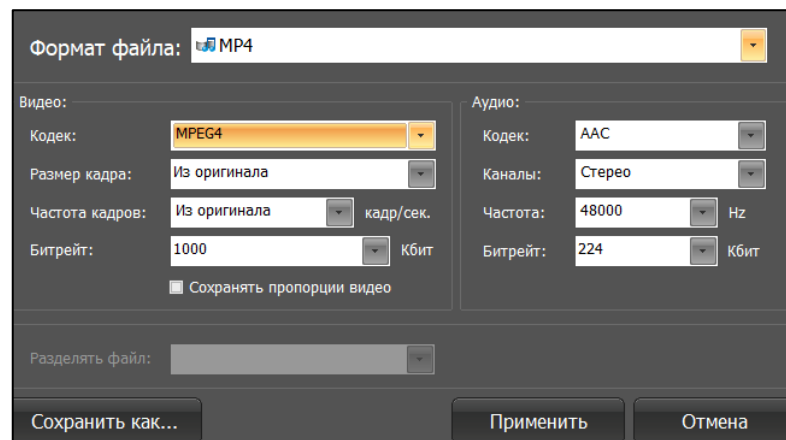


Рисунок 4.3 – Вікно налагодження відео

Цей застосунок має більшу кількість застарілих відеокодеків та форматів, які не застосовуються у наш час або застосовуються у деяких окремих випадках. У обох

програмах можна використовувати функцію попереднього перегляду та конвертувати одразу декілька відеоматеріалів одночасно.

Також було застосовано Microsoft Office Excel, яке було необхідно для побудови діаграм, на основі оцінки якості відеоматеріалів. За допомогою застосунку можна легко апроксимувати точкові діаграми. Головний інструмент, за допомогою якого проводиться згладжування в Excel – це побудова лінії тренду. Суть полягає в тому, що на основі вже наявних показників добудовується графік функції на майбутні періоди.

На рисунку 4.4 вказано приклад налагодження апроксимації результатів за допомогою логарифмічної лінії тренду.

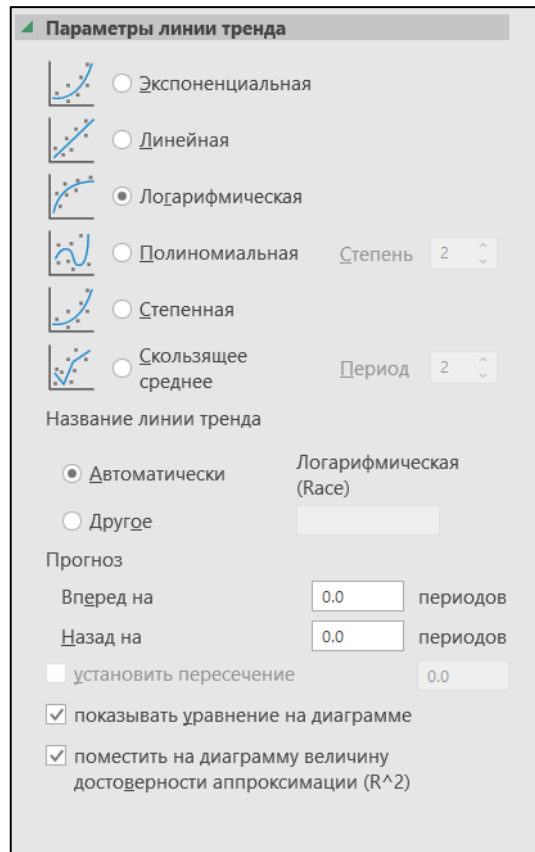


Рисунок 4.4 – Налаштування лінії тренду

Логарифмічною апроксимацією варіант краще використовувати тоді, коли спочатку дані швидко змінюються, а потім приймають збалансований вигляд

## 4.2 Реалізація коду для оцінки якості відеоматеріалів

У ході експерименту з'явилась необхідність у розрахунку якості відеоматеріалу, використовуючи індекс структурної схожості. Для цього було розроблено програмний застосунок, який реалізує алгоритм розрахунку індексу структурної схожості на мові програмування Python. Цей код завантажує відеофайл з вказаного каталогу, після чого поділяє весь потік відео на кадри та порівнює з оригіналом, у відповідь дає результат у вигляді діаграм. На рисунку 4.5 можна побачити приклад коду для розрахунку індексу структурної схожості:

```
def ssim(img1, img2, cs_map=False):
    img1 = img1.astype(numpy.float64)
    img2 = img2.astype(numpy.float64)
    size = 11
    sigma = 1.5
    window = gauss.fspecial_gauss(size, sigma)
    K1 = 0.01
    K2 = 0.03
    L = 255 # bitdepth of image
    C1 = (K1*L)**2
    C2 = (K2*L)**2
    mu1 = signal.fftconvolve(window, img1, mode='valid')
    mu2 = signal.fftconvolve(window, img2, mode='valid')
    mu1_sq = mu1*mu1
    mu2_sq = mu2*mu2
    mu1_mu2 = mu1*mu2
    sigma1_sq = signal.fftconvolve(window, img1*img1, mode='valid') - mu1_sq
    sigma2_sq = signal.fftconvolve(window, img2*img2, mode='valid') - mu2_sq
    sigma12 = signal.fftconvolve(window, img1*img2, mode='valid') - mu1_mu2
    if cs_map:
        return (((2*mu1_mu2 + C1)*(2*sigma12 + C2))/((mu1_sq + mu2_sq + C1) *
                                                    (sigma1_sq + sigma2_sq + C2)),
                (2.0*sigma12 + C2)/(sigma1_sq + sigma2_sq + C2))
    else:
        return ((2*mu1_mu2 + C1)*(2*sigma12 + C2))/((mu1_sq + mu2_sq + C1) *
                                                    (sigma1_sq + sigma2_sq + C2))
```

Рисунок 4.5 – Реалізація методу для розрахунку індексу структурної схожості

Також програма розраховувала середній індекс схожості та записувала до файлу під час усіх експериментів.

### 4.3 Опис та підготовка експерименту

До початку експерименту було обрано чотири категорії відеозаписів. Кодові назви та опис наведено у таблиці 4.1.

Таблиця 4.1 – Опис обраних відеозаписів

Кодова назва	Опис
Coding	Приклад демонстрації екрану під час написання коду, відеозапис має статичну картинку на якій частково змінюються символи та програми.
Interview	Репортер бере інтерв'ю, відеозапис має нерухому картинку, яка іноді змінює ракурс.
Nature	Приклади природи. Відео має повільне масштабування, але з великою кількістю дрібних деталей.
Race	Приклад гонки з фільму. Велика кількість різноманітних кольорів, дуже часто змінюється ракурс, фон та об'єкти перебувають в динамічному русі.

Для дослідження відеоматеріали були скорочені до 10 секундної тривалості. Відео послідовності були декодовані та закодовані з використанням наступних відеокодеків: MPEG-2, MPEG-4, H.263, H.265. Також, використовуючи кожен формат, було створено відеозаписи з різною бітовою швидкістю, а саме: 1 Мбіт/с, 3 Мбіт/с, 5 Мбіт/с, 8 Мбіт/с, 10 Мбіт/с, 20 Мбіт/с, 30 Мбіт/с, 40 Мбіт/с, 50 Мбіт/с. В цілому, було створено 144 заготовок.

У таблиці 4.2 вказано, як змінювався розмір файлу під час змінення бітової швидкості для різних видів відеокодеку.

Таблиця 4.2 – Зміна розміру файлу під час зміни бітової швидкості

Кодова назва відео	Бітова швидкість (Мбіт/с)	H.263 (розмір, Мбіт)	H.265 (розмір, Мбіт)	MPGE-2 (розмір, Мбіт)	MPEG-4 (розмір, Мбіт)
Nature	Оригінал	35.61623	35.61623	35.61623	35.61623
	50	31.09528	31.66368	31.54688	31.59215
	40	31.09528	25.14644	31.54688	31.59215
	30	25.35545	18.66159	25.22461	19.57764
	20	19.1377	12.26033	12.76953	19.57764
	10	6.58971	6.012036	6.53125	13.21636
	8	6.58971	6.012036	5.291016	6.866876
	5	5.349195	4.793689	5.291016	5.55858
	3	3.484833	1.803745	3.435547	3.63439
	1	2.221222	1.803745	2.591797	2.299603
Coding	Оригінал	2.365231	2.365231	2.365231	2.365231
	50	2.288284	0.697739	2.736328	2.247496
	40	2.288284	0.696475	2.736328	2.247496
	30	2.288284	0.69559	2.736328	2.247496
	20	2.288284	0.696358	2.736328	2.247496
	10	2.288284	0.681315	2.736328	2.247496
	8	2.288284	0.68202	2.736328	2.247496
	5	2.288284	0.682269	2.736328	2.247496
	3	2.288284	0.649037	2.736328	2.247496
	1	2.288214	0.649037	1.580078	2.247496
Race	Оригінал	19.18815	19.18815	19.18815	19.18815
	50	4.959238	18.23644	5.603516	4.97006
	40	4.959238	18.12304	5.603516	4.97006

Кінець таблиці 4.2

Кодова назва відео	Бітова швидкість (Мбіт/с)	H.263 (розмір, Мбіт)	H.265 (розмір, Мбіт)	MPGE-2 (розмір, Мбіт)	MPEG-4 (розмір, Мбіт)
Race	30	4.959238	17.29423	5.603516	4.97006
	20	4.959238	13.46735	5.603516	4.97006
	10	4.959238	7.222603	5.603516	4.97006
	8	4.959238	5.818168	5.525391	4.97006
	5	3.705074	3.679658	3.712891	3.717514
	3	2.519196	2.273201	2.734375	2.529832
	1	2.206608	0.883846	2.503906	1.581104
Interview	Оригінал	20.51918	20.51918	20.51918	20.51918
	50	2.637419	19.27972	3.207031	2.629239
	40	2.637419	19.08356	3.207031	2.629239
	30	2.637419	18.05781	3.207031	2.629239
	20	2.637419	12.66222	3.207031	2.629239
	10	2.637419	6.35741	3.207031	2.629239
	8	2.637419	6.358505	3.207031	2.629239
	5	2.637419	5.091026	3.207031	2.629239
	3	2.637419	1.965177	3.207031	2.629239
	1	2.281769	1.965177	1.248047	2.286747

Усі відеоматеріали було запущено у створеному програмному застосунку, який розрахував індекс структурної схожості набору кадрів до того ж самого набору оригіналу.

Результатом створеного застосунку також є середнє арифметичне якості для кожного відеозапису.

Наступник кроком вся інформація була внесена до застосунку Microsoft Office Excel для оцінки результатів та пошуку залежності.

На рисунку 4.6 можна побачити графіки залежності розміру файлу від обраного відеокодеку та від обраної бітової швидкості.

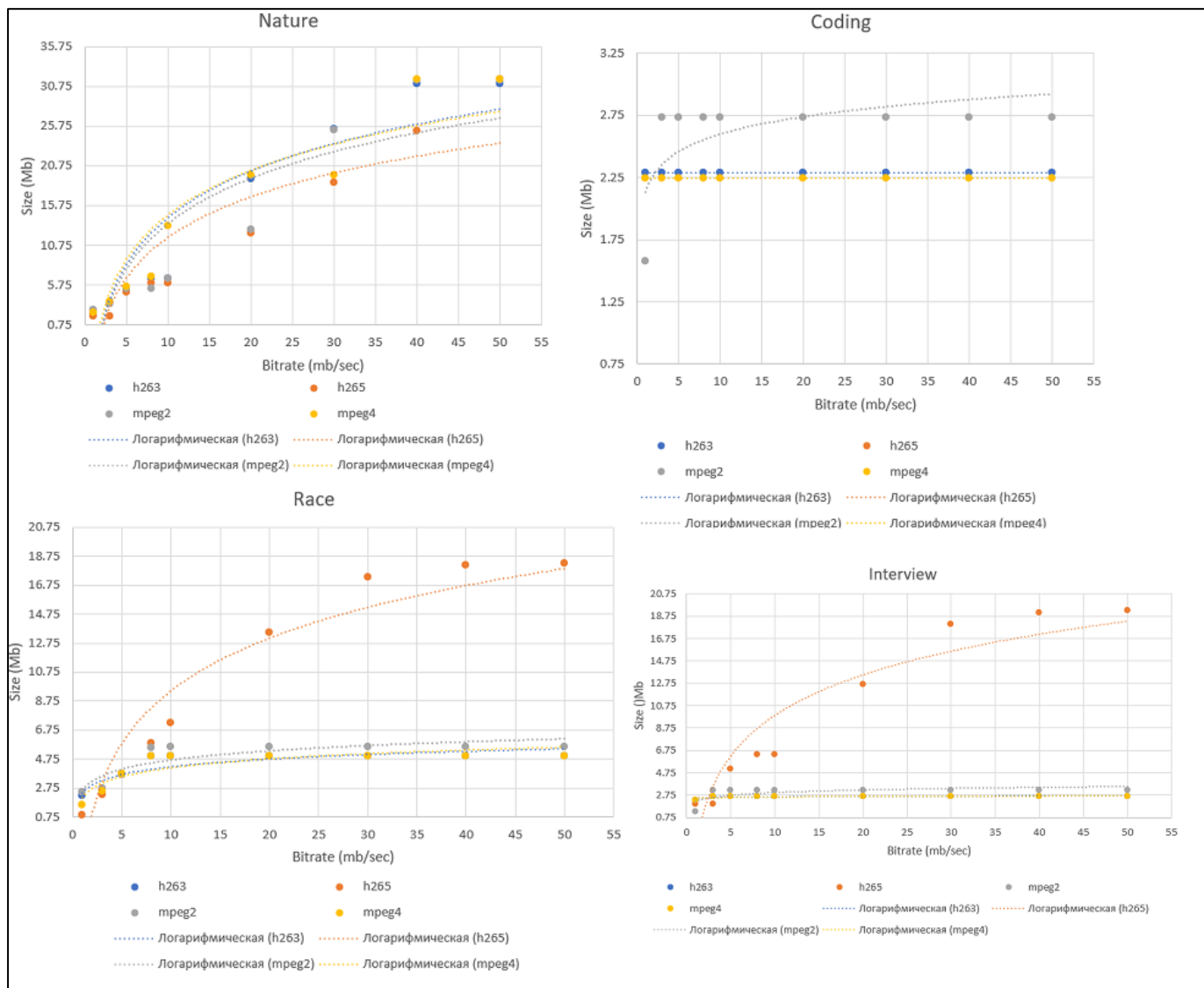


Рисунок 4.6 – Графіки залежності розміру від відеокодеку та бітової швидкості

Вісь абсцис має значення бітової швидкості від 1 до 50 Мбіт/с. Вісь ординат має значення розміру файлу (Мбіт) відповідно кожній обраній категорії відеофайлу.

В таблиці 4.3 можна побачити залежність індексу структурної схожості від бітової швидкості для обраних відеокодеків.

Таблиця 4.3 – Зміна індексу структурної схожості відео під час зміни бітової швидкості

Кодова назва відео	Бітова швидкість (Мбіт/с)	H.263 (індекс)	H.265 (індекс)	MPGE-2 (індекс)	MPEG-4 (індекс)
Nature	50	0.996262198	0.887194637	0.995907349	0.88746248
	40	0.996262198	0.887082133	0.995907349	0.88746248
	30	0.994504764	0.88695329	0.994566255	0.88668457
	20	0.992368566	0.886685138	0.988058216	0.88668457
	10	0.978225155	0.885777863	0.972482996	0.883969793
	8	0.978225155	0.885777863	0.962825184	0.879855629
	5	0.972186073	0.885318247	0.962825184	0.876586068
	3	0.944043032	0.879303242	0.933684579	0.854969247
	1	0.882768613	0.879303242	0.9006697	0.812210325
Coding	50	0.996262198	0.887194637	0.995907349	0.88746248
	40	0.996262198	0.887082133	0.995907349	0.88746248
	30	0.994504764	0.88695329	0.994566255	0.88668457
	20	0.992368566	0.886685138	0.988058216	0.88668457
	10	0.978225155	0.885777863	0.972482996	0.883969793
	8	0.978225155	0.885777863	0.962825184	0.879855629
	5	0.972186073	0.885318247	0.962825184	0.876586068
	3	0.944043032	0.879303242	0.933684579	0.854969247
	1	0.882768613	0.879303242	0.9006697	0.812210325
Race	50	0.998129646	0.931817858	0.99837069	0.93054834

Кінець таблиці 4.3

Кодова назва відео	Бітова швидкість (Мбіт/с)	H.263 (індекс)	H.265 (індекс)	MPGE-2 (індекс)	MPEG-4 (індекс)
Race	40	0.998129646	0.931815265	0.99837069	0.93054834
	30	0.998129646	0.931808908	0.99837069	0.93054834
	20	0.998129646	0.931787736	0.99837069	0.93054834
	10	0.998129646	0.93167798	0.99837069	0.93054834
	8	0.998129646	0.931620102	0.998334873	0.93054834
	5	0.995748992	0.931456314	0.995818147	0.929095343
	3	0.981469497	0.931148987	0.98826825	0.91858614
	1	0.973521846	0.929200978	0.985586395	0.910892119
Interview	50	0.997564961	0.985251997	0.99783552	0.983363829
	40	0.997564961	0.985240926	0.99783552	0.983363829
	30	0.997564961	0.985216959	0.99783552	0.983363829
	20	0.997564961	0.985151215	0.99783552	0.983363829
	10	0.997564961	0.984971331	0.99783552	0.983363829
	8	0.997564961	0.984971679	0.99783552	0.983363829
	5	0.997564961	0.984887079	0.99783552	0.983363829
	3	0.997564961	0.984169015	0.99783552	0.983363829
	1	0.997216308	0.984169015	0.986928658	0.983137779

Після розрахунку індексу структурної схожості, отриманні результати були згруповані відповідно до використаного відеокодеку, що дає змогу побачити зміни якості окремого типу відеозапису в залежності від обраного відеокодеку.

На рисунку 4.7 можна побачити як змінюється індекс структурної схожості відповідно до бітової швидкості, на динамічних відеозаписах (Nature, Race) можна помітити, що значна зміна якості відбувається на границі бітової швидкості 10-20

Мбіт/с. Також можна помітити, що формати МРGE-2 та Н.263 мають більш високі індекси структурної схожості.

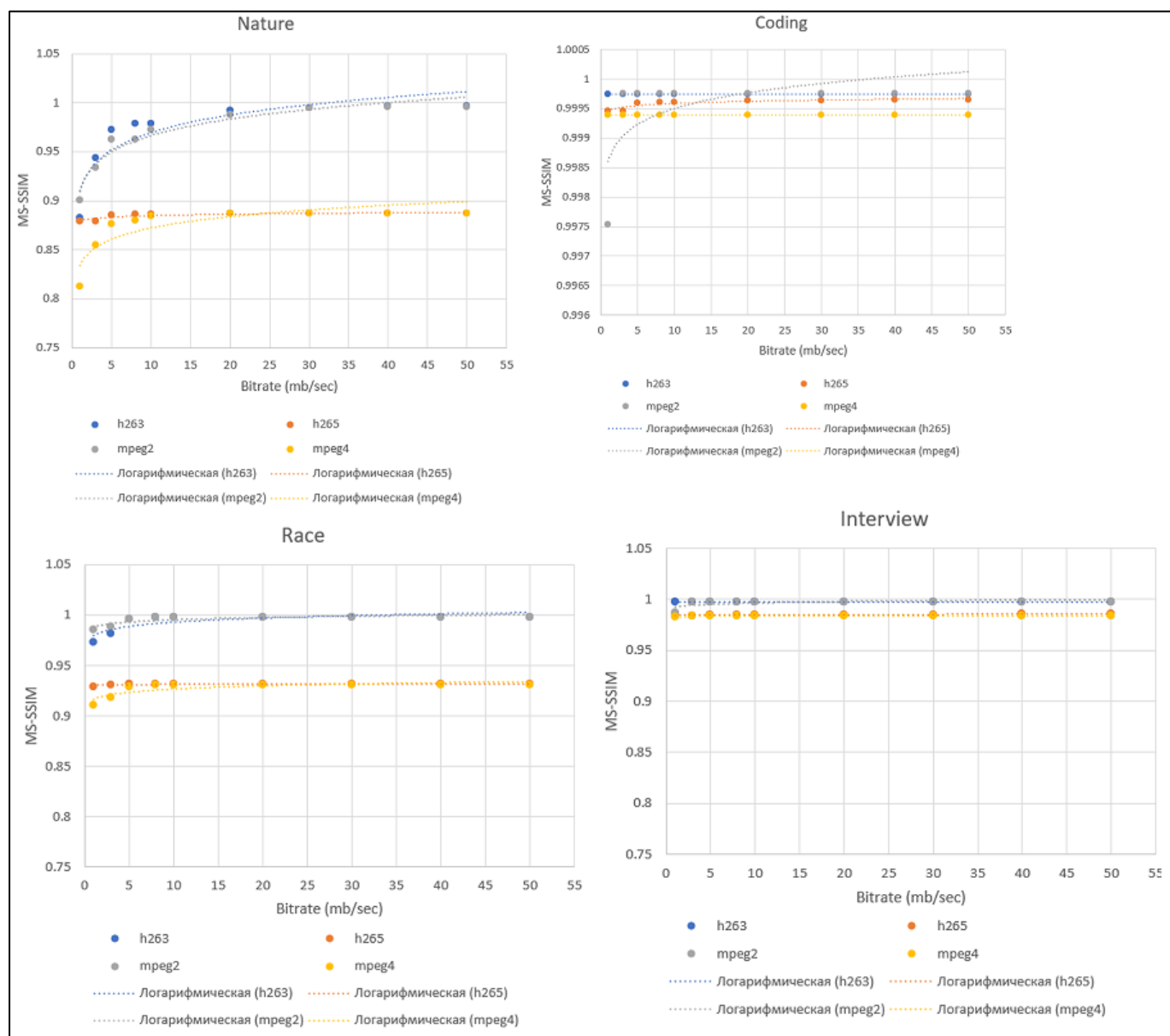


Рисунок 4.7 – Графіки залежності індексу структурної схожості від відеокодеку та бітової швидкості

Також можна помітити, що графіки мають малу розбіжність серед результатів, адже можливі результати індексу структурної схожості можуть бути від -1 до 1, де 1 – це однакові зображення.

Загальний алгоритм того, як відбувався експеримент можна побачити на рисунку 4.8.

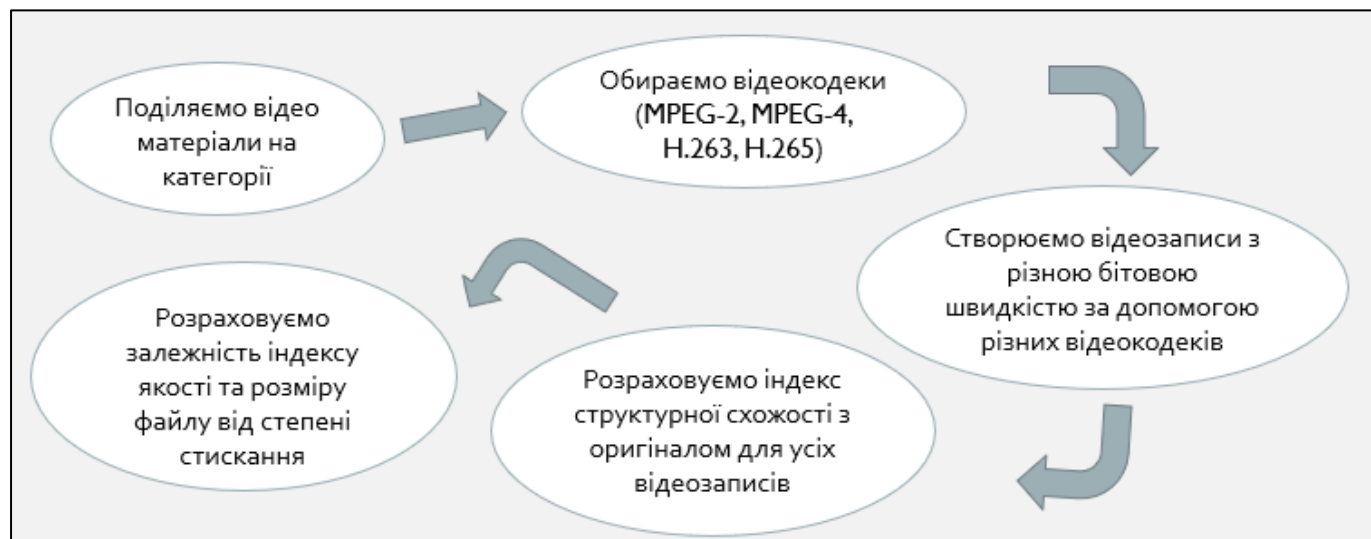


Рисунок 4.8 – Алгоритм відтворення експерименту

Останнім кроком необхідно виділити залежності та порівняти отриманні результати.

#### 4.4 Результати експерименту

На рисунку 4.9 зображено залежність індексу структурної схожості від бітової швидкості, результати згруповані за використаним відеокодеком. На діаграмах можна побачити, що усі обрані відеокодеки краще справилися з задачею кодування статичних відеозаписів (Coding, Interview).

Усі результати знаходяться в межах індексу (0.9 – 1), що дає змогу вважати, що обрані відеокодеки успішно впоралися с задачею. На основі діаграм можна сказати, що відеокодеки MPEG-2 та H.263 мають більш високі показники індексу, але

використовуючи діаграми з рисунку 4.5, можна зробити висновок, що ці відеокодеки мають більш лінійну тенденцію стиснення файлу, тобто, незалежно від бітової швидкості, MPEG-2 та H.263 сильно перетискають вихідний потік відео.

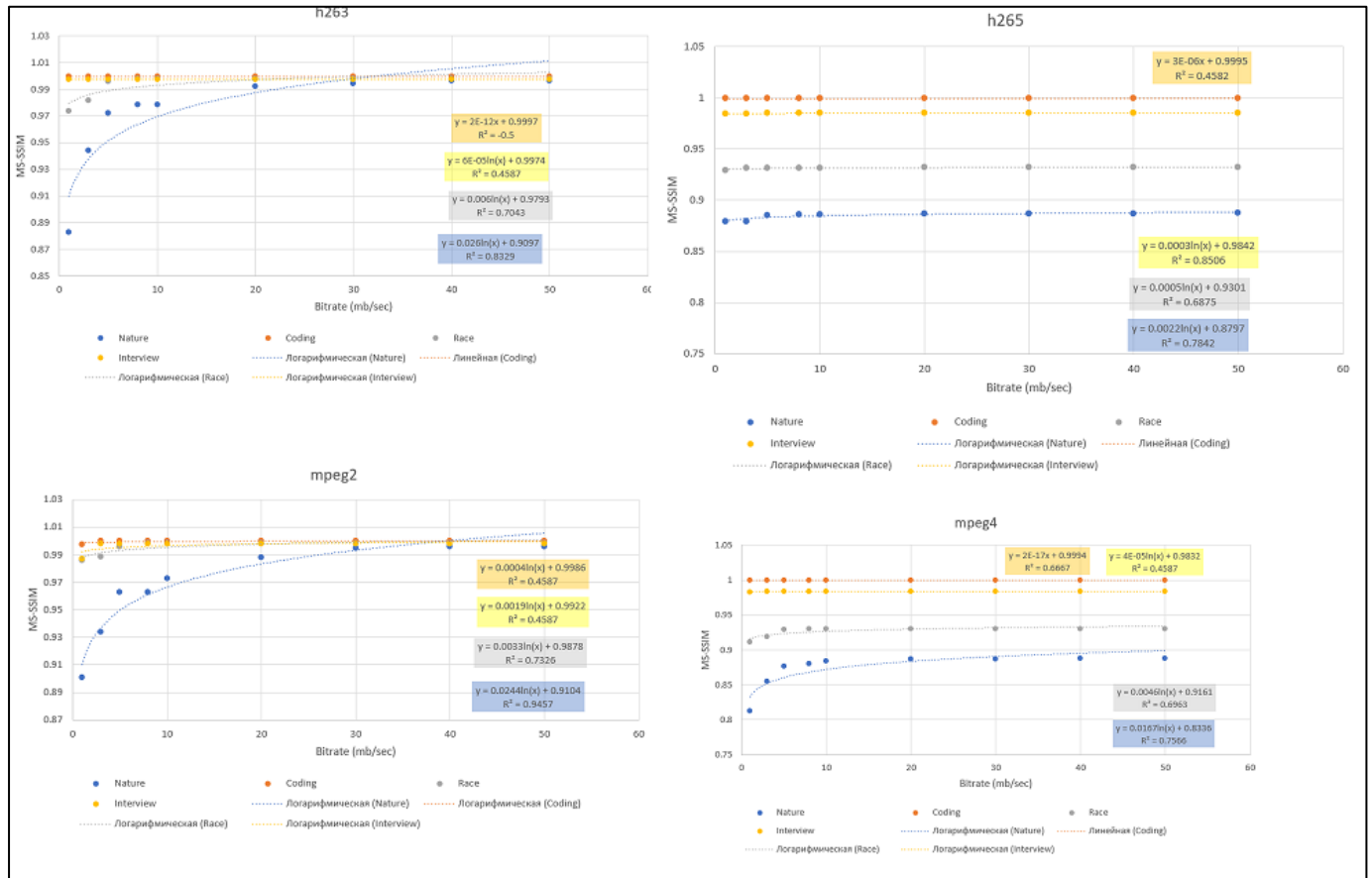


Рисунок 4.9 – Графіки залежності згрупованих результатів індексу структурної схожості від бітової швидкості

Відеокодек H.265 навпаки має дуже гарну тенденцію та логарифмічно стискає файл, та має більш лінійний показник зміни індексу якості, це вказує на те, що даний відеокодек можна використовувати при більш малої бітової швидкості, але власні алгоритми відеокодеку сильно перебільшують розмір файлів відносно оригіналу.

Відеокодек MPEG-4 має логарифмічну тенденцію зменшення розміру у порівнянні з більш старими відеокодеками MPEG-2 та H.263, та показує лінійну тенденцію зміни якості, що дає змогу вважати, що MPEG-4 можна використовувати

при більш низьких показниках бітової швидкості [30]. Та зробити висновок, що даний відеокодек краще підходить для поставленої задачі.

Як результат маємо, що для динамічних відеозаписів краще використовувати відеокодеки з логарифмічним трендом зміни розміру файлу, адже у цьому випадку можна правильно корегувати бітову швидкість. У свою чергу, для статичних відеоматеріалів краще обирати відеокодеки, які більше за все стискають файл без значної втрати якості.

## ВИСНОВКИ

Під час аналізу предметної галузі, було виявлено, що тема якості та стиснення відео у реальному часі є важливою та актуальною.

У ході роботи було проаналізовано наявні методи та алгоритми для стиснення відеоматеріалів. У ході аналізу предметної області було проведено аналіз існуючих рішень. Було виявлено, що частіше за все використовуються методи компенсації руху та кодування різниці кадрів. Були розглянуті сучасні відеокодеки, які використовують описані методи стиснення відеоматеріалів. Також були розглянуті більш застарілі версії відеокодеків. Також у більшості відеокодеків використовуються групи зображень, для кодування компенсації руху, які також є невід'ємною складовою сучасних відеокодеків.

Для заданої теми було проведено аналіз існуючих методів для реалізації клієнт-серверної взаємодії. Аналіз показав, що при сучасних технологіях, для можливості реалізації на різних платформах краще підходить технологія WebSockets. Обрана технологія є надійною, та економічною з боку збереження даних про користувача (клієнта).

Були проаналізовані наявні методи оцінки якості відеоматеріалів та обрано метод розрахунку індексу структурної схожості.

Також було розроблено програмний застосунок для реалізації розрахунку індексу структурної схожості з потоку відео. За допомогою цього застосунку були знайдені індекси для 144 заготовлених відеоматеріалів.

Була проведена експериментальна перевірка обраних відеокодеків. Таким чином були визначені залежності якості та розміру відео від бітової швидкості та обраних відеокодеків, які в свою чергу використовують досліджені методи стиснення відеоматеріалів.

У результаті роботи виявлено, що для ефективного стиснення відеоматеріалів необхідно застосовувати методами оцінювання якості, щоб регулювати процес стиснення. Встановлено, що будь-який відеокодек дає різну якість для кадрів різних типів. Якість відеозображення для опорних кадрів буде значно вищою ніж для інших кадрів. У випадку постійної бітової швидкості відеоматеріалу на повільних сценах якість буде високою, а на швидких – низькою. Тому виникає необхідність у введенні класифікації послідовностей відео за характерними ознаками для окремого регулювання бітового потоку в залежності від якості відеозображення. Краще за все виділити дві категорії: динамічні та статичні.

Як результат, бачимо, що для статичних відеоматеріалів краще обирати відеокодеки, які більше за все стискають файл без значної втрати якості. Для динамічних сцен необхідно використовувати відеокодеки з логарифмічним трендом зміни розміру файлу, адже у цьому випадку можна правильно корегувати бітову швидкість.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. An Overview of Video Compression Algorithms URL: <https://www.eetimes.com/an-overview-of-video-compression-algorithms/> (дата звернення: 21.03.2022)
2. Twitch URL: <https://www.twitch.tv/> (дата звернення: 21.03.2022)
3. Create and Multistream Live Video | Restream URL: <https://restream.io/> (дата звернення: 21.03.2022)
4. Live video streaming production software | Telestream Wirecast | Overview URL: <http://www.telestream.net/wirecast/overview.htm> (дата звернення: 21.03.2022)
5. Google Meet – Google Workspace URL: <https://google.oit.ncsu.edu/meet/> (дата звернення: 21.03.2022)
6. About Jitsi Meet | Free Video Conferencing Solutions URL: <https://jitsi.org/jitsi-meet/> (дата звернення: 21.03.2022)
7. Discord | Ваше місце для общення и отдыха URL: <https://discord.com/> (дата звернення: 21.03.2022)
8. About | Zoom URL: <https://explore.zoom.us/en/about/> (дата звернення: 21.03.2022)
9. Відеоконференції, наради й виклики | Microsoft Teams URL: <https://www.microsoft.com/uk-ua/microsoft-teams/group-chat-software> (дата звернення: 21.03.2022)
10. BigBlueButton | Open Source Virtual Classroom Software URL: <https://bigbluebutton.org/> (дата звернення: 21.03.2022)
11. Программная видеоконференцсистема для локальной сети | Выделенный сервер TrueConf Server URL: <https://trueconf.by/> (дата звернення: 21.03.2022)
12. В.В. Губинець. Сучасні тенденції архівування відеоматеріалів / П.В. Попович, В.В. Губинець // Матеріали XI Міжнародної науково-технічної конференції

молодих вчених «ЕЛЕКТРОНІКА-2018». – К., 2018, - С. 371-374 URL: <http://elconf.kpi.ua> (дата звернення: 21.03.2022)

13. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : Диалог-Мифи, 2003. – 381 с.

14. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М. : Техносфера, 2004. – 368 с.

15. K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrechko, and O. Ponomarenko, "Efficiency of Image Convolution", 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 6-8 Sept. 2019, Sozopol, Bulgaria, pp. 578–583.

16. Smelyakov, A. Datsenko, V. Skrypka and A. Akhundov, "The Efficiency of Images Reduction Algorithms with Small-Sized and Linear Details", 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), pp. 745-750, 2019.

17. Richardson I.E.G., «Video Codec Design» – John Wiley & Sons, 2002 – 304 с.

18. Описание форматов сжатия MPEG-2 и MPEG-4 URL: [https://www.sitcom.ru/tv/standart\\_mpeg\\_2\\_4.html](https://www.sitcom.ru/tv/standart_mpeg_2_4.html) (дата звернення: 21.03.2022)

19. Ричардсон Я. Видеокодирование.H.264 и MPEG-4 – стандарты нового поколения / Я. Ричардсон. – М. : Техносфера 2005. – 368 с.

20. H.264 чи H.265? Перспективи в області технологій стиснення відео URL: <https://hikvision.org.ua/ua/articles/h264-chy-h265-perspektyvy-v-oblastitehnologiy-stysnennya-video> (дата звернення: 21.03.2022)

21. Real time communication – Techtaget [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/searchunifiedcommunications/definition/real-time-communications> (дата звернення: 21.03.2022)

22. What is Real-Time Communications (RTC)? URL: <https://www.techtarget.com/searchunifiedcommunications/definition/real-time-communications> (дата звернення: 21.03.2022)
23. Лученко В., Афанасьєва І. Аналіз технологій для комунікації в режимі реального часу// Topical Issues of the Development of Modern Science, 2019, С. 797–803
- 24 R. Jesup, S. Loreto, and M. Tuexen, “RTC-Web Datagram Connection,” IETF Internet draft, Mar. 2012
25. Wang, Z. Modern Image Quality Assessment / Z. Wang, A. Bovik. Morgan and Claypool Publishing Company, 2006
26. Roufs J. A. J., Perceptual image quality: Concept and measurement // Philips Journal of Research.- 1992 - 47:35–62
27. Сравнение видеокодеков при помощи метрики PSNR URL: <https://www.ixbt.com/divideo/codex-psnr.shtml> (дата звернення: 21.03.2022)
28. Хор, А.; Цзыу Д. (август 2010 г.). Метрики качества изображения: PSNR против SSIM . 2010 20-я Международная конференция по распознаванию образов . С. 2366–2369
29. Абакумов В.Г. Способ улучшения качества видеоконтента по критерию качества результирующего изображения / П.В. Попович // Электротехнические и компьютерные системы. – 2015. – №17(93). – С.74-77
30. Абакумов В.Г. Применение объективных показателей качества видеоизображений для управления скоростью видеопотока / Попович П.В. // Автоматизация: проблемы, идеи, решения: междунар. науч.-техн. конф., 5-9 сентября 2011 г.: тезисы докл. – Севастополь: 2011. – С.198-200

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ  
КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

15. K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchychenko, and O. Ponomarenko, "Efficiency of Image Convolution", 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 6-8 Sept. 2019, Sozopol, Bulgaria, pp. 578–583.

16. K. Smelyakov, A. Datsenko, V. Skrypka and A. Akhundov, "The Efficiency of Images Reduction Algorithms with Small-Sized and Linear Details", 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), pp. 745-750, 2019.

23. Лученко В., Афанасьєва І. Аналіз технологій для комунікації в режимі реального часу// Topical Issues of the Development of Modern Science, 2019, С. 797–803