

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО»**  
**Факультет електроніки**

(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації

(повна назва кафедри)

«На правах рукопису»

УДК 681.3.06

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_

\_\_\_\_\_

(підпис) (ініціали, прізвище)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## Магістерська дисертація

зі спеціальності (спеціалізації) 171 Електроніка (Електронні та інформаційні технології кінематографії та аудіовізуальних систем)

(код і назва спеціальності)

на тему: «Система обслуговування відеоконтенту для сегменту дистанційного навчання»

Виконала студентка VI курсу, групи ДВ-72мп

(шифр групи)

\_\_\_\_\_

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник професор, д.т.н. Розорінов Г.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант \_\_\_\_\_

(назва розділу)

\_\_\_\_\_

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент \_\_\_\_\_

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2018 року

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Інститут (факультет) \_\_\_\_\_ Факультет електроніки \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Кафедра звукотехніки та реєстрації інформації \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) 171 Електроніка (Електронні та інформаційні технології кінематографії та аудіовізуальних систем)  
(код і назва)

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_ (підпис) \_\_\_\_\_ (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студентці**  
**Нестеренко Наталія Вадимівна**  
\_\_\_\_\_ (прізвище, ім'я, по батькові)

1. Тема дисертації: Система обслуговування відеоконтенту для сегменту дистанційного навчання.

науковий керівник дисертації: Розорінов Георгій Миколайович, д.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2018р. №4114-с

2. Строк подання студентом дисертації 10.12.2018р.

3. Об'єкт дослідження: дистанційне освітнє середовище, створення та забезпечення відеоконтенту для освітнього середовища

4. Предмет дослідження: методи та технології розроблення відеоконтенту та інтерактивного відеоконтенту для дистанційного освітнього середовища. Обробка та виведення відеосигналів, стиснення та транспортування без втрат, онлайн-платформи Moodle, ATutor- платформа з відкритим кодом, реалізація прямої трансляції, хмарні технології.

5. Перелік завдань, які потрібно розробити: провести огляд засобів організації дистанційного освітнього середовища та дослідити застосування відеоконтенту та інтерактивного відеоконтенту для реалізації платформи дистанційного навчання, розробити алгоритм стиснення відеоконтенту.
6. Перелік графічного (ілюстративного) матеріалу 52 рис., 21 табл., 1 презентація, 10 слайдів.
7. Орієнтовний перелік публікацій: «Аналіз ефективності різних стандартів стиснення відео» /Н.В. Нестеренко//; «Низькочастотний алгоритм відеокомпресії для збереження відеоматеріалів дистанційного навчання» / Н.В. Нестеренко// Науково-технічна конференція «Сучасні проблеми застосування електронних та інформаційних технологій в телекомунікаціях, телебаченні та цифровому кінематографі»

#### 8. Консультанти розділів дисертації\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 10.09.2017

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
	Написання першого розділу: «Основні характеристики відео».	10.10.2017	
	Написання другого розділу: «Огляд сучасних платформ для дистанційного навчання. Тенденції використання інтерактивного відеоконтенту».	15.12.2017	
	Написання третього розділу: «Програмне забезпечення створення та реалізації віртуальних освітніх технологій».	01.05.2018	
	Написання четвертого розділу: «Апаратні та програмні засоби створення відеоконтенту для сегменту дистанційного навчання».	10.10.2018	
	Підготовка матеріалів до друку та оформлення пояснювальної записки	25.11.2018	
	Підготовка та оформлення презентації для доповіді	30.11.2018	

УДК 681.3.06

## РЕФЕРАТ

Магістерська дисертація: 113 с., 15 рис., 3 табл., 4 дод., 32 джерело  
ХМАРНІ ТЕХНОЛОГІЇ, ІНТЕРАКТИВНИЙ ВІДЕОКОНТЕНТ,  
ВІДЕОТРАНСЛЯЦІЯ, КОДЕК, ВІДЕОКОДЕР, E-LEARNING, SCORM

Метою роботи є забезпечити методологічні та технологічні аспекти створення відеоконтенту для сегменту дистанційного навчання. Об'єкт дослідження: процес кодування, передавання та декодування відеосигналу у дистанційному навчанні.

У дисертації виконано аналіз стандартів кодування, стандарти оцінки якості, стек протоколів передавання відеоданих, проаналізовано існуючі платформи для дистанційного навчання та виконано їх порівняння.

Розглянуті методологічні аспекти удосконалення від звичайного відеоконтенту до інтерактивного. Запропоновано технології, що забезпечують створення високоякісного контенту з урахуванням оптимальних ресурсів апаратної частини обробки.

Розглянута типова архітектура платформ дистанційного навчання та наведені приклади реалізації. Галузь застосування: дистанційне навчання, онлайн платформи, доповнення очного навчання, відеотрансляції через мережу Інтернет.,

## **SUMMARY**

The aim of the work is to provide methodological and technological aspects of video content creation for the E-learning segment. The object of research: the process of encoding, transmitting and decoding video signals in distance learning. The thesis analyzes the coding standards, quality assessment standards, video data transmission protocols, analyzes the existing platforms for distance learning and compares them. The methodological aspects of upgrading from the usual video to the interactive are considered. We propose technologies that provide the creation of high-quality content, taking into account the optimal resources of the part of the processing. The typical architecture of the distance learning platforms is considered and examples of implementation are presented. Field of application: distance education, online platform, full-time training, video broadcasting over the Internet.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	8
ВСТУП .....	9
1 ОСНОВНІ ХАРАКТЕРИСТИКИ ВІДЕО .....	12
1.1 Якість відеозображення .....	12
1.2 Кадрова частота(fps) та кадрова розгортка .....	14
1.3 Роздільна здатність .....	16
1.4 Основні формати стиснення .....	17
1.5 Кодеки .....	21
2 ОГЛЯД СУЧАСНИХ ПЛАТФОРМ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ. ТЕНДЕНЦІЇ ВИКОРИСТАННЯ ІНТЕРАКТИВНОГО ВІДЕОКОНТЕНТУ .....	26
2.1 Інтерактивний відеоконтент у дистанційному навчанні .....	27
2.2 Застосування технологій віртуальної реальності для навчання і досліджень .....	31
2.3 Прямі трансляції для дистанційноо навчання .....	36
2.4 Основний принцип побудови типової архітектури систем дистанційного навчання .....	38
2.5 ATutor – модульна система дистанційним керуванням навчанням з відкритим кодом .....	42
2.6 Moodle .....	44
3 АПАРАТНІ ТА ПРОГРАМНІ ЗАСОБИ СТВОРЕННЯ ВІДЕКОНТЕНТУ ДЛЯ СЕГМЕНТУ ДИСТАНЦІЙНОГО НАВЧАННЯ .....	51

	7
3.1 Хмарні технології в дистанційному навчанні.....	51
3.2 Реалізація прямої трансляції.....	55
3.3 Аналіз ефективності різних стандартів стиснення відео.....	63
3.4 Низькочастотний алгоритм відеокompresії для збереження відеоматеріалів дистанційного навчання.....	67
4 СТАРТАП-ПРОЕКТ.....	70
4.1 Загальні відомості.....	70
4.2 Технологічний аудит ідеї проекту.....	74
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	75
4.4 Розроблення ринкової стратегії проекту.....	79
4.5 Розроблення маркетингової програми стартап-проекту.....	81
ВИСНОВКИ.....	83
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	86
ДОДАТОК А. ABSTRACT.....	89
ДОДАТОК Б.....	95
ДОДАТОК В.....	99
ДОДАТОК Г.....	112

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

E-Learning	–	Дистанційне навчання
CODEC	–	Coder/Decoder (кодер/декодер);
МБ	–	Мегабайт;
Мб/с	–	Мегабіт за секунду;
PSNR	–	Пікове відношення сигналу до шуму, вимірюється в дБ
PAL	–	Построкове змінення кольорової фази у системі аналоговогоцифрового телебачення
HDTV	–	Формат відео з високою роздільною здатністю
CBR	–	Постійний бітрейт
VBR	–	Змінний бітрейт
VR	–	Віртуальна реальність
SCORM	–	Стандарти для системи дистанційного навчання
LCMS	–	Система управління навчанням
API	–	Програмний інтерфейс



## ВСТУП

**Актуальність дослідження.** Система дистанційного навчання має багато переваг у сучасному світі, забезпечуючи безперервний потік інформації для саморозвитку, загального обміну інформацією. Зростаючі інформаційні потоки привели до швидкого застаріння знань, сучасному фахівцю необхідно постійно підвищувати свою кваліфікацію. При цьому підвищення кваліфікації і перепідготовки кадрів в більшості випадків повинно проводитися без відриву від діяльності, що стає можливим з використанням технологій відкритої освіти. Дистанційне навчання- це спосіб навчання на відстані, при якому викладач і ті, які навчаються фізично знаходяться в різних місцях, якщо враховувати не тільки транспортні витрати, а й витрати на організацію всієї системи очного навчання, то стає зрозумілим, дистанційне навчання є ефективним практично для будь-якої людини, що бажає підвищити свій освітній рівень, не залежно від віку, поточної зайнятості, інтересів і можливостей.

Розуміння що більшість теоретичних баз з інформацією майже не дають потрібної ефективності у засвоєнні навчального процесу приводить до пошуку нових засобів донесення інформації. За останні роки суттєво вагомою частиною пошукових запитів користується відеоконтент

- 67% випадків використання інтернету пов'язано з відео
- 70% відвідувачів інтернету цілеспрямовано заходять до мережі аби подивитися відео
- За 24 години відео переглядається більш ніж 800 мільйонів людей

І ця тенденція тільки зростає. Переваги використання відеоконтенту для дистанційного навчання очевидні, більшість людей сприймає інформацію візуально, адже у цифровий вік просто слів та тексту вже недостатньо.

**Метою дослідження** забезпечити методологічні та технологічні аспекти створення відеоконтенту для сегменту дистанційного навчання

**Об'єкт дослідження** – процес підготовки відеоконтенту для сегменту дистанційного навчання

**Предмет дослідження** – відео формати та характеристики, процеси адаптації, кодування та декодування, збереження та передавання відеоінформації, платформи дистанційного навчання.

**Методи дослідження** – теоретичне дослідження процесу формування відеоконтенту для існуючих платформ дистанційного навчання, аналіз складових дистанційного навчання та розробка відповідного алгоритму підготовки відеоінформації

**Наукова новизна отриманих результатів:** запропоновано технології, що забезпечують доцільну підготовку для формування відеоконтенту для дистанційного навчання, проаналізовано сучасні потреби для стандартів якості відео, способів зберігання, транслявання та обробки відеосигналів.

**Практична цінність отриманих результатів** – Отримані данні, дозволяють реалізувати проект по організації власної платформи інтерактивного дистанційного навчання та забезпечити якісний відеоконтент на базі цієї платформи.

**Апробація результатів роботи.** Результати роботи були апробовані на таких конференціях:

1. І всеукраїнська науково-технічна конференція «Сучасні технології кіно та аудіовізуальних систем»[Електронний ресурс] / Нестеренко Н.В.: НТУУ «КПІ ім. Ігоря Сікорського», 2018. – С. 23-26 – Режим доступу: [http://elconf.kpi.ua/wp-content/uploads/2014/01/ELCONF2018\\_sbornik\\_full\\_2.pdf2](http://elconf.kpi.ua/wp-content/uploads/2014/01/ELCONF2018_sbornik_full_2.pdf2). Н.В. Нестеренко низькочастотний алгоритм відеокомпресії для збереження відеоматеріалів дистанційного навчання. 2. Аналіз ефективності різних стандартів стиснення відео

# 1 ОСНОВНІ ХАРАКТЕРИСТИКИ ВІДЕО

**Відео**—широкий спектр технологій запису, обробки, передачі, зберігання й відтворення візуального і аудіовізуального матеріалу[1].

Відеодані представляють собою потік аналогових сигналів. Комп'ютерна обробка відео інформації полягає в перетворенні їх у цифровий формат з подальшим зберіганням цих даних на жорсткому або іншому пристрої зберігання інформації. Оцифровка відеосигналу включає дві стадії: дискретизація даних аналогового відеопотоку, тобто зняття відліків з певною частотою, і перетворення кожного такого відліку в цифровий еквівалент або квантування. При роботі з такими даними неможливо обійтися без стиснення, в цьому випадку потрібен час, в залежності від обчислювальною потужності задієної для розпакування даних.

## 1.1 Якість зображення

Якість зображення – дозвіл, тобто кількість відтворених вертикальних ліній. Сприйняття якості зображення залежить від просторової чіткості (наскільки чітко помітні частини сцени, чи є очевидним спотворення, та часові затримки - зображення має бути плавним та природнім). Однак думка глядача про якість зображення також залежить від інших факторів, таких як середовище переглядання, психічний стан людини та ступінь взаємодії глядача з візуальною сценою. Ця оцінка, по суті, поверхнева, так як існує багато інших, не менш важливих, параметрів, настільки ж помітних людському оку. Якість відео вимірюється також за допомогою об'єктивних засобів оцінки таких, як PSNR[2] (peak signal to noise ratio / пікове відношення сигналу до шуму, вимірюється в дБ). Принцип дії аналогічний середньоквадратичному відхиленню, проте і недоліки вона має тіж самі.

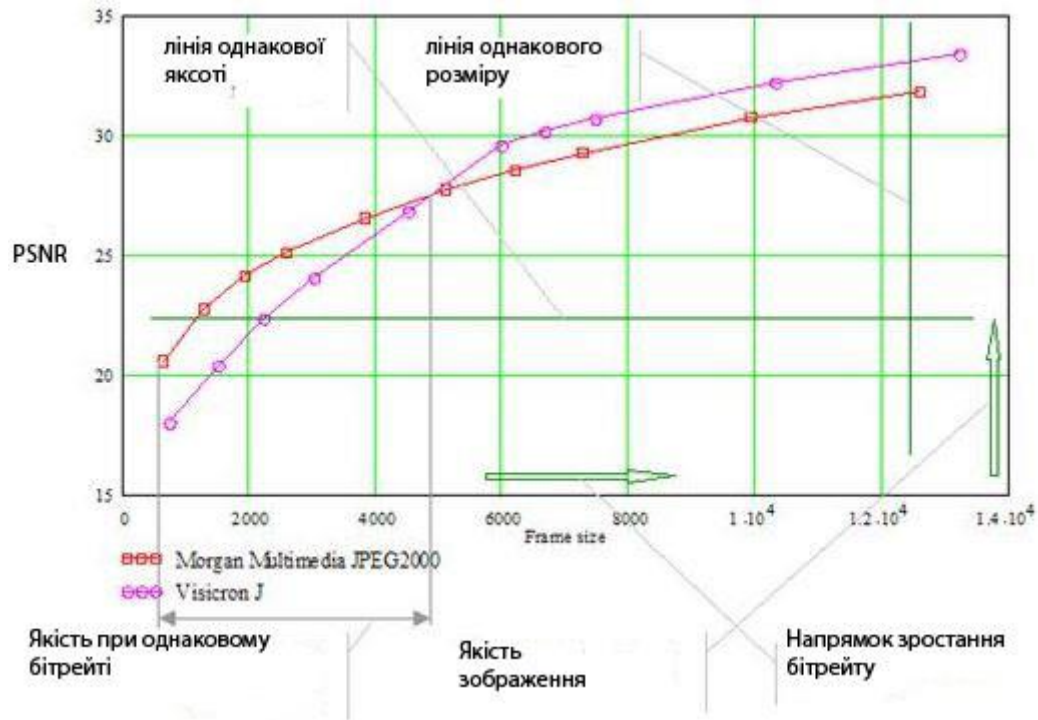


Рисунок 1.1- Залежність показника PSNR від середнього розміру кадру

На рис.1.1 зображена залежність показника метрики від середнього розміру кадру. Кожна гілка відповідає певному кодеку. Гілки побудовані на опорних точках, кожна з яких відповідає конкретному бітрейту. На кожній гілці знаходиться за десять точок (кожна послідовність стискається на 10 налаштованих бітрейтів). При порівнянні кодеків не слід звертати увагу на те, як високо розташовані гілки кодеків. Чим вище знаходиться гілка - тим вище якість послідовності, стиснутої даними кодеком.

Окрім PSNR існують і інші метрики для об'єктивної оцінки якості відео такі як:

MSAD- це усереднена абсолютна різниця значень кольорних компонентів у відповідних точках зображень, що порівнюються. Використовується для порівняння кодеків або фільтрів.

Blurring measure – дозволяє порівняти ступінь розмиття між двома зображеннями.

SSIM Index- порівняння відбувається за 3 компонентами: яскравість, контраст, структура.

Помітний вплив на суб'єктивну оцінку якості мають візуальна увага (глядач сприймає сцену, більше концентруючись на послідовності ділянок зображення, а не сприймаючи зображення повністю) та так званий «ремісійний ефект» (наша оцінка візуальної сцени залежить від недавно переглянутого матеріалу, а не від матеріалу, що був переглянутий раніше). Усі ці фактори заважають виміряти візуальну якість точно та кількісно[3]. Як виглядає система суб'єктивної оцінки якості можна спостерігати на рис.1.2

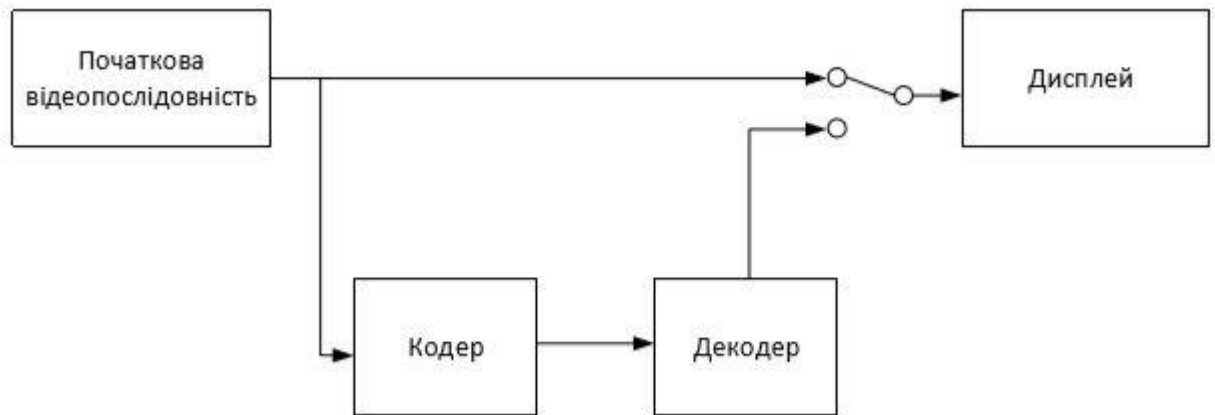


Рисунок 1.2- Суб'єктивна система оцінки якості зображення

## 1.2 Кадрова частота(fps) та кадрова розгортка

Кадр-одиниця зображення відео.

Частота кадрів у секунду - це число нерухомих зображень, що змінюють одне одного впродовж 1 секунди відеопотоку, створюючи ефект руху об'єктів на екрані. Чим більша частота кадрів на секунду, тим плавнішим і природнішим буде здаватися рух. Мінімальний показник, при якому рух буде сприйматися однорідним - приблизно 10 кадрів на секунду (це значення індивідуально для

кожної людини). У традиційному плівковому кінематографі використовується частота 24 кадру в секунду[4]. Системи телебачення PAL використовує 25fps, а система NTSC використовує 29,97 fps. Комп'ютерні оцифровані відеоматеріали гарної якості, як правило, використовують частоту 30 кадрів в секунду. Сучасні професійні камери можуть знімати з частотою до 120 кадрів в секунду. А спеціальні камери для надшвидкої зйомки з частотою до 1000 кадрів у секунду. Стандарти PAL та NTSC мають черезстрокову розгортку.

Розгортка зображення – процес послідовного перетворення за установленим законом яскравості і кольоровості. Іншими словами, це спосіб здобування растру. Для скорочення переданого потоку вдвічі застосовується черезстрокова розгортка, при якій кожен кадр передається двома послідовними напівкадрами - полями. Поле складається з телевізійних рядків. Одне поле містить парні рядки, друге - непарні. Такий режим розгортки позначається значком «і» від англ. interlace. Він був розроблений в епоху аналогового телебачення, коли не було можливості передавати сигнали з широкою смугою пропускання. Перші цифрові формати і HD використовували цей тип для зменшення ширини відеопотоку. Недоліком такого режиму є наявність ефекту «гребінки» на рухомих об'єктах при відтворенні на пристроях відображення з прогресивною (порядковою) розгорткою. При порядковій передачі на інших ділянках зображення таких проблем не виникає, однак ширина смуги пропускання або потік такого відеосигналу буде вдвічі більшими. При прогресивній розгортці (p) частоти дискретизації для схеми 4: 2: 2 будуть рівними для

Y - 27 МГц, для Cr / Cb - 13,5 МГц.

Через різні стандарти, які прийняті у різних країнах, в телевізійному мовленні, кіно і відео виробництві з'явилась велика кількість різних стандартів, які можуть частково або повністю підтримувати всі засоби зчитування відеоінформації. Основними є:

- на основі форматів PAL: 25i, 25p, 50p

- на основі форматів NTSC: 29.97i, 30i, 29.97p, 30p, 59.94p, 60p
- кіноформат: 23.98p, 24p

Ще одним важливим параметром є співвідношення сторін кадру відеозображення. Типовими форматами для відео є стандарт 4: 3 (1,33: 1) або широкоекранний - 16: 9 (1,77: 1). Широкоекранний режим іноді записується на відео зі стисненням по горизонталі до 4: 3, а при відтворенні розтягується. Така технологія називається анаморфування відеозапису і при записі широкоформатних фільмів дає можливість повніше використовувати кадр телебачення стандартної чіткості.

### 1.3 Роздільна здатність

Цифровий відеосигнали характеризуються за тим самим принципом, що і розрішення комп'ютерних моніторів – роздільною здатністю горизонтальних і вертикальних вимірювань в пікселях. При оцифруванні аналогового відео стандартної чіткості дозвіл складає  $720 \times 576$  пікселів для європейського стандарту розкладення 625/50 (PAL і SECAM), при частоті кадрів 50 Герц і  $720 \times 480$  пікселів для американського стандарту розкладення 525/60 (NTSC), при частоті 59,94 Герц (два поля,  $2 \times 29,97$ ). У виразі  $720 \times 480$  першим числом позначається кількість точок у рядку (горизонтальна роздільна здатність), а другим числом кількість активних рядків, що беруть участь в побудові зображення (вертикальна роздільна здатність). Стандарт цифрового відео HDTV з високою роздільною здатністю передбачає розрішення до  $1920 \times 1080$  при частоті оновлення 50 Герц (60 Гц для США) з прогресивною розгорткою. Тобто 1920 пікселів на рядок, 1080 рядків[5]. А для UHD TV  $3840 \times 2160$  точок при кадровій частоті 50 і 60 Гц

Ширина (інакше кажуть швидкість) відеопотоку або бітрейт- це кількість оброблюваних біт відеоінформації за секунду часу (вимірюється «біт / с» - біт в

секунду або, частіше, «Мбіт / с» - мегабіт в секунду. Чим вище ширина відеопотоку, тим, як правило, краще якість відео. Наприклад, для формату VideoCD ширина відеопотоку складає всього приблизно 1 Мбіт / с, а для DVD складає близько 5 Мбіт / с. Звичайно, суб'єктивно різницю в якості не можна оцінити як п'ятикратну, але об'єктивно це так. А формат цифрового телебачення HDTV використовує ширину відеопотоку близько 10 Мбіт / с, а для UHD TV, або 4K - 15 Мбіт/с. За допомогою швидкості відеопотоку оцінюється якість відео при його передачі через Інтернет.

Розрізняють два види управління шириною потоку в відеокодеку - постійний бітрейт (англ. Constant bit rate, CBR) і змінний бітрейт (англ. Variable bit rate, VBR). Концепція VBR використовується частіше усього і максимально зберігає якість відео, зменшуючи при цьому сумарний обсяг переданого відеопотоку. При цьому на швидких сценах руху, ширина відеопотоку зростає, а на повільних, де кадри змінюються повільно, ширина потоку падає. Це зручно для буферизованих відеотрансляцій і передачі збереженого відеоматеріалу через комп'ютерні мережі. Але для безбуферних систем реального часу і для прямого ефіру це не підходить - у цих випадках необхідно використовувати постійну швидкість відеопотоку.

#### **1.4 Основні формати стиснення**

Стиснення без втрат (фактично без зниження якості зображення) в цифровому відеозаписі виконується методами, аналогічними тим, які використовуються в архіваторах WinZIP або WinRAR. Однак через деякі особливості відеоінформації такі алгоритми в даному випадку недостатньо ефективні (мінімальний розмір стисненого файлу складає 1/3 від вихідного) і тому практично не застосовується.

Стиснення з втратою якості є основним методом зменшення розміру відеофайлів. Такі алгоритми дозволяють визначити ту частину інформації, яку



глядач, найімовірніше, не помітить при перегляді фільму, і видалити її з файлу. Основними форматами цифрового відео, що використовують стиснення з втратами, на сьогоднішній день є Apple QuickTime, Intel Indeo, MPEG-1, MPEG-2, MJPEG і MPEG-4. У розробці також знаходяться специфікації стандарту MPEG-7, але про нього поки говорити ще рано[6].

### Intel Indeo

Даний формат був розроблений корпорацією Intel для стиснення відеоданих з використанням нових можливостей процесорів Intel Pentium MMX. Крім підтримки потокової передачі даних і функцій захисту авторських прав, цей стандарт реалізує кілька новаторських на момент його появи функцій. Він дозволяє застосовувати до відеопослідовностей різні ефекти (наприклад, змінювати яскравість або контрастність) в реальному часі, декодувати не весь кадр, а, наприклад, центральний фрагмент, робити частину кадру одного відеоролика прозорою і накладати два відеозаписи один на одного. Останній ефект часто використовують в програмах телевізійних новин, коли коментатор зображується на тлі відеорепортажу з місця подій.

Однак формат Indeo не набув великого поширення. А з виходом MPEG-4, в якому також присутні всі ці можливості, даний стандарт взагалі не використовується.

### MPEG-1

Формат стиснення відеоданих MPEG-1 був розроблений Motion Picture Expert Group - міжнародна організація, що створює стандарти стиснення відеоінформації. Він підтримує максимальний дозвіл кадру 4095 x 4095 пікселів при частоті їх зміни до 60 fps. Однак зазвичай використовують дозвіл 352 x 288, відповідне якості запису на звичайну касету VHS.

Використання для кодування алгоритмів MPEG-1 дозволяє отримувати відеопотік шириною до 1,5 Мбіт в секунду з роздільною здатністю кадру 352x288 пікселів для PAL або 352x240 для NTSC, після чого на звичайному CD може

вміститися 74 хвилини відео зі звуком якості VHS (як у звичайного відеомагнітофона). На рис.1.3 зображено як всі кадри відеоролика підрозділяються на три типи: I-, P- та B-кадри. До першого типу (I-кадри, Intra Frames) відносяться початкові кадри. Їх зображення зберігаються у повному обсязі у форматі JPEG. Для P-кадрів (передбачені кадри) записуються лише відмінності від попереднього I-кадру, що вимагає набагато меншого дискового простору. Для B-кадрів (Bi-DirectiOnally Interpolated Frames) збережені відмінності від попереднього та наступного I- або P-кадра .

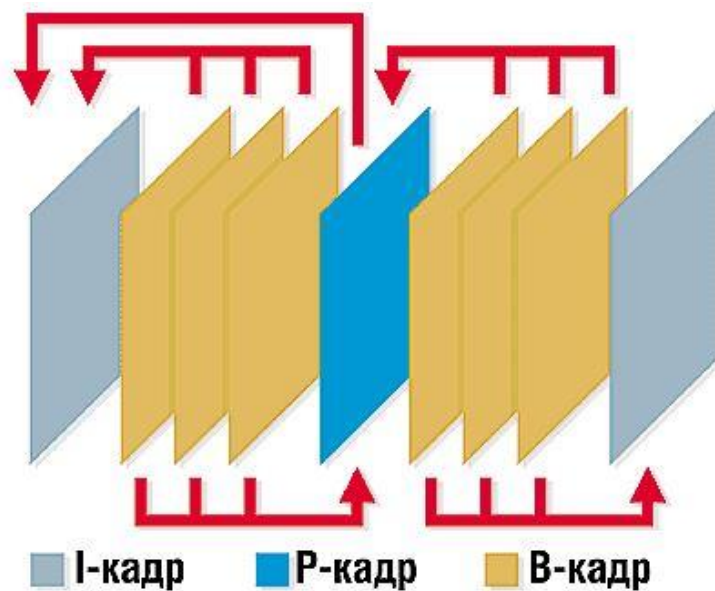


Рисунок 1.3- Графічне зображення кодеку MPEG1

## MPEG-2

MPEG-2 являє собою подальше розширення MPEG-1. У ньому збільшено рекомендований розмір кадру - тепер він становить 1920 x 1080 точок, додана підтримка шестиканального звуку. Однак для відтворення відео в цьому форматі потрібна вища обчислювальна потужність комп'ютера.

Слід зазначити, що велася робота над створенням стандарту MPEG-3 Він повинен був стати базовим для систем цифрового телебачення високої чіткості

HDTV. Але робота над ним була перервана, оскільки потрібні для HDTV вимоги вдалося реалізувати у вигляді невеликих розширень до MPEG-2.

Цей формат зараз досить широко поширений на Заході: його використовують для передачі відео по супутникових каналах і кабельних мережах цифрового телебачення, крім того, всі відеодиски DVD записані саме в цьому форматі.

### MJPEG

Фактично MJPEG (Motion JPEG) - це перехідний формат від стиснення звичайних фотографій до стиснення відео. Кожен кадр записується в форматі JPEG, а потім поміщається в відеоряд. MJPEG використовується переважно в платах відеомонтажу, наприклад Fast AV Master, MiroVideo DC50 і т. Д. Він дозволяє зменшити потік відеоданих з 30 Мб/с до 6 Мб/с. Для застосування в побутових відеопрогравачах цей стандарт мало придатний через низький коефіцієнта стиснення (5: 1) і відсутність можливостей для синхронізації відеоряду та звуку.

### MPEG-4

Формати MPEG-1 і MPEG-2 не забезпечували реальної можливості трансляції відео у мережі Internet, а при створення інтерактивного телебачення на їх основі – файли мали великий розмір[7]. Для його радикального зменшення, а також реалізації інших функцій, необхідних для передачі потокового відео, було розпочато роботу над специфікаціями нового формату - MPEG-4. Він орієнтований не стільки на стиснення відео, скільки на створення так званого "мультимедійного контенту" - злиття інтерактивного телебачення до мережі інтернет, 3D-графіки, тексту і т.і.

Що стосується самого відео, то найважливішим нововведенням стало подальше вдосконалення технологій розкладання сцени на об'єкти і алгоритмів їх ефективного стиснення. Для типових об'єктів розроблені окремі алгоритми передбачення і опису їх рухів - це стосується, зокрема, ходи людей, найбільш

поширених жестів, міміки. Тепер такі зміни в кадрах немає потреби записувати взагалі - їх можна розрахувати програмно.

В MPEG-4 підтримується відображення тексту різними шрифтами поверх відеозображення. Більш того, цей текст може бути озвучений за допомогою синтезатора мови з можливістю імітації чоловічих і жіночих голосів. При необхідності голос синхронізується з рухами особи диктора відповідно до вимовними фонемами. Також може синтезуватися звучання деяких музичних інструментів. Стиснення оцифрованих звукозаписів здійснюється більш ефективно за допомогою спеціально розробленого кодека AAC (Advanced Audio Codec).

## 1.5 Кодеки

Як було з'ясовано в попередньому розділі, для стиснення відео можуть використовуватися різні стандарти. Але при цьому, вибравши певний алгоритм перетворення даних, можна стиснути відео абсолютно різними інструментами або програмними засобами, що дає на виході абсолютно різні результати. Оцінити розбіжність можна на прикладі якості і характеристик відео записаного на мобільний телефон, завантаженого з мережі відеоролика в HD-форматі або фільму з Blu-Ray диска. Вірогідно, що у одного картинка неякісна, в іншого звук, а третій, навпаки - еталон якості, хоча всі вони закодовані з використанням одного стандарту - MPEG-4.

Багато в чому, всі ці відмінності як раз і визначаються кодеком - спеціальною програмою, що здійснює стиснення (кодування) вихідного матеріалів. При цьому кожен з них використовує свій власний алгоритм, який впливає як на якість, так і на швидкість кодування.

Відеокодування представляє собою процес стиснення чи ущільнення цифрової відеопослідовності у меншу кількість бітів. Необроблене чи нестиснене

зазвичай потребує великий бітрейт (приблизно 216 Мбіт для 1 секунди нестиснутого відео телевізійної якості) і стиснення необхідне для практичного зберігання і передавання цифрового відео.

Компресія містить додаткову пару систем – компресор (кодер) та декомпресор (декодер). Кодер перетворює вихідні дані у стиснену форму (займаючи меншу кількість бітів) перед передаванням чи зберіганням, а декодер перетворює стиснені дані у вигляд початкових відеоданих. Пара кодера та декодера часто описується як кодек (enCO/DECoder)[8]



Рисунок 1.4- Система роботи кодеку

При стисненні даних досягається усунення надмірності, тобто відпадає необхідність в надлишковій кількості даних. Численні типи даних забезпечують повну статистичну надмірність та можуть бути ефективно стиснені з використанням методу стиснення без втрат так, що відновлені дані можуть бути більш точними ніж початкові дані. На жаль, стиснення без втрат інформації зображення і відео дає лише помірне стиснення. Найкращий результат, що може бути досягнутий стисненням без втрат дає стандарт JPEG-LS – ступінь стиснення до 3-4 разів.

**H.261** Рекомендація ITU-T H.261 - Video codec for audiovisual services at p x 64 kbit / s. Дана рекомендація описує метод кодування / декодування відеозображення для використання в системах відеоконференцій при швидкостях передачі даних p x 64 Кбіт / с, де p може приймати значення від 1 до 30 [9].

**H.263** - це відеокодек, ІТУ-Т, призначений для передачі відео по каналах з досить низькою пропускнуою здатністю (зазвичай нижче 128 кбіт / с). Застосовується в програмному забезпеченні для відеоконференцій. Стандарт H.263 був спочатку розроблений для використання в системах, що базуються на H.324 (PSTN і інші системи відеоконференцій і голосового зв'язку), але згодом знайшов застосування в H.323 (відеоконференції, засновані на RTP / IP).

**H.320** ( відеоконференції, засновані на ISDN), RTSP (потокове аудіо та відео) і SIP (Інтернет-конференції).

**H.263** це розвиток стандарту H.261, попередньої розробки ІТУ-Т - стандарту відеостиснення, і алгоритмів MPEG-1 і MPEG-2. Перша версія була завершена в 1995 році і була гарною заміною для застарілого H.261 на каналах з будь-якою пропускнуою здатністю.

Новий розширений кодек від ІТУ-Т (у партнерстві MPEG) після H.263 - це стандарт H.264, також відомий як AVC та MPEG-4 [10].

Оскільки H.264 має суттєво розширені можливості у порівнянні з H.263, він став основним при розробці програмного забезпечення для відеоконференцій. Більшість сучасного програмного забезпечення цього напрямку підтримує H.264, також як і H.263 або H.261.

Гарною ознакою відповідності стандарту являється можливість кодера взаємодіяти з еталонним програмним забезпеченням H.264 - є загальнодоступним для завантаження розробниками. Якщо сторонній кодер чи декодер може успішно взаємодіяти із стандартним програмним, це вказує на те, що кодек (CODEC) з великою ймовірністю відповідає стандарту. Однак, це не стовідсотково підтверджує відповідність. Історія процесу стандартизації відеокодерів показана на рис. 1.4:

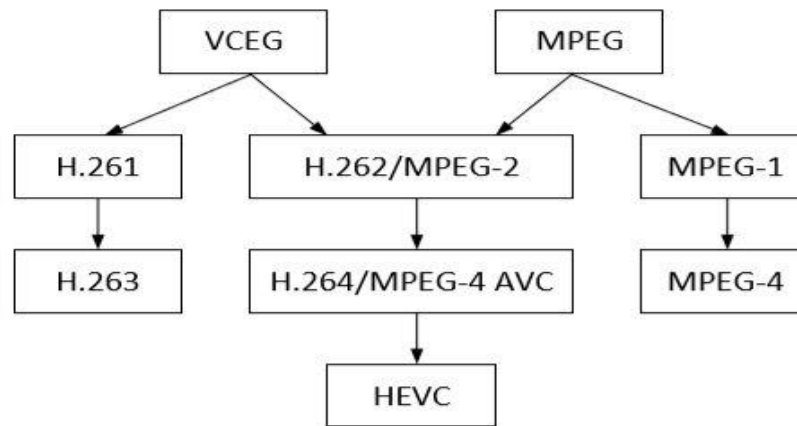


Рисунок 1.5 – Процес стандартизації відеокодеків

### Висновки до розділу

У даному розділі були розглянуті основні характеристики та формати відеоінформації, а також технічні аспекти які впливають на створення відеоконтенту.

Отже для сегменту дистанційного навчання доцільним є опиратися на суб'єктивну оцінку якості відео, атже цільова аудиторія планує переглядати відео у наукових та пізнавальних цілях, а не зберігати та використовувати у подальшій постобробці.

Для забезпечення оптимального співвідношення якість/обчислювальна потужності та для захвату більшої цільової аудиторія доцільним буде створювати початковий відеоматеріал у стандартах PAL: 25р, 50р

NTSC: 29.97р, 30р.

За роздільної здатністю, на сьогоднішній день, сфера зорієнтована на використання таких стандартів цифрового відео як HDTV з розрешенням до 1920 × 1080 при частоті оновлення 50 або 60 Герц з прогресивною розгорткою та UHD TV або 4K 3840×2160 точок при кадровій частоті 50 і 60 Гц. Для прямих трансляцій, відеоконференцій та вебінарів доцільно використовувати постійну

швидкість бітрейту або CBR, а для збереження в якості навчальних матеріалів VBR-змінний бітрейт.

Відео послідовність формується з наступних один за одним кадрів, які змінюються 25 або 30 разів у секунду. Якщо оцифрувати їх з глибиною кольору в 16 біт, то кожен кадр буде займати приблизно 1,2 МВ, а це значить, що для зберігання 1 секунди відео буде потрібно приблизно 30 МВ дискового простору, і на півторагодинний фільм піде близько 150 GB. Це як мінімум уп'ятеро більше ємності жорстких дисків, якими оснащуються сучасні домашні комп'ютери. Для зменшення обсягу файлів, що містять оцифроване відео, на сьогоднішній день використовується MPEG-4- стандартом стиснення мультимедійного контенту. А H.264 за рахунок того, що має суттєво розширені можливості у порівнянні з іншими доступними кодеками, став основним при розробці програмного забезпечення для відеоконференцій і не тільки.



## **2 ОГЛЯД СУЧАСНИХ ПЛАТФОРМ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ. ТЕНДЕНЦІЇ ВИКОРИСТАННЯ ІНТЕРАКТИВНОГО ВІДЕОКОНТЕНТУ**

Перспективне використання можливостей Інтернету як глобального інтерактивного дистанційного навчального середовища є безмежним, а технологічний потенціал дуже великий і не досліджений у повній мірі. Використання нових методик навчання, повністю змінює парадигму сучасної освіти. Новий інструментарій, що заснований на навчальних відеоматеріалах, особливо адаптувався з появою швидкісного Інтернету, що повністю видозмінило принципи, що покладені в основу класичної (радянської та ранньої пострадянської) освіти. Сучасні технології надають кожному можливість стати автором чи співавтором інформації в мережі Інтернет.

Використання сучасних технологій роблять освіту більш інтерактивною, тобто такою, яка залучає до участі і надає можливості активної взаємодії на всіх рівнях. Серед можливостей, які відкриває дистанційне навчання— самоосвіта протягом тривалого періоду; вільне спілкування у мережі з однодумцями, колаборація та кооперація між учнями та вчителями; вплив на навчальний процес (зворотній зв'язок, висловлення реакції та відгуків, можливість ділитись навчальною інформацією); вільний та великий вибір навчальних матеріалів, програм; курсів, колективна взаємодія; креативність та творчість у медіапросторі. сприяє покращенню комунікативних та колективних навичок, індивідуальних можливостей створення власного навчального контенту. Усе це надає можливість ділитись розробками з іншими користувачами курсів, а також колективно добувати знання. Сьогодні можливо працювати з веб-платформами без знань мов програмування – кожен при бажанні має можливість створити блог, відео та віртуальну спільноту тощо[11].

Система дистанційного інтерактивного навчання, можемо синтезувати та поєднувати такі модулі: адміністрування системи; модуль організації та підтримки навчального процесу; модуль розроблення та підтримки тестів; модуль розроблення та представлення всіх видів навчальних матеріалів у системі; модуль введення-виведення навчальної інформації у різних форматах; модуль інтерактивної взаємодії користувачів курсів: лектор–студенти, студенти–студенти, студенти–лектор; модуль реєстру активності користувачів.

## 2.1 Інтерактивний відеоконтент у дистанційному навчанні

Існує чимало причин того, чому люди обирають інтерактивні відео для навчання. По-перше, це неймовірно різноманітне середовище, яке можна застосувати до будь-якого навчального процесу.

Дослідження, проведене Forrester Research, показує, що заклик до дій (СТА) всередині відео відображає підвищення коефіцієнта конвертації на 5% -12% над лінійним відео. Інтерактивний контент отримав відгуки від учнів "Скажи мені, я забуду. Покажи мені, я запам'ятаю. Залучіть мене, я зрозумію".



Рисунок 2.1– Приклад розгалуженого сценарію для інтерактивного контенту

Оскільки користувач працює за допомогою розгалужених сценаріїв, їм пропонуються кілька реалістичних варіантів, з якими вони, імовірно, стикаються у цій галузі. Заклик до дії повинен бути зроблений менш ніж за 10 секунд.

Порівняно з неінтерактивним відео, яке перші кілька хвилин переживає високі показники відхилення переглядів, інтерактивне відео відображає рівень завершення 90 відсотків і вище. А оскільки глядачі часто повертаються щоб дослідити всі варіанти відповідей у інтерактивному відео, то найчастіше можна спостерігати повторні перегляди

Відео або класичні відео – чудовий варіант для навчання, оскільки вони пропонують високоякісний та захоплюючий формат. Проте вони не забезпечують будь-який контроль над учням за межами відтворення / паузи / зупинки.

Це, звичайно, не є проблемою, якщо у вас є короткі відеоролики, але це може перешкоджати навчанню, коли: відео довші.

Стратегії інтерактивного відео залучають саме цей контроль, щоб забезпечити інтерактивність у відео та зробити його більш доступним в курсах e-Learning існує декілька методів[12]:

1. Інтерактивне відео на основі мікронавчання для концептуального навчання. Цей підхід поєднує в собі взаємодію (відповідну навчальному курсів e-Learning) для створення високоякісного досвіду навчання.

2. Інтерактивні відеоролики з експертами

Інтерактивні відеоролики можуть бути ідеальними для створення відеозаписів. Учні захоплюються експертними порадами та рекомендаціями. Використовуючи анімації на основі інфографіки та типографіки для підтримки діалогів експерта, він створює захоплюючий віртуальний досвід класу. Крім того, взаємодії різних типів учнів на певних етапах роблять навчання ефективним.

3. Віртуальна класна кімната, що поставляється з використанням інтерактивного формату відео

Легкість та гнучкість доступу до них у дорозі створюють веб-трансляції–підкасти, які ідеально підходять для підтримки продуктивності. Вони приходять з додатковою перевагою – бути стислими та розважальними.

4. Сценарний тренінг із прийняття рішень із використанням інтерактивних відео (відео YouTube). Виклик пасивності може стати шоу. Ви можете перевернути це, вибравши інтерактивне відео на основі навчання, яке забезпечує широкий діапазон взаємодії користувачів (включаючи оцінювання та розгалуження на основі відповіді учня).

Інтерактивний відеофільм – це фільм, в якому глядач на певних етапах перегляду може обирати продовження сюжету. Способи застосування інтерактивних фільмів дуже різноманітні: 1. Створення завдань-квестів 2. Створення відеоінструкції 3. Спосіб реалізації ділової гри 4. Навчання роботі з ПЗ 5. Тестування, моделювання процесів 6. Відпрацювання навичок розв'язування задачі 7. Створення рольових ігор. Створення інтерактивних фільмів дозволяє найбільш повним чином використовувати переваги інтерактивної складової навчання. У процесі перегляду фільму студенту надається можливість управляти демонстрацією як окремих інформаційних фрагментів, так і цілих розділів фільму. Як правило, той кого навчають сам визначає найбільш складні, з його точки зору, фрагменти досліджуваного матеріалу і, або повторно повертається до їх перегляду, або приступає до більш поглибленого їх вивчення. Таким чином, забезпечується індивідуальний підхід до навчання навіть на етапі самопідготовки. Досвід застосування інтерактивних навчальних фільмів свідчить про те, що даний напрямок є дуже перспективним як при навчанні в навчальних центрах, так і при самостійній підготовці. За допомогою відео та звуку може бути переданий сценарій реальній життєвій ситуації. Можливості інтерактивного відео (нелінійний підхід) можуть дозволити учню вивчити ситуацію у вигляді, максимально наближеній до реальності. Учень приймає рішення і може спостерігати за результатами своїх дій. Однак існують

деякі обмеження, оскільки складність реальної життєвої ситуації ніколи не може бути відтворена з абсолютною точністю. Незважаючи на ці можливі недоліки, здатність інтерактивного відео конкретизувати навчальний матеріал і здійснювати нелінійний підхід, дозволяє інтенсифікувати навчальний процес і реалізувати особистісно-орієнтований підхід до навчання, який передбачає відповідальність учня за своє навчання та реалізацію особистих стилів і цілей навчання. В даний час створити інтерактивне відео можна безпосередньо в YouTube. Записані фрагменти відео лекції редагуються в YouTube. YouTube має великий інструмент для створення і редагування відеоанотацій, який дозволяє робити відео інтерактивним. Про інтерактивному відео можна сказати, що це технологія, яка дозволяє користувачеві вибирати Популярні сюжети, бачити виноски про об'єкти, вибирати посилання на інші ресурси. Технологія інтерактивного відео може бути корисна, перш за все, для дистанційного навчання. Прожектори дозволяють створювати активну ділянку в будь-якій частині відео, яка посилається на інше відео на YouTube. Це дуже зручно для уроків та вікторини. По суті, можна задавати студентові запитання, чи серію запитань – і коли він відповідає показати йому персоналізовану відповідь відео з поясненнями і посиланнями на чергове запитання. При цьому інтерактивний навчальний фільм відрізняється від існуючих електронних підручників і систем дистанційного навчання, в яких застосовується енциклопедичний підхід (про все потроху), тим, що інтерактивний навчальний фільм позиціонується як навчальний посібник в концентрованому вигляді, що дає тільки потрібну інформацію по предмету.

## 2.2 Застосування технологій віртуальної реальності для навчання і досліджень.

Віртуальна реальність – ідеальна навчальне середовище. Сприйняття віртуальної моделі з високим ступенем достовірності дозволяє якісно і швидко готувати фахівців в різних областях: авіація, управління технологічними процесами, медицина, дистанційне керування технічними засобами тощо Тут слід приділяти особливу увагу якості системи візуалізації, тому що картинка повинна бути абсолютно достовірною.

Освіта з використанням віртуальної реальності, дозволяє наочно вести лекції і семінари, проводити тренінги, показувати які аспекти реального об'єкта або процесу, що в цілому дає колосальний ефект, покращує якість і швидкість освітніх процесів, і зменшують їх вартість. Технології віртуальної реальності дозволяють в повній мірі використовувати те, що людина отримує 80% інформації з навколишнього світу за допомогою зору, при цьому люди запам'ятовують 20% того, що вона бачать, 40% того, що вона бачить і чує, і 70% того, що вони бачать, чують і роблять.

Компонування систем віртуальної реальності спільно з різними елементами тренажерних технологій (крісла на гідроприводах, платформах, системи управління зі зворотним зв'язком, трекінг та ін.) Дозволяє створювати тренажери і атракціони з великим коефіцієнтом адекватності (занурення). Наприклад використання стереовізуалізації надзвичайно необхідно для тренування льотчиків, тому що під час дозаправки в повітрі або посадці необхідно відчуття перспективи і дістанції.

Сприйняття глибини і тривимірності інформації критично важливо для дистанційного керування апаратами – батискафами, різними роботами і безпілотними літальними апаратами і т.д.

Величезним напрямком технологій віртуальної реальності в науці, є візуалізація складних і розрізнених даних в єдину, наочну модель досліджуваного об'єкта або процесу, яка дозволяє експертам в повній мірі використовувати свою інтуїцію і зорову пам'ять при різних дослідженнях, наприклад в біології, історичних реконструкціях.

Сучасна медицина немислима без візуалізації досліджень внутрішніх органів людини (томографія, УЗД, рентген і т.д.) При створенні нових ліків, отримавши в своє розпорядження тривимірну комп'ютерну модель тих чи інших молекул і маючи можливість управляти їх рухом, експериментатор може домогтися суміщення моделей (коли «опуклості» однієї точно відповідають «западин» іншій); в цей час і відбувається так звана стикування, яка імітує реальні процеси при взаємодії реальних біологічних молекул. «Побувавши» при цих процесах, вчені можуть зрозуміти, як реальні молекули зв'язуються між собою за допомогою з'єднання певних рецепторних ділянок, і пояснити, наприклад, як вірус проникає в клітину. Це дозволяє, в свою чергу, спробувати синтезувати речовини, що блокують активність вірусу.

В цілому, можливості технологій віртуальної реальності для навчання і досліджень мають надзвичайно високий потенціал застосування.

Концепція використання технологій віртуальної реальності для навчання і науки в повній мірі реалізована в програмно-апаратному комплексі Віртуальної реальності для освіти.

### **Основні функції VE 3D ieCenter**

Інтерактивна високоякісна візуалізація для наукових і освітніх цілей, віртуальне моделювання та прототипування різних процесів і об'єктів[13];

Створення інтерактивних освітніх курсів і їх подальша демонстрація для студентів і викладачів в системах віртуальної реальності, звичайних РС, 3D Intranet і 3D Internet;

3D-візуалізація наукових даних в реальному масштабі часу і високоякісна візуалізація імітаційного моделювання;

VIP презентації для керівництва та осіб, котрі приймають рішення;

Віртуальна відпрацювання взаємодії людини і різних технічних пристроїв і систем;

Центр створення інтерактивних віртуальних макетів, освітніх тренажерів і симуляторів.

Переваги VE 3D ieCenter:

Простота використання, економія часу при візуалізації даних, простота роботи з ПО Virtools або 3DVIA Studio Pro, яке не вимагає спеціальних навичок програмування;

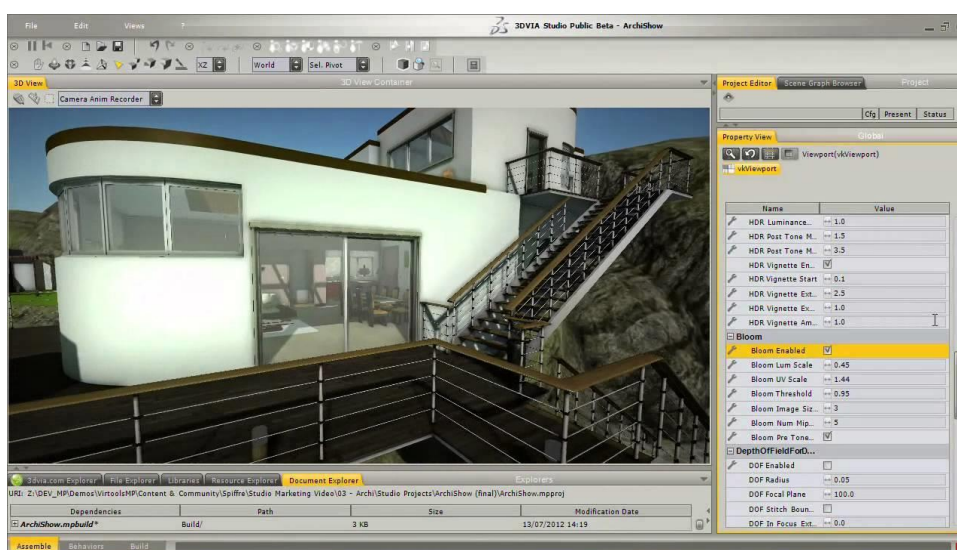


Рисунок 2.1– Інтерфейс програми 3DVIA Studio Pro

Скорочення площ, необхідних для розміщення лабораторного та навчального обладнання, за рахунок використання комп'ютерних імітаційних моделей, тренажерів і симуляторів;

Робота в реальному масштабі часу з інтерактивними віртуальними макетами, моделями, отримання реального досвіду і знань ще на етапі розробки;



Використання концепції 3D. Можливість створення інтерактивних 3D освітніх програм і подальша їх демонстрація в системах віртуальної реальності (3D візуалізації), демонстрація на звичайних PC (створення CD, DVD), створення 3D сайтів (за технологією 3D інтернет)[14]; Можливість інтеграції в існуючі аудіо-візуальні комплекси. Можливість демонстрації звичайної 2D інформації. Состав 3D ieCenter: Проекційна система 3D візуалізації різної конфігурації (від одного екрана до шести, або складних конфігурацій). Як правило, для сприйняття 3D інформації потрібні спеціальні окуляри, що дозволяють користувачеві сприймати стереоскопічний ефект; Графічний генератор. Являє собою потужну спеціалізовану графічну станцію, або графічний кластер дозволяє абсолютно синхронно обробляти і видавати необхідний потік візуальної інформації. Так само до графічного генератору відносяться засоби дистанційного контролю і управління всім програмно-апаратним комплексом віртуальної реальності; Програмний комплекс. (Virtools, 3DVIA Studio Pro) являє собою набір різних модулів: Інструментарій розробки, засоби портування 3D інформації, модулі візуалізації, засоби роботи з периферією віртуальної реальності, бібліотеки штучного інтелекту, бібліотеки фізичних законів. У всьому світі більше 400 Вузів та наукових центрів використовують ці програми. Набір різних пристроїв, що дозволяють "підсилювати" занурення у віртуальну реальність і ступінь інтерактивності взаємодії з досліджуваним набором віртуальних моделей за допомогою імітації різних каналів сприйняття інформації (слух, нюх, дотик, смак). Це системи трекінгу різних типів, 3D миші і такі системи управління, як рукавички віртуальної реальності, пристрої імітації тактильних відчуттів – haptic, пристрої імітації запахів, звукова багатоканальна система і т.д. Компанія VE Group інтегрувала в концепції 3D ieCenter всі останні світові досягнення в області програмно-апаратних комплексів віртуальної реальності. Вартість програмно-апаратних комплексів віртуальної реальності 3D ieCenter визначається, в залежності від конкретних вимог користувача, розміру

приміщення, функціональних можливостей, існує можливість вибору комплектації і розмірів.

3D інтернет для дистанційної освіти. Представлена 3D інформації в Інтернет відкриває нові перспективи для інформаційних технологій в навчанні. Останні тенденції розвитку інформаційних технологій все сильніше пов'язані з поняттями «Інтернет», «тривимірна графіка», «віртуальна реальність». Для з'єднання цих понять, тобто вирішення проблеми відображення просторових об'єктів у всесвітній мережі розробляються різні технології, які об'єднані загальним терміном web 3D або 3D Інтернет. Зусилля вирватися за рамки двовимірного подання інформації привело до створення динамічних віртуальних технологій. Першою спробою в цій області була розробка мови VRML[15]. Реалізації цієї мови на сьогоднішній день не цілком задовільняють потреби розробників і користувачів. Нове покоління технологій web3D або 3D інтернет повною мірою реалізовано в програмному комплексі 3DVIA Studio Pro. Використання модуля Multi-user Publisher і 3DVIA Player дозволяє публікувати в інтернеті віртуальні освітні середовища, створені за допомогою модулів розробки 3DVIA Studio. Потім за допомогою 3DVIA Player кожен бажаючий може відтворити віртуальний світ за допомогою свого персонального комп'ютера, підключеного до інтернету. Це відкриває воістину гігантські перспективи в області дистанційної освіти.

### **2.3 Прямі трансляції для дистанційноо навчання**

Останнім часом в мережі Інтернет активно розвиваються технології стрім-нарад[16] (в перекладі з англ. stream - потік), серед веб-користувачів цей термін означає пряму трансляцію що відбувається на екрані персонального комп'ютера або в житті. На даний момент популярними платформами для проведення стрім-віщування є такі сервіси- YouTube, Periscope, LiveCoding, Twitch, а також

соціальні мережі – Вконтакте, Facebook, Instagram. Початок найпростішої прямої трансляції (з камери мобільного пристрою) здійснюється в один клік. При цьому всі користувачі, підписані на автора трансляції, дізнаються про неї за допомогою швидких, автоматично створених повідомлень, нагадувань з можливістю підключення до перегляду та участі в масовому чаті. Для організації та проведення складніших стрім-нарад потрібно використання персонального комп'ютера, веб-камери і спеціального програмного забезпечення (відеокодера).

У рамках нашого дослідження під технологією стрім-мовлення в процесі навчання будемо розуміти сукупність методів підготовки, трансляції і збереження аудіовізуальної інформації з екрану персонального цифрового пристрою і веб-камери при дистанційному навчанні з використанням сучасних телекомунікаційних сервісів мережі Інтернет. Безумовно, сучасні стрімтехнології можуть мати широке застосування в системі вищої освіти, в тому числі і при навчанні ІТ-дисциплін. Схема реалізації прямої трансляції зображена на малюнку 2.2

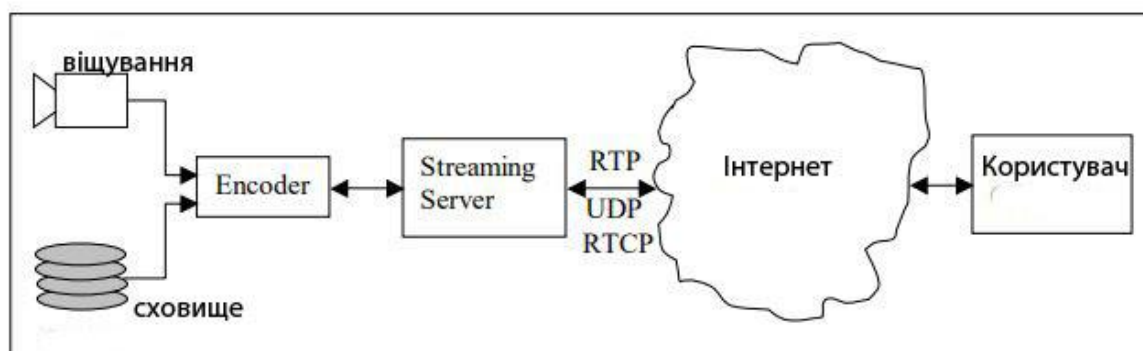


Рисунок 2.2– Схема реалізації стрім-віщування

Для проведення стрім-нарад необхідно наявність:

-персонального цифрового пристрою (комп'ютер, ноутбук, планшет) з високими технічними характеристиками;

-високошвидкісне підключення до Інтернету;

-мікрофона і веб-камери.

Для більш просунутих трансляцій рекомендується використання професійного обладнання: звукові і відеокарти, мікрофони, камери, аудіо– та відеомікшери. Як було згадано раніше, в якості платформи для проведення стрім–нарад пропонується використання сервісу YouTube і вільне програмне забезпечення OBS. Використання в комплексі даних продуктів має ряд переваг в порівнянні з аналогічними веб-сервісами і програмами[17]:

- висока якість стрім-мовлення – до 1920x1080, 60 fps;
- можливість перемотувати стрім в початок і подивитися будь-який момент у реальному часі;
- запис трансляції доступен моментально після закінчення;
- щоб залишати коментарі, не потрібно наново реєструватися, так як у багатьох вже є аккаунт Google;
- можливість редагувати збережену трансляцію;
- YouTube – один з найбільших місць для розміщення відеофайлів в світі;

OBS має широкі можливості: захоплення зображення, відео та аудіо з різних джерел (веб-камери, екран робочого столу, вікна, ігри, програми, графічні зображення, презентації та ін.), використання декількох сцен, налаштування їх розташування і переходів між ними. проектування методів використання стрім–технологій у навчальній діяльності

Основна ідея використання стрімтехнологій у навчальній діяльності полягає у виконанні наступної послідовності дій:

1.Публікація в новинній стрічці навчального онлайн-курсу списку тем майбутніх трансляцій з описом приблизною дати і часом проведення прямих ефірів, також можливо прикріплення необхідних документів до ознайомленню до

майбутніх зустрічей. Варто зазначити, якщо мова йде про підготовку майбутніх фахівців в області ІТ, то це можуть бути прямі трансляції не тільки від викладача, але і від самих студентів.

2. Підготовка матеріалів, налаштування обладнання, програм і сервісів, необхідних для проведення стрім-віщування, публікація в новинній стрічці більш точної інформації з прикріпленням посилання на його показ.

3. Проведення стрім-мовлення в реальному часі, серед учасників можливо спілкування в режимі групового чату.

4. За закінчення стрім-мовлення автоматично генерується готовий скрінкасти, який можливо відредагувати (Додати коментарі, прибрати зайві моменти та ін.) і заново завантажити в матеріали навчального онлайн-курсу. Більш детально застосування скрінкастинга при навчанні ІТ-дисциплін описано в роботі[18]. Для проектування методів використання стрім-технологій у навчальній діяльності були використані ідеї, запропоновані

#### **2.4 Основний принцип побудови типової архітектури систем дистанційного навчання**

З розвитком комп'ютерних мереж та інформаційних технологій дистанційне навчання або E-learning розвивається швидко. Веб-навчальні середовища стають дуже популярними. Дистанційне навчання означає спосіб навчання, який доставляється або покращується за допомогою електронних технологій. Дистанційне навчання визначається як "ефективний навчальний процес, створений шляхом поєднання цифрової доставки навчального контенту з підтримкою та послугами"[19] Це інструкція, що повністю передана електронним способом через веб-браузер, через Інтернет або через інтрамережу (приватну інтернет мережу) або через компакт-диск чи мультимедійні платформи DVD. Використання інтелектуальної інформації в архітектурі дистанційного

навчання дозволяє отримати персоналізовану систему електронного навчання, адаптовану до цілей та характеристик кожного учня . Атрибутами платформ зазвичай є:

- Постійність: навчальна інформація доступна безперервно ,а отже час у який користувачі можуть засвоювати та вивчати інформацію обирається кожним персонально.
- Автономія: користувачі можуть вибрати завдання і мати можливість вирішувати пріоритет завдання без втручання сторонніх осіб.
- Соціальні можливості: користувачі можуть спілкуватися з іншими учнями через соціальні мережі та співпрацювати для виконання завдання.
- Реакційна здатність: створювачі курсів можуть робити відповідну реакцію відповідно до змін у потребах користувачів, сприймаючи відгуки.

У традиційних системах дистанційного навчання існує один шар, а всі агенти та служби пов'язані між собою. Сучасні веб–платформи для дистанційного навчання зазвичай використовують архітектури з 4 шарами, спостерігати приклад якої можна на рисунку 2.1. Середній шар заснований на Multi Agent System –це система, що базується на декількох взаємодіючих інтелектуальних агентах. Типовий метод оновлення вмісту через "Питання-відповідь" між електронними учнями та цією моделлю за допомогою SCORM (збірник специфікацій і стандартів, розроблений для систем дистанційного навчання. Містить вимоги до організації навчального матеріалу і всій системи дистанційного навчання. SCORM дозволяє забезпечити сумісність компонентів і можливість їх багаторазового використання: навчальний матеріал представлений окремими невеликими блоками, які можуть включатися в різні навчальні курси і використовуватися системою дистанційного навчання незалежно від того, ким, де і за допомогою яких засобів вони були створені. SCORM заснований на стандарті XML.) [19], для повторного використання та обміну вмістом курсу.

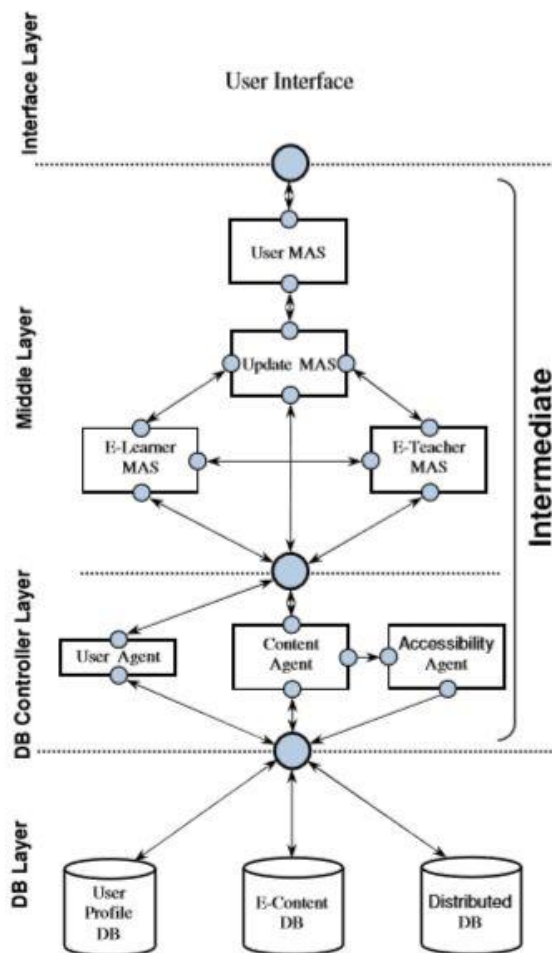


Рисунок 2.3 – Архітектура системи дистанційного навчання

Чотири шари в цій моделі, а саме: шар інтерфейсу, середній шар, шар контролера база даних і рівень баз даних. Служба керування інтерфейсом використовується для зв'язку між інтерфейсним шаром та середнім шаром. Ця служба отримує запит від інтерфейсного шару і, якщо це запит на перевірку, відправляє його в середній шар, після підготовки відповідної відповіді в середньому шарі, ця служба відправляє відповідь на інтерфейсний рівень. Служба підключення використовується для зв'язку середнього шару та контролера рівня DB. Ця служба отримує запит від багатоагентних систем у середньому шарі, і якщо це запит на перевірку, він відправляється відповідному агенту на шар контролера DB після підготовки відповідної відповіді на рівні

контролера DB, ця служба відправляє відповідь на середній шар. Служба доступу до даних використовується для зв'язку між шаром регулятора DB та шаром DB. Служба доступу до даних управляє доступом до баз даних (вставлення, виділення, оновлення та видалення).

**Інтерфейсний шар (Interface layer).** Користувачі використовують цей шар для взаємодії з системою, і цей рівень показує прийнятний користувацький запит. У процесі навчання користувач може двома способами отримати вміст у цьому шарі:

- Вибрати вміст вручну
- Автоматично вибирати вміст за системою

**Середній шар (Middle layer).** Цей шар використовується для підтвердження запитів користувачів, ідентифікації користувача, визначення доступу користувача, оновлення вмісту, вибору автоматичного вмісту для користувача на основі продуктивності, створення та оновлення вмісту за допомогою викладача або методу відповіді на запит і відповідь, а також керувати запитом користувача власноруч.

Шаблон контролера баз даних контролера DB (DBCL) - це шар, який забезпечує спрощений доступ до даних, що зберігаються у постійному сховищі на рівні BD. Цей шар відповідає за комунікації з BD, вибір попереднього рівня запитуваного вмісту та редагування інформації про користувача. Цей шар складається з трьох агентів. Користувальницький агент використовується для ідентифікації користувача, що додає чи видаляє користувача, зберігає продуктивність користувача тощо. Запит на вміст, отриманий від служби з'єднання до агента контенту. Контент-агент, що доставляє вміст, повинен отримати запит (CNR) на службу доступу до даних для пошуку в BD електронного контенту, якщо CNR існує в електронному контенті BD, служба доступу відповідає цьому CNR на контент-агент. Якщо CNR не існує, служба



доступу до даних повертає false до агента контенту, і цей агент відправляє SNR до агента доступності та контакту доступу з розподіленою BD через службу доступу до даних, а потім повертає цей CNR або false для агента контенту. Якщо ми потрапляємо в CNR в цьому шарі, то він відправляє назад на попередній шар, і ми надсилаємо вміст запиту update

**Блок DB Layer.** Цей рівень складається з баз даних, а саме BD профілю користувача. У профілю користувача ми можемо зберігати налаштування користувача (переваги користувальницьких засобів типу букви, кольору, іконки, шрифт тощо), продуктивність користувача (час, що витрачається користувачем в кожному розділі курсу, час, витрачений на отримання правильної відповіді, для тестової веб-сторінки - час, витрачений на відповідь на всі питання та ознаки пунктуації, отримані на тестовій сторінці).

## **2.5 ATutor – модульна система дистанційним керуванням навчанням з відкритим кодом**

ATutor є системою управління навчання – Web-based Learning Content Management System (LCMS)[20]. Його використання дозволяє викладачам легко організувати різні курси навчання. Студенти ж отримують адаптивну і просту середу навчання. Доступність вихідного коду і відкриті інструменти, що застосовуються для побудови сервера курсів, дозволяють в разі крайньої необхідності внести будь-які зміни. За всього необхідного для створення і управління курсами і процесом навчання, в його складі є і засоби обміну повідомленнями. Особлива увага приділяється безпеці. За допомогою додаткових модулів можна наростити функціональність. Наприклад забезпечити роботу з фото у тексті та відео обміну інформацією з іншими навчальними системами, конференції, вебінари та інші.

ATutor містить ряд технологій, що гарантують те, що вміст буде доступно для всіх потенційних користувачів, включаючи тих хто використовує повільні інтернет канали, також ранні версії Web-браузерів і старі монітори, і для людей з обмеженою працездатністю. Для цього використовується приховування зайвих елементів управління, динамічне меню, мітки полів форми та альтернативна навігація, фіксація останньої позиції, набір гарячих клавіш. Це дозволяє користувачеві відразу переходити до вмісту.

Система підтримує три види курсів: публічні – доступні всім, захищені – вимагають реєстрації та приватні, що вимагають крім реєстрації, підтвердження доступу інструктором. Курси доступні поточному користувачеві відображаються в "Browse Courses" (Всі курси). Експорт вмісту можливий в IMS / SCORM сумісні пакети, які можуть переглядатися автономно і / або додаватися в іншу e-learning систему. Для зв'язку з учнями реалізовано такі можливості як опитування громадської думки, новини, повідомлення та тематичні форуми, електронна пошта, RSS потоки. Для перевірки засвоєння матеріалу можуть бути створені тести. Додатково з версії 1.4 в ATutor було інтегровано два елементи, що розширюють можливості AChecker і ACollab. AChecker - експериментальна розробка Adaptive Technology Resource Center [21] дозволяє визначати доступність ресурсів в мережі. ACollab являє собою єдину робочу середу для груп користувачів займаються спільними дослідженнями і розробкою документації. До речі він може використовуватися і як самостійний додаток

## **2.6 Moodle**

MOODLE (Modular Object-Oriented Dynamic Learning Environment – модульна об'єктно-орієнтоване середовище) – це відкрита система управління навчанням, орієнтована на організацію взаємодії між викладачем і студентами, і яка також використовується для створення дистанційних курсів. Середою

MOODLE[22] розроблена на PHP з використанням SQL-бази, має модульну архітектуру, що дозволяє легко розгалужувати можливості.

Система MOODLE надає зручні засоби управління контентом і різні форми організації занять. Дистанційний курс може включати різні елементи: лекції, практичні (самостійні) завдання, форум, чат та ін. При цьому можна використовувати текст, презентації, таблиці, схеми, графіка, відеоматеріали, посилання в мережі Інтернет, допоміжні файли і інші матеріали. За результатами виконання студентами завдань викладач може виставляти оцінки і коментувати.

Оскільки система MOODLE є орієнтованою на використання в дистанційному навчанні, вона має широкий набір засобів комунікації. Можливості, що надаються системою, дозволяють забезпечити індивідуальну роботу викладача з кожним студентом. Це не тільки електронна пошта та обмін вкладеними файлами, але і форуми, чати, блоги та ін.

Щоб отримати доступ до ресурсів необхідно бути зареєстрованим в системі. Для початку роботи з системою необхідно зайти на сайт навчального закладу, де встановлена система Moodle. На рис. 2.4 показана сторінка авторизації. При кожній спробі ввійти в систему проводиться ідентифікація користувача. Система ідентифікує користувача за допомогою реєстраційного логіну та пароля. При вході в систему вносять необхідну інформацію про учасника навчального курсу: особові дані, назва вибраної спеціальності. Реєстрацію користувачів навчального процесу, проводить куратор, також для студентів передбачена можливість самостійно зареєструватись. Ця функція може бути активована адміністратором системи, а її використання залежить від політики конфіденційності навчального закладу.

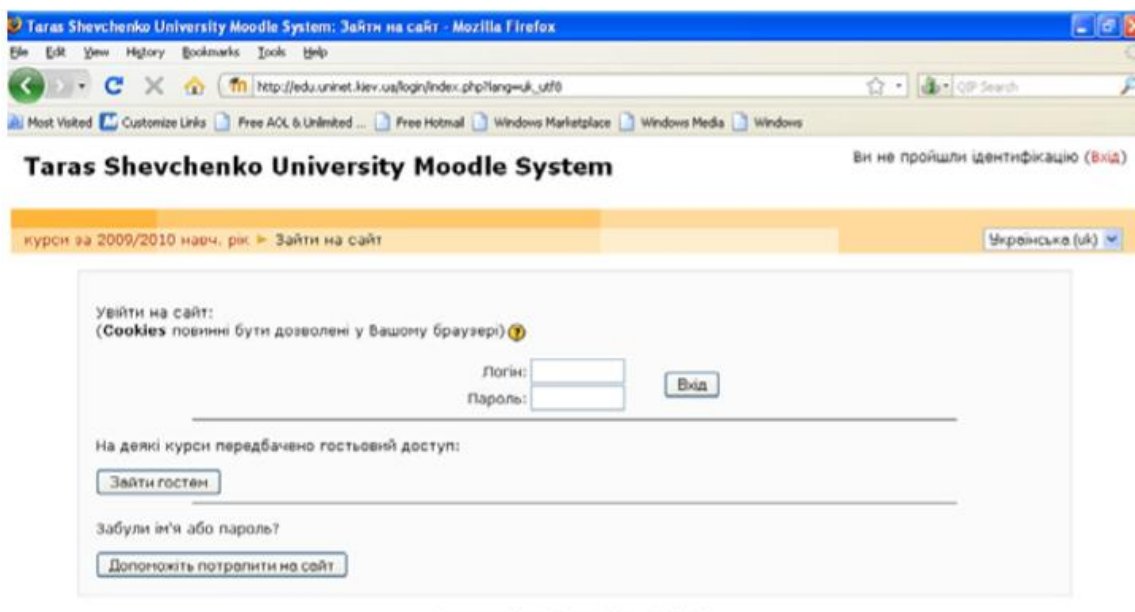


Рисунок 2.4– Сторінка авторизації системи Moodle

Зазвичай система Moodle може бути доступною для декількох профілів таких як користувач – викладач та користувач– студент, що при роботі мають різний рівень доступу. Також існує дві можливості входу в систему: 1 – гостьовий, для незареєстрованих користувачів. Гість не може створювати повідомлення на форумах, редагувати сторінки спільних проектів, проходити тести, виконувати завдання. 2– авторизований доступ для зареєстрованих користувачів, що надає можливість отримувати повний доступ до ресурсів СУН у відповідності до прав наданих їм адміністратором системи.

Особливості роботи з курсом залежать від того, який формат було обрано викладачем при створенні адміністратором облікового запису. СУН Moodle надає можливість створювати курси в таких форматах[23]:

Щотижневий – заняття розподіляються по тижнях, розклад організовується потижнево;

Тематичний – навчальний матеріал розподіляється за темами. Він включає тематичні плани, які відображають зміст наповнення навчальних курсів та робочі плани, у яких показаний розподіл матеріалу курсу по лекціях. Зазвичай, лекції

групують у тематично завершені модулі. По закінченні роботи над модулем, проводять тестування ступеня засвоєння пройденого матеріалу. Формат форум (Соціальний) – побудований на основі форуму ("дошка оголошень"). Він може використовуватися не тільки для створення навчального курсу, але і як одна велика "дошка повідомлень". Найбільш зручними для створення курсу є потижневий та тематичний формати. Потижневий формат застосовують, в разі створення курсу, для вивчення якого істотним є послідовність вивчення матеріалу. Тематичний формат пристосований для вивчення курсу, який структуровано у відповідності з навчальними темами дисципліни (курсів лекцій)[24]. Такий формат є більш прийнятним у випадку створення курсу з тематичними модулями.

На рис. 2.5. зображена WEB-сторінку навчального курсу та блоки з яких вона складається. Вона містить такі елементи: 1 навігаційне меню в якому відображається повний шлях від головної сторінки СУН Moodle до поточної; 2 загальний зміст містить перелік документів про мету курсу, програми та ін.; 3 розклад – визначає кількість тижнів викладання курсу. Кожне поле (комірка) представляє тиждень. Всі ресурси розділу призначені для роботи на одному тижні (лекції, практичні завдання та ін.) мають бути розміщені у відповідному полі (комірці). Курс, таким чином буде поділений на тижні;



Рисунок 2.5- Сторінка редагування навчального курсу

- кнопка вмикання режиму редагування доступна лише для викладачів курсів. Після переходу до режиму редагування, викладач може виконувати різні операції: змінювати розташування блоків на сторінці, змінювати та додавати навчальний матеріал, завдання тощо;

- блоки: (секції, на які розбитий інтерфейс WEB-сторінки) де знаходиться додаткова інформація, а також засоби, що дозволяють виконувати додаткові функції. Блоки викладач може розташувати, так як йому буде зручно (або так, як, на його думку, буде зручно студентам).

На рис.2.5 ці блоки показані в правій частині рисунку послідовно зверху до низу: • новини форуму – обговорення навчального матеріалу, що є в даному курсі; • майбутні події – список заходів, які заплановані і будуть відбуватися найближчим часом, інформація про додаткові навчальні заходи; • останні новини – останні повідомлення на форумі новин; • звіт про діяльність – останні дії по редагуванню навчального курсу та завдань до нього; • керування – пропонує різні можливості для редагування курсу .

### **Висновки до розділу**

Одна з головних задач сучасної освіти – це створення стійкої мотивації у студентів до отримання знань, інша – пошук нових форм та інструментів освоєння цих знань за допомогою творчих рішень у студентів. Дистанційне навчання або E-Learning – це спосіб, за допомогою якого навчальний контент викладається за допомогою електронних технологій. У традиційній системі електронного навчання існує лише один шар, а всі агенти та послуги пов'язані між собою. Розглянута архітектура системи електронного навчання в даній роботі містить чотири рівні рівня. Він базується на парадигмі багатоагентних систем. У цьому дослідженні ми пропонуємо новий метод оновлення змісту через "Питання-відповідь" між електронними учнями. Ми використовуємо еталонну модель об'єкта об'єктивного змісту (SCORM) для повторного використання та спільного використання змісту системи електронного навчання. Для цього була розроблена система, яка застосовує різні технології, які гарантують різні життєво важливі функції для систем електронного навчання. Ці функції включають

інтелект, розподілений характер, адаптивне ставлення, взаємодія, доступність та безпека. Ми застосували багат шарове моделювання, щоб забезпечити гнучку та легку систему електронного навчання

Електронне навчання стало частиною освіти. У даній роботі ми представили наш підхід до розробки багатоагентної архітектури системи електронного навчання. Для цього ми розглянули систему, яка застосовує різні технології, які забезпечують гнучку та легку систему електронного навчального середовища, такі функції, як розвідка, розподіл, адаптивність, взаємодія, доступність, особистість, безпека та використання багат шарового простору. Інтелектуальний агент та технологія MAS – це ефективний метод надання інтелектуальної та персоналізованої системи. Ми запропонували новий спосіб оновлення вмісту через "Питання-відповідь" між електронними учнями за допомогою системи оновлення Multi Agent. За допомогою Агента доступності ця модель може працювати з SCORM для повторного використання та обміну вмістом електронного навчання.

**До переваг дистанційного навчання належать:**

- Свобода і гнучкість. Можливість навчатися одночасно в різних місцях, на різних курсах не тільки в одному, а й у декількох університетах чи навіть країнах.
- Індивідуальність.
- Самостійний вибір студентами темпу навчання, вибору розділів, які варто було б повторити або вилучити.
- Створення власного графіка навчання студентами у звичній для них обстановці і в зручний час.
- Навчання інкогніто (не розголошуючи свого імені) через певні обставини (вік, стан, посаду і т.д.), зареєструвавшись під іншим іменем.
- Отримання освіти інвалідами та людьми з різними відхиленнями. • Набуття студентами таких якостей, як самостійність, мобільність і відповідальність.



- Навчання більшої кількості людей різних вікових груп порівняно з іншими формами навчання.

- Просте формування віртуальних спільнот: викладачів, студентів тощо завдяки використанню сучасних інтернет технологій, за допомогою яких стає можливим обговорення між викладачами певних проблем, вирішення спільних завдань, обмін досвідом чи інформацією тощо.

### **Недоліки дистанційного навчання**

- Немає прямого очного спілкування між студентами та викладачем. Подання матеріалу позбавляється емоційного зафарбування, важко створити творчу атмосферу в групі тих, хто навчається.

- Необхідна наявність відповідного технічного та програмного забезпечення, можливість доступу до інформації та використання засобів дистанційного навчання. Користувач повинен бути забезпеченим персональним комп'ютером та доступом в Інтернет.

- Високі вимоги щодо постановки задачі навчання, адміністрування процесу.

- Ключовою проблемою є проблема аутентифікації користувача при перевірці знань. Неможливо точно сказати, хто на іншому кінці дроту.

- Відсутній постійний контроль над тими, хто навчається; відчувається нестача практичної роботи.

- Великі затрати на проектування та створення системи дистанційного навчання, організацію курсів дистанційного навчання і купівлю необхідного обладнання.

- Розроблення курсів дистанційного навчання є дуже трудомістким процесом, створення однієї години інтерактивного мультимедійного матеріалу займає понад 1000 годин роботи професіоналів.

## **3 АПАРАТНІ ТА ПРОГРАМНІ ЗАСОБИ СТВОРЕННЯ ВІДЕКОНТЕНТУ ДЛЯ СЕГМЕНТУ ДИСТАНЦІЙНОГО НАВЧАННЯ**

### **3.1 Хмарні технології в дистанційному навчанні**

Термін «хмарні обчислення» (англ Cloud computing) застосовуються для будь-яких сервісів, що реалізуються через мережу Інтернет. Суть хмарних технологій полягає в наданні користувачам віддаленого доступу до послуг, обчислювальних ресурсів і додатків (включаючи операційні системи і інфраструктуру) через інтернет. Розвиток цієї сфери хостингу (хостинг–послуга з розміщення обладнання клієнта на території провайдера із забезпеченням підключення його до каналів зв'язку з високою пропускнуою спроможністю) було обумовлено потребою, що виникла в програмному забезпеченні і цифрових послугах, якими можна було б управляти зсередини, але які були б при цьому більш економічними і ефективними. Інтернет-послуги, також відомі як «хмарні сервіси», можна розділити на три основні категорії[25]:

- інфраструктура як сервіс
- платформа як сервіс
- програмне забезпечення як сервіс

У порівнянні з традиційним підходом, хмарні сервіси дозволяють управляти більшими інфраструктурами, обслуговувати різні групи користувачів у межах однієї хмари, а також означають повну залежність від провайдера хмарних послуг. При наданні послуг хмарного сервісу використовується тип оплати «плата за використання». Зазвичай за одиницю виміру часу роботи приймається хвилина чи година користування ресурсами. При оцінці обсягів даних за одиницю виміру приймається Мегабайт інформації, що зберігається. У цьому випадку користувач оплачує рівно той обсяг ресурсів, який їм в реальності використовувався протягом певного часу. Крім того, хмарна інфраструктура надає користувачеві

можливість при необхідності «піднімати» або «опускати» максимальні ліміти ресурсів, користуючись тим самим еластичністю сервісу. Користувачеві хмарних сервісів немає необхідності піклуватися про інфраструктуру, яка забезпечує працездатність наданих йому сервісів. Всі завдання з налаштування, усунення несправностей, розширення інфраструктури та ін. Бере на себе сервіс-провайдер.

Хмари можуть бути публічними або приватними.

Приватна хмара – інфраструктура, призначена для використання однією організацією, що включає кілька споживачів (наприклад, підрозділів однієї організації). Приватне хмара може перебувати у власності, управлінні та експлуатації як самої організації, так і третьої сторони (або будь-якої їх комбінації), і вона може фізично існувати як всередині так і поза юрисдикцією власника.

Публічне хмара (англ. Public cloud) – інфраструктура, призначена для вільного використання широкою публікою. Публічна хмара може перебувати у власності, управлінні та експлуатації комерційних, наукових і урядових організацій (або будь-якої їх комбінації). Публічна хмара фізично існує в юрисдикції власника постачальника послуг.

Гібридна хмара (англ. Hybrid cloud) – це комбінація з двох або більше різних хмарних інфраструктур (приватних, публічних), що залишаються унікальними об'єктами, але пов'язаних між собою стандартизованими або приватними технологіями передачі даних і додатків (наприклад, короткочасне використання ресурсів публічних хмар для балансування навантаження між хмарами).

Громадська хмара (англ. Community cloud) – вид інфраструктури, призначений для використання конкретною спільнотою споживачів з організацій, що мають спільні завдання. Громадське хмара може перебувати в кооперативній (спільній) власності, управлінні та експлуатації однієї або більше організацій спільноти або третьої сторони (або будь-якої їх комбінації), і фізично існує як всередині так і поза юрисдикцією власника

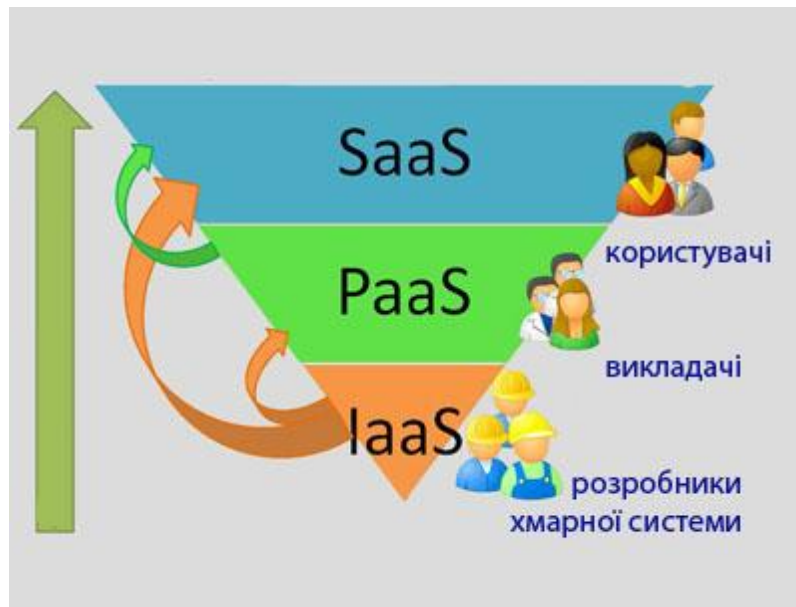


Рисунок 3.1– Модель роботи з хмарною технологією для різних категорій користувачів

На практиці межі між усіма цими типами обчислень розмиті

### **Інфраструктура як сервіс (IaaS)**

Інфраструктура в оренду. Користувачеві надається «чистий» екземпляр віртуального сервера з унікальною IP-адресою або набором адрес і частина системи зберігання даних. Для управління параметрами, запуском, зупинкою цього примірника провайдер надає користувачеві програмний інтерфейс (API).

### **Платформа як сервіс (PaaS)**

PaaS можна уявити як готову до роботи віртуальну платформу, що складається з одного або декількох віртуальних серверів з встановленими операційними системами і спеціалізованими додатками. Більшість хмарних провайдерів пропонують користувачеві вибір з маси готових до використання хмарних середовищ.

### **Програмне забезпечення як сервіс (SaaS)**

Концепція SaaS надає можливість користуватися програмним забезпеченням як послугою і робити це віддалено через Інтернет. Даний підхід дозволяє не купувати програмний продукт, а просто тимчасово скористатися ним при виникненні потреби.

### **Переваги хмарних обчислень**

- Користувач оплачує послугу тільки тоді коли вона йому необхідна, а найголовніше він платить тільки за те, що використовує.
- Хмарні технології дозволяють економити на придбанні, підтримки, модернізації ПО і устаткуванні.
- Масштабованість, відмовостійкість і безпека - автоматичне виділення і звільнення необхідних ресурсів в залежності від потреб програми. Технічне обслуговування, оновлення ПО виробляє провайдер послуг.
- Віддалений доступ до даних в хмарі - працювати можна з будь-якої точки на планеті, де є доступ в мережу Інтернет.

### **Недоліки хмарних обчислень**

- Користувач не є власником і не має доступу до внутрішньої хмарної інфраструктури. Збереження призначених для користувача даних сильно залежить від компанії провайдера.
- Недолік, актуальний для українських користувачів: для отримання якісних послуг користувачеві необхідно мати надійний і швидкий доступ в мережу Інтернет. Особливим є відеоконтент, що створюється у форматі 4к, адже ресурсів для обробки, передачі, та трансляцій подібного матеріалу не має жоден з сучасних представлених провайдерів в Україні.
- Деякі дані не можна довірити провайдеру в Інтернеті не тільки для зберігання, але й для обробки
- Є ризик, що провайдер онлайн сервісів одного разу не зробить резервну копію даних, і вони будуть втрачені в результаті краху сервера.

- Довіряючи свої дані он-лайн-сервісу, ви втрачаєте над ними контроль і обмежуєте свою свободу (Користувач буде не в змозі змінити якусь частину своєї інформації, вона буде зберігатися в умовах, що не підвладні йому).

#### Застосування хмарних технологій

Як приклад використання хмарних технологій в освіті, можна назвати електронні щоденники і журнали, особисті кабінети для учнів і викладачів, інтерактивна приймальня і інше. Це і тематичні форуми, де учні можуть здійснювати обмін інформацією. Це і пошук інформації, де учні можуть вирішувати певні навчальні завдання навіть у відсутності педагога або під його керівництвом. Для цього можна використовувати:

- комп'ютерні програми
- електронні підручники
- тренажери
- діагностичні, тестові й навчальні системи
- прикладні та інструментальні програмні засоби
- лабораторні комплекси
- системи на базі мультимедіа-технології
- телекомунікаційні системи (наприклад, електронну пошту, телеконференції
- електронні бібліотеки та інше.

### **3.2 Реалізація прямої трансляції**

Пряма відеотрансляція – це процес передавання медіаданих (аудіо та відео) таким чином, що дозволяє почати обробку сигналу до того як він буде повністю прийнятий. Прикладом може слугувати програвання відеороликів на веб-сторінках. Пряма трансляція з прогресивною розгорткою, також відома як прогресивне завантаження, означає процес отримання звичайного відеофайлу і

початок його обробки до його повного завантаження. Вона не потребує спеціальних протоколів, але потребує формат, який може оброблятися на основі часткового контенту. Це відома технологія, коли непарні рядки пікселів приймаються та відображаються перед парними рядками. Вони відображаються з половинною роздільною здатністю, поки остаточні рядки не заповнять повне зображення. Потокове передавання з прогресивною розгорткою не володіє гнучкістю справжнього потокового передавання, оскільки швидкість передавання даних не може регулюватись “на льоту” і передавання не може бути розбите на декілька потоків. Якщо таким чином файл доставляється швидко і глядач слухає чи дивиться лише початок файлу, то це звужує пропускну здатність каналу. Глядачу надається весь файл, який він може завантажити без усяких зусиль. Справжня пряма трансляція використовує потоковий протокол для керування передаванням. Отримані пакети не складуються у файл; хоча, не слід використовувати пряму трансляцію для захисту від копіювання – якщо немає шифрування від сервера до клієнта, то відновити файл з даних не буде складною задачею. Істинне потокове передавання може бути адаптивним. Це означає, що швидкість передавання буде автоматично змінюватись в залежності від умов передавання.

Пряму трансляцію можна забезпечити за допомогою різних протоколів передавання даних. У даному випадку стандартний протокол для відеокодера Teradek Streaming використовує у своїй основі UDP-протокол та забезпечує передавання відео на декодери Teradek. Протокол Teradek Streaming – це “Push” з’єднання, через яке відеокодер передає відео та аудіо одержувачу. Адреса назначення може бути як Unicast індивідуальною IP-адресою або як Multicast адресою для забезпечення підключення великої кількості клієнтів до однієї трансляції. З іншої сторони можливе використання RTMP-протоколу, де відеокодер виступає як клієнт медіасервера. Налаштування відеокодера схожі на ті, що використовуються в Adobe Flash Media Live Encoder, де необхідно виділити

URL-сервера (CDN-сервер, порт або посилання) та назву трансляції. MPEG-4 Visual/PART 2 MPEG-4 Visual (Part 2 of ISO/IEC 14496, “Кодування аудіо-візуальних об’єктів”) – це об’ємний документ, що охоплює широкий спектр функціональних можливостей, пов’язаних з кодуванням та представленням візуальної інформації. У стандарті розглядаються наступні типи даних: рухоме відео (прямокутні кадри); відеооб’єкти (довільні області рухомого відео); двовимірні та тривимірні об’єкти сітки (які представляють об’єкти, здатні до деформації); анімовані людські обличчя та тіла; □ статичні текстури (статичні зображення). Стандарт визначає набір інструментів кодування, які є розроблені для представлення вищезазначених типів даних у компресовану (закодовану) форму. Завдяки різноманітному набору інструментів і підтримуваних типів даних стандарт MPEG-4 Visual може застосовуватись в численній кількості задач, що включають у себе:

- застарілі відеосервіси, такі як цифрове телевізійне мовлення, відеоконференції та зберігання відео;

- об’єктні відеопослідовності, в яких відео сцена може складатися з декількох окремих відеооб’єктів, кожен з яких кодується незалежно;

- рендеринг комп’ютерної графіки з використанням двовимірної та тривимірної сітки, що здатна деформуватися та з використанням анімованих людських обличчя та тіл;

- гібридні відеопослідовності, що поєднують реальне (натуральне) відео, статичні нерухомі зображення і комп’ютерну графіку;

- потокowe відео через веб-мережу і мобільні канали; високоякісне редагування і розповсюдження відео для робочого середовища студій. Незважаючи на значний набір інструментів кодування, що визначений стандартом, в основі MPEG-4 Visual лежить доволі простий механізм кодування відеоданих – блоковий відеокодек, який використовує компенсацію руху, за якою слідує дискретне конусне перетворення, квантування та ентропійне кодування.



Синтаксис цього основного кодека (при деяких обмеженнях) ідентичний синтаксису базового H.263. Більшість інших функцій та інструментів, що підтримуються стандартом, отримані шляхом додавання функцій до цього основного ядра кодека, у той час як інструменти для кодування сітки, нерухомі зображення та параметри анімації обличчя та тіл розробляються окремо. H.264 / MPEG-4 Part 10 H.264 має більш вузьку область застосування, ніж MPEG-4 Visual, і призначена в першу чергу для підтримки ефективного та надійного кодування і передавання прямокутних відеокадрів. Його первинною задачею було забезпечити аналогічну функціональність більш ранніх стандартів, таких як H.263+ та MPEG-4 Visual (Simple Profile), але з суттєво покращеними характеристиками стиснення і покращеною підтримкою надійності передавання. До цільових галузей застосування стандарту відносяться: двосторонній відеозв'язок (відеоконференція чи відеотелефонія), кодування для ширококомовних і високоякісних відео– та відеопотокових мереж[25]. Поле (черезрядкового відео) або кадр (черезрядкового чи прогресивного відео) кодується, щоб створити кодоване зображення. Кодований кадр має свій номер, який сигналізується в бітовому потоці. Цей процес не обов'язково пов'язаний з порядком декодування, і кожне кодоване поле має пов'язаний лічильник порядку кадрів, який визначає порядок декодування полей. Попередньо кодовані зображення (опорні зображення) можуть застосовуватись для міжкадрового передбачення наступних кодованих зображень.

Кодоване зображення складається з множини макроблоків, кожен з яких містить 16 x 16 зразків яскравості і пов'язаних з ними зразків кольоровості (8 x 8 Cb та 8 x 8 зразків Cr в поточному стандарті). І-макроблоки передбачуються за допомогою інтра-передбачення з декодованих виборок в поточному кадрі. Прогнозування формується або для повного макроблоку, або для кожного блока 4 x 4 зразків яскравості (і пов'язаних з ними зразків кольоровості). Рмакроблоки передбачуються, використовуючи внутрішнє прогнозування із зразкового

зображення. Кодований макроблок може бути поділений на частини (блоки) розміром 16 x 16, 16 x 8 чи 8 x 8 зразків яскравості та кольоровості. Якщо обрано блок розміру 8 x 8, кожен субблок може у подальшому бути поділений на блоки 8 x 8, 8 x 4, 4x 8 чи 4 x 4.

Якщо отримувач не може прийняти потік з високою швидкістю передавання даних, то на стороні відправки швидкість передавання буде знижена і якість відеопотоку буде знижена. Це може бути здійснено шляхом змін у потоці або шляхом переключення клієнта на інший потік, можливо від іншого сервера. Потокове передавання відео можна розділити на дві категорії: за вимогою та в режимі реального часу. При потоковому передаванні відео за вимогою клієнт запитує аудіозапис чи відео та отримує його; зазвичай ніхто інший не отримує той самий файл в один й той самий час. У випадку передавання у режимі реального часу відправник визначає, що відправляти, й отримувач програє запис чи відео так, як воно відправлено, з невеликою та послідовною затримкою. У кожній категорії є свої складності. З потоковим передаванням за вимогою служба має відкривати файли по мірі запиту на них і створювати потоки, які йдуть до кожного клієнта. Якщо навантаження на систему є великим, їй, можливо, прийдеться маніпулювати великої кількістю окремих потоків. Такий потік може відставати й глядачі будуть вимушені чекати. Це дратує клієнта, але є прийнятним, якщо затримка є не досить довгою. При потоковому передаванні у режимі реального часу служба зазвичай керує відомою кількістю каналів, але вона має підтримувати їх на швидкості, з якою йде передавання відеопотоку. Якщо ширина каналу не дозволяє передати відеопотік, зазвичай потік пропускається, а не призупиняється. Передавання відео у реальному часі може бути типу точка-точка (один відправник та один одержувач) або широкомовним (один відправник та багато одержувачів)[27] VIOP-конференція є прикладом двонаправленої відеотрансляції типу точкатоčka. Сервери для прямих

відеотрансляцій зазвичай підтримують більше ніж один протокол, відмовляючись від використання альтернатив, якщо перший вибір не працює.

Потокове передавання включає протоколи на декількох різних рівнях моделі OSI. Нижчі рівні (фізичний, канальний та мережевий) зазвичай приймаються як дані. Протоколи прямого передавання даних включають: – транспортний рівень, що відповідає за отримання даних від одного кінця до іншого; – сесійний рівень, який організує процес потокового передавання в постійні блоки, такі як фільми та трансляції; – рівень представлення, який керує мостом між інформацією, як її бачить додаток та інформацією, яка передається через мережу Інтернет; – прикладний рівень, на якому визначається як додаток обмінюється даними з мережею. Найбільш активно використовується транспортний протокол TCP. Він призначений для забезпечення надійного передавання даних. Це означає, якщо пакет не був отриманий, протокол вживає заходів по його відновленню. Надійність – це безумовно дуже важлива характеристика для потокового передавання, але завдяки цьому ми жертвуємо своєчасним отриманням потоку клієнтом. Потокове передавання у режимі реального часу цінує як раз своєчасне отримання контенту, тому часто використовує протокол UDP (User Datagram Protocol). UDP продовжує надсилати пакети, навіть якщо вони не були отримані клієнтом і не прикладає додаткових зусиль для повторної відправки пакетів. Деякі брандмауери можуть блокувати UDP-потоки, тому що вони налашовані тільки для TCP-зв'язку[27]. Підтримка правильного протокола для потокового передавання не обов'язково означає, що ПЗ буде програвати прийнятий потік. Необхідно мати ПЗ, яке підтримує як і протокол для потокового передавання, так і відповідне кодування.

До протоколів, які забезпечують передавання поточкових відеоданих, відносять протоколи RTSP, RTMP, HTTP, HLS та інші. Транспорт, або транспортний контейнер описує, як стиснуте відео упаковують в байти для передавання від одного учасника сесії до іншого через певний протокол

(наприклад, контейнери MPEG-TS, RTMP, RTP). Сімейство RTP Real Time Transport Protocol (RTP) існує вже давно і часто використовується саме для потокового передавання даних. Цей протокол визначений стандартом IETF RFC 3550. RTP побудований на UDP і розроблений спеціально для передавання даних у реальному часі. Дуже рідко RTP використовує TCP для передавання даних. Хоча він знаходиться поверх UDP (або TCP), він досі вважається частиною транспортного рівня. RTP тісно пов'язаний з RTCP (Real Time Control Protocol), який працює на сеансному рівні. Основна функція RTCP в забезпеченні зворотнього зв'язку по якості розподілення даних, дозволяючи коректувати швидкість передавання даних. Деякі протоколи зазвичай використовуються з RTP, але не пов'язані з ним. Протокол потокового передавання у реальному часі RTSP (Real Time Streaming Protocol), який визначено в стандарті IETF RFC 2326, є протоколом рівня представлення, який описується як “мережевий віддалений протокол”. Він деяким чином нагадує HTTP-протокол і містить запити на запуск таких дій як програвання, призупинення та запис. Стек протоколів RTP, RTCP та RTSP іноді згадується в джерелах як “RTSP”. Протоколи RTP, RTCP та RTSP працюють на різних портах. Зазвичай, коли RTP займає порт N, RTCP буде працювати на порту N+1. RTP-сесія може містити декілька потоків, які мають об'єднуватись на стороні одержувача; наприклад, аудіо та відеоконтент можуть передаватись по різних каналах. URL-адреси UDP широко не підтримуються браузерами, тому для здійснення потокового передавання RTP/UDP на веб-сайт необхідно використовувати сторонній модуль. Одним з найпоширеніших є Adobe Flash.

RTP також використовується такими медіапрогравачами як RealPlayer, Windows Media Player та QuickTime Player. На Android та iOS пристроях не підтримуються RTP-сумісні програвачі. Але існують сторонні додатки, в тому числі RealPlayer для Android. RTMP Real Time Messaging Protocol (RTMP) – протокол обміну повідомленнями у реальному часі. Це є пропрієтарний протокол,

який використовується зазвичай Adobe Flash, але також реалізується деяким стороннім ПЗ. Цей протокол використовується поверх TCP, хоча це не є обов'язковою вимогою. RTMP працює через сеансовий рівень. Популярність використання протоколу є прямим результатом поширення Flash, і вона буде знижатись по мірі зменшення використання Flash-плеєрів. Наприклад, iOS не має підтримки RTMP та Flash, тому iPhone, iPod та iPad не будуть приймати RTMP-потоки. Хоча Flash зазвичай асоціюється з власними фаловими форматами, RTMP працює з усіма мультимедійними форматами. Протокол може бути тунельований через HTTP (так званий RTMPT), що дозволяє використовувати його за брандмауєрами, де RTMP-потоки заблоковані. Іншими варіантами для обходу Firewall є RTMPE (легке шифрування), RTMPTE (тунелювання та легке шифрування) і RTMPS (шифрування по SSL). HLS Новою тенденцією в потоковому передаванні даних є використання HTTP-протокола з протоколами, які підтримують адаптивні бітрейти. Таке поєднання в теорії не є вдалим, тому що HTTP у використанні з TCP/IP призначений для надійної доставки даних, а не для підтримання постійного потоку, але з розповсюдженням високошвидкісних з'єднань ця проблема нівелюється. Протокол розроблений компанією Apple і не є розповсюдженим за межами продуктів компанії. Існують сайти, за допомогою яких можна визначити чи підтримує той чи інший браузер протокол HLS. HTTP Live

Streaming – це H.264-відео- аас- або mp3-аудіопотоки, запаковані в транспорт MPEG-TS [29]. Потік розбивають на сегменти і передають через HTTP-протокол. Існує два режими роботи HLS – за запитом і режим трансляції наживо. У першому випадку плей-лист містить посилання на всі фрагменти від першого до останнього. У другому – плей-лист містить тільки посилання на останні кілька фрагментів. Передбачена підтримка адаптивного бітрейту. Недоліком технології є велика затримка під час з'єднання через розбиття потоку на файли тривалістю 10 с., що є неприйнятним для трансляції в реальному часі. Dynamic Streaming over

HTTP (DASH) MPEG DASH (Dynamic Adaptive Streaming over HTTP) - це проект стандарту ISO (ISO/IEC 23009-1) для адаптивної потокової відеотрансляції через HTTP. В майбутньому він має замінити існуючі технології потокової трансляції, такі як Microsoft Smooth Streaming, Adobe Dynamic Streaming і Apple HLS. Стандарт має аналогічну проблему, як і в HLS – суттєву затримку в часі через розбиття потоку на велику кількість файлів. HTML5 Хоча специфікація HTML5 не є протоколом передавання даних, буде доречно згадати її. HTML5 представляє теги <audio> та <video>, властивості DOM, які дозволяють JavaScript керувати вмістом цих тегів. Це є тільки протокол прикладного рівня без визначення нижчих рівнів. Реалізації HTML5 можуть вказувати формати, які обробляються цією специфікацією. Припускається, що сервер буде поступово завантажувати контент, і він буде завантажувати його повністю, навіть якщо програвання буде призупинено, до тих пір, як браузер повністю не видалить цей елемент[29].

### **3.3 Аналіз ефективності різних стандартів стиснення відео**

В останні кілька десятиліть цифрові технології стиснення відео стали необхідністю для створення, зв'язку та споживання візуальної інформації. Відео складається з великої кількості нерухомих зображень, і оскільки зображення приймаються за порівняно короткий час, обидва сусідні зображення схожі. Відомо, що існує вагома кореляція між сусідніми зображеннями або кадрами. Ця кореляція в напрямку часу називається міжкадровою кореляцією. Цю міжкадрову кореляцію можна експлуатувати та ефективно зменшувати, щоб досягти стиснення без втрат якості

Якість відео – це показник, який використовується для вимірювання деградації відео порівняно з оригінальним відео[30]. Якість відео можна оцінити

або за допомогою математичних моделей (об'єктивних), або запитати користувачів за їх оцінкою (суб'єктивно)

Об'єктивні моделі розроблені на основі наявності інформації про вихідний сигнал.

•**Повні довідкові методи (ПДМ):** метод FR – це найбільш точний метод, що вимагає більших обчислювальних зусилля. У цьому методі кожен отриманий піксель порівнюється з відповідним піксель оригінального відео, і нічого не відомо про процес кодування або передачі, що здійснювався.

•**Зниження стандартних методів (ЗСМ):** у ЗС методі не потрібна наявність повного вихідного відео. Зазвичай витягуються особливості отриманих та вихідних відеофільмів і порівнюються, щоб отримати оцінку якості. Цей метод може бути використаний, коли частина оригіналу відео не доступна .

•**Безреєстраційні методи (БМ):** метод оцінки якості відео без наявності вихідного сигналу, цей метод характеризується найменшою точністю.

Щоб розпочати стиснення відео, отримано статичну версію ffmpeg . У Payton був написаний набір кодів для кодеків , які будуть використовуватися, та його параметри, включаючи аудіо, відео бітрейт та розмір кадру. Для створення локалізованого сервера, де файли можуть бути завантажені та перетворені було використано web-середовище Tomcat. Головна увага приділяється порівнянню різних типів кодерів на основі необхідної швидкості передачі даних, якості відео та розміру вихідного файлу.

Для дослідження був обраний файл .mp4 тривалістю 00:02:08 з 3 різними форматами файлів, їх розмір вказаний у табл 1. Під час компресії відео розмір файлу практично не брався до уваги і системі було дозволено самостійно обрати налаштування кодека за замовчуванням.

У табл 2 порівняльне дослідження було проведено ще раз, встановлене значення бітової швидкості 320 кбіт / с для всіх кодеків, а зміна розміру вихідного файлу за якістю відео. Було встановлено, що для певного кодека тип розміру

вихідного файлу майже залишається постійним незалежно від розміру вхідних даних. З табл 2 видно, що помітне зменшення розміру вихідного файлу відбулося тільки для кодеку типу H.264 ,але при цьому якість вихідного відео залишилась задовільною .

**Таблиця 1**– Порівняння якості відео при використанні стандартних кодеків стиснення(бітрейт за замовчуванням)

		Кодеки					
		Mpeg2 video		Mpeg4		H.264	
Вхідні данні	Розмір	Розмір	Якість	Розмір	Якість	Розмір	Якість
.mp4	9.59 Мб	7.87 Мб	середня	6.34 Мб	середня	11.8 Мб	Найвища
.mkv	13.5 Мб	5.18 Мб	середня	5.18 Мб	середня	6.7 Мб	Найвища
.3gp	5.73 Мб	3.72 Мб	середня	3.72 Мб	середня	2.58 Мб	Задовільна



**Таблиця 2** – Порівняння якості відео при використанні стандартних кодеків стиснення (константа бітрейту 320 кбіт/с)

		Кодеки					
		Mpeg2 video		Mpeg4		H.264	
Вхідні данні	Розмір	Розмір	Якість	Розмір	Якість	Розмір	Якість
.mp4	9.59 Мб	13.1 Мб	середня	6.94 Мб	середня	6.91 Мб	висока
.mkv	13.5 Мб	12.9 Мб	середня	6.90 Мб	середня	6.91 Мб	висока
.3gp	5.73 Мб	12.0 Мб	середня	6.99 Мб	середня	6.87 Мб	Задовільна

### **3.4 Низькочастотний алгоритм відеокompresії для збереження відеоматеріалів дистанційного навчання**

Завдяки швидкому збільшенню роздільної здатності зображення в сучасних відеосистемах смуга пропускної здатності пам'яті та споживаної потужності виникла як критична проблема кодування відео. Не зважаючи на багато вбудованих алгоритмів стиснення, запропонованих для подолання цих проблем, належним чином вони поки не виправдовують себе з достатньо високою пропускною спроможністю та без втрати якості.

Оцінка руху (ОР) є потужним інструментом для підвищення ефективності кодування в таких стандартах кодування відео як MPEG2 та H.264 / AVC. Алгоритм відповідності блоків (АВБ) використовується в процесі ОР. АВБ, який знаходить найкращий відповідний блок, порівнюючи поточний блок з усіма блоками-кандидатами в системі відліку, вимагає великої кількості даних довідкової системи. Внаслідок цього частий доступ до зовнішньої пам'яті дає

дефіцит пропускної здатності, а також надмірне споживання електроенергії через перемикання на порти введення-виведення в пам'яті та шинної системі. Це стає більш серйозною проблемою по мірі зростанням якості зображення[32].

Більшість алгоритмів ЄК використовують просторову кореляцію між сусідніми пікселями у початкових джерелах відео. Щоб спостерігати міжпіксельну просторову кореляцію в початкових джерелах відео, ми витягли статистичні характеристики з п'ятнадцяти 1080p (1920x1080) відеопослідовностей понад 200 кадрів на кожен. Статистичні ознаки показані на рис.3.3, де ось x – різниця значень пікселів між прогнозованим пікселем та оригінальним пікселем. Існує чотири різні способи прогнозування: це передбачено середнім значенням лівого і правого пікселів (горизонтальне середнє прогнозування, ГСП), яке передбачається середнім значенням верхнього та нижнього пікселів (середнє значення прогнозу, СЗП), передбачено значенням лівих пікселів (горизонтальний прямий прогноз, ГПП) і передбачається значенням верхнього пікселя (вертикальне пряме прогнозування, ВПП). Середнє передбачення та пряме прогнозування визначаються відповідно наступним чином:

$$P'_{i,sp} = \frac{(P_{i-1} + P_{i+1} + 1)}{2}$$

$$P'_{i,dir} = P_{i-1}$$

Де  $P'_i$  є прогнозованим значенням поточного пікселя, а  $P_{i-1}$  - значення лівого (верхнього) сусіднього поточного пікселя для прогнозів у горизонтальному (вертикальному) напрямку. На рис.3.3 показано, що просторова кореляція у горизонтальному напрямку сильніше, ніж у вертикальному напрямку. Зокрема, понад 90% відмінностей, тобто прогнозування мінус оригінальне значень пікселів, концентруються між -3 та 3, коли попередження виконується ГСП. В інших методах передбачення більшість відмінностей аналогічно сконцентроване поблизу нуля. Ієрархічне прогнозування та групування:

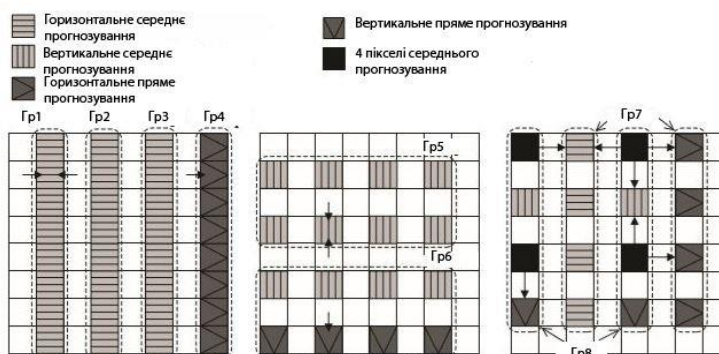


Рисунок 3.3- Схема прогнозування значень поточних пікселів у горизонтальному та вертикальному напрямках

Відповідно до теореми кодування джерел Шеннона, кодери зі змінною довжиною краще, ніж кодери з фіксованою довжиною, для кодування символів, коли щільність імовірності символів є сильно неоднорідною, наприклад, концентрується близько нуля, як на ГСП дає найвищу точність прогнозування серед усіх методів прогнозування, тобто більшість розбіжностей зосереджені в нулі. Іншими словами, коли пікселі в заданому блоці передбачені ГСП[33], більш високу ефективність кодування можна отримати, використовуючи кодування зі змінною довжиною подібно Хаффману, у порівнянні з кодуванням з фіксованою довжиною. Запропонований метод дворівневого прогнозування для корпусу  $8 \times 8$  блоків рівня 1- прогнозований піксель із сусідніх пікселів, тоді як прогноз рівня 2 використовує двопіксельні відстані для прогнозування пікселя. Після прогнозу залишки, отримані шляхом віднімання прогнозованого значення пікселя від значення початкового значення пікселя, класифікуються у вісім груп

**Таблиця 3–** Групи залишків віднімання прогнозованих значень від початкових

Гр <sub>1</sub>	000	(0 біт)
Гр <sub>2</sub>	001	(8 біт)
Гр <sub>3</sub>	011	(24 біт)
Гр <sub>4</sub>	100	(32 біт)
Гр <sub>5</sub>	100	(32 біт)
Гр <sub>6</sub>	100	(32 біт)
Гр <sub>7</sub>	101	(40 біт)
Гр <sub>8</sub>	110	(48 біт)

### **Висновки до розділу**

Є деякі очевидні переваги використання власного рішення для кодування відео. Наприклад, внутрішнє рішення дає більший контроль над всіма робочими відеороликами, ніж у сторонній службі кодування з використанням відео-хмар. Управління всією технологією кодуванням відео надає вам можливість керувати та кодувати великі відео файли, що вимагають великої пропускної здатності. Редагування та переміщення важких відеофайлів в системі кодування для хмар іноді може тривати довше, якщо ви працюєте з компанією, яка працює на хмарній основі. Однак створення власної системи кодування призводить до багатьох обмежень. Наприклад, налаштування та підтримка програмного та апаратного забезпечення для кодування відео вимагатиме величезної серверної бази. Потрібні висококваліфіковані ІТ-фахівці, щоб керувати всіма завданнями кодування відео. Вартість встановлення, підтримання та оновлення внутрішньої системи кодування відео, а також наймання кваліфікованих ІТ-персоналу в більшості випадків перевищує використання служби кодування відео-хмар третьої сторони.

У даній роботі ми пропонуємо алгоритм вбудованого стиснення без втрат з високим ступенем стиснення. Апаратна архітектура запропонованого алгоритму має дуже низьку затримку. Завдяки середньому або прямому прогнозуванню та методу пакування з усіченими бітами понад 60% даних стискаються без погіршення якості та бітрейту. Запропонована архітектура має лише 30 запізнених циклів.

Автор зосередився на порівнянні різних стандартів кодування на основі необхідної швидкості передачі, розмірі та якості відео. Результати експериментів показують, що H.264 забезпечує найкращі результати у порівнянні з іншими стандартами кодування, хороша якість при значному зниженні бітної швидкості до 50 %. Що являє собою оптимальний кодек для технічного забезпечення сегменту дистанційного навчання.

## 4 СТАРТАП-ПРОЕКТ

### 4.1 Опис ідеї проекту

Платформи дистанційного навчання поділяються на 2 типи- ті що встановлюються та адаптуються під середовище компанії– внутрішня платформа дистанційного навчання, та ті, що зберігаються у просторі Інтернету та являються – платформами загального користування. У данній роботі були розглянуті та проаналізовані переваги та недоліка обох типів цих платформ. Відштовхуючись від цієї інформації автор пропонує створити власну платформу дистанційного навчання для кафедр Національного Технічного Університету України «Київський Політехнічний Інститут » імені Ігоря Сікорського.

Основна ідея полягає в використуванні хмарного середовища Amazone Cloud. Ряд перевах данного хмарного сховища задовільняє усі потреби, що можуть виникнути в ході реалізації. За для технічного забезпечення відеоконтенту– як основне джерело подання інформації його потрібно обробити, аби використувати якого менше платних ресурсів хмарноо сховища. У Додатку Б – у середовищі Rayton було розроблено ефективний алгоритм стиснення за для зменшення об’ємів відеоінформації,у Додатку В представлений список класів для технічної підтримки системи дистанційного навчання для рекомендованих форматів відео, а у Дадотку Г представлений оптимізаційний скрипт для взаємії відеоконтенту з хмарним середовищем Amazone Cloud. Економічні показники рохраховані нижче.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Amazon Cloud – це простий і зрозумілий хмарний сервіс для онлайн зберігання відеоконтенту.	1. Онлайн створення і обробки відео контенту.	Дешевизна
	2. Освітнє середовище.	Доступна бібліотека до відеолекцій
	3. Дистанційне навчання.	Не має вимог до технічних параметрів до ПК

Опис до таблиці 4.2:

W – слабка сторона;

N – нейтральна сторона;

S – сильна сторона.

Таблиця 4.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W	N	S
		Amazon Cloud	<u>ICloud</u>	<u>Moodle</u>			
1	Об'єкти застосування	Домашнє та корпоративне використання	Домашнє	Корпоративне			+
2	Контакт-центр	+	+	+		+	

3	Вартість пристрою	Умовно безкоштовна, платний акант – 15 ГБт	Умовно безкоштовна, платний акант – 10ГБ	За 1 курс залежності від об'єму		+	
4	Абон.плата сервісу	0.5\$/міс	3.85\$/міс	350\$		+	

#### 4.2 Технологічний аудит ідеї проекту

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Контакт-центр	«Контакт-центр по запиту» послуга 0-800	наявна	доступна
2	Персональний онлайн сервіс	Програмне забезпечення для ОС: Windows, Android, Mac	необхідно розробити	доступна



### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4 - Попередня характеристика потенційного ринку

п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Динаміка ринку (якісна оцінка)	Зростає
3	Наявність обмежень для входу (вказати характер обмежень)	відсутні
5	Середня норма рентабельності в галузі (або по ринку), %	73%

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Якісне програмне забезпечення, для роботи в Інтернет мережі.	Студенти та викладачі НТУУ «КПІ» ім Ігоря Сікорського і не тільки	Залежно від цільової групи програмне забезпечення може бути безкоштовним, або умовно безкоштовним.	- надійність - доступність - простота - зручність - швидкість

Таблиця 4.6 - Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг та зниження цін
2	Втрата монополії	Втрата рангу єдиного гаранту якості технології, конкретної спроможні готові платформи дистанційної освіти	Якісне та кількісне нарощування інтенсивності

Таблиця 4.7 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Монополія	Інноваційний тип послуг	Стандартизація на високому рівні
2.Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3.Міжгалузєва	Конкуренція з іншими галузями	Необхідність співробітництва в окремих сегментах

4.Товарно-видова	Вбудовані модулі обробки схожі, які реалізовані програмно	За необхідності, використання кодів подібного типу
5.Цінова	Можливість заощадити за допомогою діагностики потреб користувача	Гнучка політика цін
6.Марочна	Кожна діагностика має бути стандартизованою	Отримання монополії надстандартом синхронізації

Таблиця 4.8 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Технологічні постачальники	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним компаніям
Висновки:	Незначна	Можливість виходу на ринок є	Постачальники диктують цінову політику на свої послуги	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови

					від продукта
--	--	--	--	--	-----------------

Таблиця 4.9 - Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурсів
2	Надання персональних сервісних послуг 24/7	Сервісна підтримка апаратної та програмної частини
3	Синхронізованість	Синхронізація з усіма ОС.
4	Спектр застосувань	Використання для ряду потреб користувачів.

Таблиця 4.10 - Порівняльний аналіз сильних та слабких сторін онлайн програми

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні							
			3	2	1	0	1	2	3	
1	Раціональніший ціновий показник	13			+					

2	Надання персональних сервісних послуг 24/7	15			+				
3	Синхронізованість	20	+						
4	Спектр застосувань	17		+					

Таблиця 4.11 - SWOT- аналіз стартап-проекту

Сильні сторони: надання персональних сервісних послуг 24/7, синхронізованість	Слабкі сторони: раціональніший ціновий показник
Можливості: використання для ряду потреб користувачів	Загрози: незацікавленість клієнтів, втрата монополії

#### 4.4 Розроблення ринкової стратегії проекту

Таблиця 4.12 - Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Університети	Середня	Високий	Високий	Середня
Сторонні особи	Високий	Високий	Середній	Середня

Web ресурси (сайти, клуби, соц.мережі)	Висока	Висока	Середній	Середня
Які цільові групи обрано: web ресурси				

Таблиця 4.13 - Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення гаранту якості державного рівня	Встановлення єдиного універсального Веб-додатка	Розробка і випуск власних програм	Стратегія диференціації
2	Дешевизна проекту	Раціональніші витрати на обладнання, та послуги	Маловідомі партнери з надаванням послуг в сфері ІТ.	Стратегія лідерства по витратах

Таблиця 5.14 - Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першоходом» на ринку?	Чи буде компанія шукати нових споживачів, або забирати	Чи буде компанія копіювати основні	Стратегія конкурентної поведінки

		існуючих у конкурентів?	характеристики товару конкурента, і які?	
1	ні	Забирати існуючих та шукати нових	Характеристики апаратної частини	Стратегія виклику лідера

Таблиця 4.15 - Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Синхронізованість	Якість, надійність, сервісність
2	Мінімальні витрати	Стратегія лідерства по витратах	Широкий спектр застосування	Дешевизна, раціональність, тех. підтримка

#### 4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.16 - Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, сервісність	сервісність
2	Дешевизна	Раціональне використання коштів	дешевизна

Таблиця 4.17 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Дешеві послуги, стандартизована якість послуг		
II. Товар у реальному виконанні	Властивості/характеристики:	М/Нм	Вр/Тх /Тл/Е/Ор
	1) Варстість обслуговування, 2) Кількість елементів 3) Строк безвідмовної праці 4) Технологічна собівартість товару	1) М 2) М 3) М 4) М	1)Е 2) Пр 3)Нд 4)Тх
Якість: держстандарт якості, високоякісні технології			



III. Товар із підкрі- пленням	До продажу – програмне забезпечення Після продажу – персональний онлайн сервіс
--	---

Таблиця 5.18 - Визначення меж встановлення ціни

№	Рівень цін на товари замітники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	70-500 у.о./од	5-250 у.о./од	Високий	Н.50 у.о. – В.180 у.о.

Таблиця 4.19 - Концепція маркетингових комунікацій

	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рек-ламного пові-домлення	Концепція рекламного звернення
	Зацікавленість в якісному продукті з раціональним	Мережні ресурси	Синхронізованість з будь-якими ОС	Зацікавити у покращеннях пов'язаних із зростаючою популярністю	Представлення продукції відправною точкою на шляху до безпеки

	використан-ням ресурсів			ю товару та послуг	
	Зацікавленість у великій кількості продукту із дотриманням умов якості	Мережні ресурси	Широкий спектр застосування	Зацікавити у позитивних сторонах	Представлення якісної роботи з клієнтами

## ВИСНОВКИ

Метою дослідження було забезпечити методологічні та технологічні аспекти створення відеоконтенту для сегменту дистанційного навчання. У ході дослідження були виявлено:

### 1. Переваги дистанційного навчання

- Свобода і гнучкість. Можливість навчатися одночасно в різних місцях, на різних курсах не тільки в одному, а й у декількох університетах чи навіть країнах.
- Індивідуальність.

Самостійний вибір студентами темпу навчання, вибору розділів, які варто було б повторити або вилучити.

- Створення власного графіка навчання студентами у звичній для них обстановці і в зручний час.

• Навчання інкогніто (не розголошуючи свого імені) через певні обставини (вік, стан, посаду і т.д.), зареєструвавшись під іншим іменем. • Отримання освіти інвалідами та людьми з різними відхиленнями. • Набуття студентами таких якостей, як самостійність, мобільність і відповідальність. • Навчання більшої кількості людей різних вікових груп порівняно з іншими формами навчання.

- Просте формування віртуальних спільнот: викладачів, студентів тощо завдяки використанню сучасних інтернет технологій, за допомогою яких стає можливим обговорення між викладачами певних проблем, вирішення спільних завдань, обмін досвідом чи інформацією тощо.

### 2. Недоліки дистанційного навчання

- Немає прямого очного спілкування між студентами та викладачем. Подання матеріалу позбавляється емоційного зафарбування, важко створити творчу атмосферу в групі тих, хто навчається.

- Необхідна наявність відповідного технічного та програмного забезпечення, можливість доступу до інформації та використання засобів дистанційного навчання. Користувач повинен бути забезпеченим персональним комп'ютером та доступом в Інтернет.

- Високі вимоги щодо постановки задачі навчання, адміністрування процесу.
- Ключовою проблемою є проблема аутентифікації користувача при перевірці знань. Неможливо точно сказати, хто на іншому кінці дроту.

- Відсутній постійний контроль над тими, хто навчається; відчувається нестача практичної роботи.

- Великі затрати на проектування та створення системи дистанційного навчання, організацію курсів дистанційного навчання і купівлю необхідного обладнання.

- Розроблення курсів дистанційного навчання є дуже трудомістким процесом, створення однієї години інтерактивного мультимедійного матеріалу займає понад 1000 годин роботи професіоналів.

### 3. Основні характеристики та формати відеоінформації, а також технічні аспекти які впливають на створення відеоконтенту.

Отже для сегменту дистанційного навчання доцільним є опиратися на суб'єктивну оцінку якості відео, атже цільова аудиторія планує переглядати відео у наукових та пізнавальних цілях, а не зберігати та використовувати у подальшій постобробці.

Для забезпечення оптимального співвідношення якість/обчислювальна потужності, та для захвату більшої цільової аудиторія доцільним буде створювати початковий відеоматеріал у стандартах PAL: 25р, 50р

NTSC: 29.97р, 30р.

За роздільної здатністю, на сьогоднішній день, сфера зорієнтована на використання таких стандартів цифрового відео як HDTV з розрішенням до 1920 × 1080 при частоті оновлення 50 або 60 Герц з прогресивною розгорткою та

UHDTV або 4K 3840×2160 точок при кадровій частоті 50 і 60 Гц. Для прямих трансляцій, відеоконференцій та вебінарів доцільно використовувати постійну швидкість бітрейту або CBR, а для збереження в якості навчальних матеріалів VBR-змінний бітрейт.

Відео послідовність формується з наступних один за одним кадрів, які змінюються 25 або 30 разів у секунду. Якщо оцифрувати їх з глибиною кольору в 16 біт, то кожен кадр буде займати приблизно 1,2 МВ, а це значить, що для зберігання 1 секунди відео буде потрібно приблизно 30 МВ дискового простору, і на півторагодинний фільм піде близько 150 GB. Це як мінімум уп'ятеро більше ємності жорстких дисків, якими оснащуються сучасні домашні комп'ютери. Для зменшення обсягу файлів, що містять оцифроване відео, на сьогоднішній день використовується MPEG-4- стандартом стиснення мультимедійного контенту. А H.264 за рахунок того, що має суттєво розширені можливості у порівнянні з іншими доступними кодеками, став основним при розробці програмного забезпечення для відеоконференцій і не тільки.

#### 4. Переваги використання власного рішення для кодування відео.

Внутрішнє рішення дає більший контроль над всіма робочими відеороликами, ніж у сторонній службі кодування з використанням відео-хмар. Управління всією технологією кодуванням відео надає можливість керувати та кодувати великі відео файли, що вимагають великої пропускної здатності. Редагування та переміщення важких відеофайлів в системі кодування для хмар іноді може тривати довше, якщо ви працюєте з компанією, яка працює на хмарній основі. Однак створення власної системи кодування призводить до багатьох обмежень. Наприклад, налаштування та підтримка програмного та апаратного забезпечення для кодування відео вимагатиме величезної серверної бази та кваліфікованих фахівців.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Макаренко, К.О. Трапезон, А.М. Чермянін. Основні вимоги до оформлення атестаційних робіт, дипломних та курсових проектів: методичні рекомендації для студентів усіх форм навчання факультету електроніки. – К.: ФЕЛ НТУУ “КПІ”, 2006. – 112 с.
2. Wikipedia.Відеосигнал URL://ru.wikipedia.org ;
3. Порівняння відеокодеків за допомогою метрики PSNR URL: <http://www.ixbt.com/divideo/codex-psnr.shtml> ;
4. V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards: Algorithms and Architectures, Kluwer, 1997– 56с.;
5. Конструювання та технологія виробництва техніки реєстрації інформації. У 3 кн. Кн. 1. Системи та пристрої реєстрації інформації: навчальний посібник / Є.М. Травніков, Г.Г. Власюк, В.В. Пілінський, В.М. Співак, В.Б. Швайченко. – К.: КАФЕДРА, 2013. – 216 с.
6. Цифровая видеозапись URL <https://dic.academic.ru>
7. W. B. Pennebaker, J. L. Mitchell, C. Fogg and D. LeGall, MPEG Digital Video Compression Standard, Chapman & Hall, 1997
8. Обзор сучасних форматів  
URL:[https://itc.ua/articles/cifrovoe\\_video\\_obzor\\_formatov\\_6185/](https://itc.ua/articles/cifrovoe_video_obzor_formatov_6185/)
9. IEEE Transactions on Circuits and Systems for Video Technology, special issue on H.264/AVC, to appear in 2003–5с.
- 10.Цифрове відео: огляд форматів  
URL:<http://itc.ua/article.phtml?ID=6185&IDw=45&pid=20>
- 11.Формати відеозапису URL:[http://www.diapaz.kz/source/tv\\_format.html](http://www.diapaz.kz/source/tv_format.html)

12. W. Li and X. Li, "Design of a Personalized Learning System Based on Intelligent Agent for E-learning", In Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 187-190c.
13. Christine Redecker, Kirsti Ala-Mutka and Yves Punie Learning 2.0: The Impact of Web 2.0 Innovations on Education and Training in Europe
14. Віртуальна реальність URL: <https://www.microsoft.com/en-us/windows/windows-mixed-reality>
15. Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams," RFC3016, Internet Engineering Task Force (IETF), November 2000, 55c.
16. Arbuzov S. S. Tekhnologii podkastinga kak sredstvo aktivizatsii uchebnoy deyatel'nosti studentov pri obuchenii komp'yuternym setyam // Pedagogicheskoe obrazovanie v Rossii. - 2015. - № 7. - S. 30-35.  
URL: <https://cyberleninka.ru/article/n/ispolzovanie-strim-tehnologiy-pri-distantcionnom-obuchenii-it-distsiplinam>
17. А. В. Слепухін, І. Н. Семенової, за основу взята таблиця «Проектування методів навчання і методів використання ІКТ в змішаній моделі навчання 2017, 15 с.
18. SCORM 2004 3rd Edition Content Aggregation Model Version 1.0.  
<http://www.adlnet.gov>, 2006
19. T. Mayes and S. de Freitas, "Review of E-Learning Theories, Frameworks, and Models," Joint Information Systems Committee (JISC) e-Learning Models Desk Study, 2004 c.98
20. C. Feier et al., "Towards Intelligent Web Services: The Web Service Modeling Ontology (WSMO)," Proc. Int'l Conf.

21. Платформа дистанційного навчання Moodle URL: [www.moodle.org](http://www.moodle.org)
22. Робота викладачів і студентів в системі управління навчанням Moodle : навчальний посібник / С. В. Лисоченко, Б. Б. Сусь, О. В. Третяк, В. Ф. Ходаковський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012 с.115
23. Moodle : навчальний посібник / С. В. Лисоченко, Б. Б. Сусь, О. В. Третяк, В. Ф. Ходаковський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 64 с.
24. Хмарні технології URL: <http://wiki.vspu.ru/workroom/tehnol/index>
25. H.264/MPEG-4-AVC URL: [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC)
26. Basics of streaming protocols URL: [www/garymcgath.com/streamingprotocols/html](http://www.garymcgath.com/streamingprotocols/html)
27. Basics of streaming protocols URL: [www/garymcgath.com/streamingprotocols/html](http://www.garymcgath.com/streamingprotocols/html)
28. HTML5 video URL: [https://ru.wikipedia.org/wiki/HTML5\\_video](https://ru.wikipedia.org/wiki/HTML5_video)
29. Connex URL: <https://tv-project.com/connex.html>
30. H.264 Video Compression Standard. New Possibilities with Video Surveillance”, White Paper, Axis Communications
31. Y. Lee and et al, «A New Frame Recompression Algorithm Integrated with H.264 Video Compression,» Circuits and Systems, IEEE International Symposium on, pp. 1621-1624, May 2007.
32. R.R. Osorio and et al, «A Combined Memory Compression And Hierarchical Motion Estimation Architecture For Video Encoding In Embedded Systems,» Digital System Design, EUROMICRO Conference on, с. 269-274, 2006.



**ДОДАТОК А**  
**ABSTRACT**

The promising use of the possibilities of the Internet as a global interactive, remote learning environment is boundless, and the technological potential is very large and not explored to the fullest. The use of new teaching methods completely changes the paradigm of modern education. A new toolkit based on educational video materials, especially accelerated with the emergence of high-speed Internet, which completely changed the principles that underlie the classical (Soviet and early post-Soviet) education. Modern technologies give everyone the opportunity to become an author or co-author of information on the Internet.

The use of modern technologies makes education more interactive, that is, one that engages in participation and provides opportunities for active interaction at all levels. Among the opportunities offered by distance learning - self-education for a long period of time; free networking with like-minded people, collaboration and collaboration between pupils and teachers; impact on the learning process (feedback, feedback and feedback, the ability to share training information); free and large selection of educational materials, programs; courses, collective interaction; creativity and creativity in the media space. promotes communication and collective skills, and the individual's ability to create their own educational content. Usn this provides the opportunity to share the developments with other users of the courses, as well as collectively acquire knowledge. Today it is possible to work with web platforms without knowledge of programming languages - everyone can create a blog, video, virtual community, etc., if desired.

- 67% of Internet usage related to video
- 70% of Internet visitors purposefully visit the network for video viewing
- More than 24 hours of video browsing over 800 million people

And this tendency only grows. The benefits of using video content for distance learning are obvious, most people perceive information visually, because the digital age is simply not enough words and text.

The purpose of the study is to provide methodological and technological aspects of creating video content for the E-learning segment

The object of research is the process of preparing video content for the E-learning segment

Subject of research - video formats and characteristics, processes of adaptation, coding and decoding, storage and transmission of video information, E-learning platforms.

Research methods - theoretical study of the process of video constitution formation for existing distance learning platforms, analysis of components of distance learning and the development of an appropriate algorithm for video information preparation

Scientific novelty of the obtained results: tried technologies that provide the appropriate training for the formation of a video tape for distance learning, analyze the contemporary needs for video quality standards, methods of storage, transmission and processing of video signals.

Practical value of the obtained results. The received data allow to realize the project of organization of own platform of international distance learning and to provide high-quality video content on the basis of this platform.

The aim of the study was to provide methodological and technological aspects of video content creation for the E-learning segment. The study revealed:

1. Advantages of distance learning

- Freedom and flexibility. Ability to study simultaneously in different places, at different courses, not only in one, but also in several universities or even countries.
- Individuality.

Independent student choice of learning pace, selection of chapters that should be repeated or deleted.

- Creating your own training schedule for students in their usual environment and at a convenient time.

- Teaching incognito (without disclosing your name) due to certain circumstances (age, state, position, etc.) by registering under a different name.
- Education for people with disabilities and people with different abnormalities.
- Students gain such qualities as autonomy, mobility and responsibility.
- Teaching more people of different age groups than other forms of learning.

- Simple formation of virtual communities: teachers, students, etc. through the use of modern Internet technologies, through which it becomes possible to discuss between teachers of certain problems, solving common problems, exchange of experience or information, etc.

## 2. Disadvantages of distance learning

- There is no direct face-to-face communication between students and the teacher. Submission of material deprives emotional coloring, it is difficult to create a creative atmosphere in the group of learners.

- Appropriate hardware and software, access to information and the use of distance learning facilities are required. The user must be provided with a personal computer and access to the Internet.

- High requirements for setting training objectives, process administration.
- A key issue is the problem of authenticating the user when checking knowledge. It's impossible to say exactly who on the other end of the wire.

- There is no permanent control students; there is a lack of practical work.

- Large costs for the design and creation of a distance learning system, the organization of distance learning courses and the purchase of the necessary equipment.

- Development of distance learning courses is a very time-consuming process, creating over one hour of interactive multimedia material takes over 1,000 hours of work for professionals.

3. The main characteristics and formats of video information, as well as the technical aspects that affect the creation of video content.

So, for the distance learning segment, it is expedient to rely on a subjective evaluation of the quality of the video, but the target audience is planning to view videos in science and cognitive purposes rather than store and use in further post-processing.

To provide the optimal quality / computing power ratio, and to capture a larger target audience, it will be expedient to create an original video in PAL standards: 25p, 50p

NTSC: 29.97p, 30p.

By today's resolution, the field is focused on the use of digital video standards such as HDTV up to  $1920 \times 1080$  resolution up to 50 or 60 Hz with upgrade scan and UHD TV or 4K  $3840 \times 2160$  pixels at 50 and 60 Hz cadres. For live broadcasts, videoconferences and webinars it is expedient to use constant speed of bitrate or CBR, and to save as training materials VBR-variable bitrate.

The video sequence is formed from successive frames that change 25 or 30 times per second. If you digitize them with a 16-bit color depth, each frame will take about 1.2 MB, which means that about 1 in 30 seconds of video will require about 30 MB of disk space, and about 1.5 GB will take about one and a half hours. This is at least five times the capacity of hard drives that are equipped with modern personal computers. To reduce the size of files containing digital video, today is using MPEG-4 standard for compression of multimedia content. And H.264 due to the fact that it has significantly enhanced capabilities in comparison with other available codecs, became the main part in software development for video conferencing and not only.

4. Advantages of using your own video encoding solution.

The internal solution gives more control over all working videos than the third-party video coding service. Managing all video encoding technology allows you to manage and encode large video files that require high bandwidth. Editing and moving heavy video files in the cloud-coding system can sometimes take longer if you are working with a cloud-based company. However, the creation of its own coding system leads to many limitations. For example, setting up and maintaining software and hardware for video encoding will require a huge server base and skilled professionals.

#### 5. Application of cloud technologies

An example of using cloud technologies in education can be called electronic diaries and magazines, personal rooms for students and teachers, interactive reception and more. These are thematic forums where students can exchange information. This is the search for information where students can solve certain learning tasks, even in the absence of a teacher or under his guidance. You can use this to:

- computer programs
- electronic textbooks
- simulators
- diagnostic, test and training systems
- Applied and software tools
- laboratory complexes
- systems based on multimedia technology
- telecommunication systems (eg, e-mail, teleconferencing)
- electronic libraries and more.

**ДОДАТОК Б**  
**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВІДЕОКОМПРЕСІНГУ**

```

import sys
import Tkinter as tk
import tkFileDialog
import ttk
import os
import time
import subprocess

WINDOW_TITLE = "VIDEO EZY 2: Electric Boogaloo"

class Application(tk.Frame):

    DEBUG = False

    def __init__(self, master=None):
        tk.Frame.__init__(self, master)
        self.pack()
        self.create_widgets(master)

        self.list_of_files = []

    def create_widgets(self, master):
        # box listing selected files
        self.selected_files_string = tk.StringVar()
        self.selectedTxt = tk.Entry(self, textvariable =
self.selected_files_string, width=50)

        # button for browsing for files
        self.browseBtn = tk.Button(self, text="Browse...",
command=self.pick_files)

        # checkbox for if keep the old files or not
        self.keep_old = tk.IntVar()
        self.keep_old.set(1)
        self.keepChk = tk.Checkbutton(self, text="Keep old files",
variable=self.keep_old)

        # convert button
        self.convertBtn = tk.Button(self, text="CONVERT!",
command=self.start)

        # progress label
        self.progress = ttk.Progressbar(self, orient="horizontal",
mode="determinate")

        # placing all the items
        self.selectedTxt.grid(row=0, column=0, columnspan=5)
        self.browseBtn .grid(row=0, column=5, columnspan=1)
        self.keepChk .grid(row=1, column=0, columnspan=1)
        self.convertBtn .grid(row=1, column=1, columnspan=5,
sticky=tk.W+tk.E)
        self.progress .grid(row=2, column=0, columnspan=6,
sticky=tk.W+tk.E)

```



```

def pick_files(self):
    # ask for files
    selected_files = tkFileDialog.askopenfilenames(parent=self,
title="Choose a file")

    # turn the list of files into a string
    string_of_files = ", ".join([str(file).replace("/", "\\") for file in
selected_files if str(file) != ""])

    # set the text in the selected_files text box
    self.selected_files_string.set(string_of_files)

    self.progress["value"] = 0

def start(self):
    if self.selected_files_string.get() == "":
        print("No files selected")
        return

    # disable the buttons and checkbox so you cant change anything
    self.selectedTxt.configure(state=tk.DISABLED)
    self.browseBtn.configure(state=tk.DISABLED)
    self.keepChk.configure(state=tk.DISABLED)
    self.convertBtn.configure(state=tk.DISABLED)

    # what number are we up to?
    self.count = 0

    # all the files to convert
    self.list_of_files = self.selected_files_string.get().split(", ")

    self.progress["value"] = 0
    self.file_count = len(self.list_of_files)
    self.progress["maximum"] = len(self.list_of_files)

    root.title(WINDOW_TITLE + " [{} / {}]".format(self.count,
self.file_count))

    self.after(100, self.convert)

def convert(self):
    filename = self.list_of_files.pop(0)

    path, f = filename.rsplit("\\", 1)
    filename_new = os.path.join(path, "small-") + f.rsplit(".",
1)[0] + ".mp4"

    self.execute('ffmpeg -y -i "{}" -strict -2 -vf scale=-1:720
"{}"'.format(filename, filename_new))

    if self.keep_old.get() == 0:
        self.execute('del "{}"'.format(filename))

```

```

self.count += 1
self.progress["value"] = self.count

root.title(WINDOW_TITLE + " [{} / {}]".format(self.count,
self.file_count))

if self.count < self.file_count:
    self.after(100, self.convert)

else:
    # after completion
    self.selected_files_string.set("")
    self.selectedTxt.configure(state=tk.NORMAL)
    self.browseBtn.configure(state=tk.NORMAL)
    self.keepChk.configure(state=tk.NORMAL)
    self.convertBtn.configure(state=tk.NORMAL)
    root.title(WINDOW_TITLE)

def execute(self, command):
    if self.DEBUG:
        print(command)
    else:
        CREATE_NO_WINDOW = 0x08000000
        DETACHED_PROCESS = 0x00000008

        #os.system(command)
        output = subprocess.call(command,
creationflags=CREATE_NO_WINDOW, shell=True)

# create root and configure
root = tk.Tk()
root.title(WINDOW_TITLE)
root.resizable(0,0)

# start up the application
app = Application(master=root)
app.mainloop()

```

**ДОДАТОК В**  
**СПИСОК КЛАСІВ КОДЕКІВ, ЩО ПІДТРИМУЮТЬСЯ**

```

class BaseCodec(object):
    """
    Base audio/video codec class.
    """

    encoder_options = {}
    codec_name = None
    ffmpeg_codec_name = None

    def parse_options(self, opt):
        if 'codec' not in opt or opt['codec'] != self.codec_name:
            raise ValueError('invalid codec name')
        return None

    def _codec_specific_parse_options(self, safe):
        return safe

    def _codec_specific_produce_ffmpeg_list(self, safe):
        return []

    def safe_options(self, opts):
        safe = {}

        # Only copy options that are expected and of correct type
        # (and do typecasting on them)
        for k, v in opts.items():
            if k in self.encoder_options:
                typ = self.encoder_options[k]
                try:
                    safe[k] = typ(v)
                except:
                    pass

        return safe

class AudioCodec(BaseCodec):
    """
    Base audio codec class handles general audio options. Possible
    parameters are:
    * codec (string) - audio codec name
    * channels (integer) - number of audio channels
    * bitrate (integer) - stream bitrate
    * samplerate (integer) - sample rate (frequency)
    Supported audio codecs are: null (no audio), copy (copy from
    original), vorbis, aac, mp3, mp2
    """

    encoder_options = {
        'codec': str,
        'channels': int,
        'bitrate': int,
        'samplerate': int
    }

    def parse_options(self, opt):
        super(AudioCodec, self).parse_options(opt)

```

```

safe = self.safe_options(opt)

if 'channels' in safe:
    c = safe['channels']
    if c < 1 or c > 12:
        del safe['channels']

if 'bitrate' in safe:
    br = safe['bitrate']
    if br < 8 or br > 512:
        del safe['bitrate']

if 'samplerate' in safe:
    f = safe['samplerate']
    if f < 1000 or f > 50000:
        del safe['samplerate']

safe = self._codec_specific_parse_options(safe)

optlist = ['-acodec', self.ffmpeg_codec_name]
if 'channels' in safe:
    optlist.extend(['-ac', str(safe['channels'])])
if 'bitrate' in safe:
    optlist.extend(['-ab', str(safe['bitrate']) + 'k'])
if 'samplerate' in safe:
    optlist.extend(['-ar', str(safe['samplerate'])])

optlist.extend(self._codec_specific_produce_ffmpeg_list(safe))
return optlist

```

```
class SubtitleCodec(BaseCodec):
```

```
    """
```

```
    Base subtitle codec class handles general subtitle options. Possible
    parameters are:
```

```
    * codec (string) - subtitle codec name (mov_text, subrib, ssa only
    supported currently)
```

```
    * language (string) - language of subtitle stream (3 char code)
```

```
    * forced (int) - force subtitles (1 true, 0 false)
```

```
    * default (int) - default subtitles (1 true, 0 false)
```

```
    Supported subtitle codecs are: null (no subtitle), mov_text
```

```
    """
```

```
    encoder_options = {
```

```
        'codec': str,
        'language': str,
        'forced': int,
        'default': int
    }
```

```
    def parse_options(self, opt):
```

```
        super(SubtitleCodec, self).parse_options(opt)
```

```
        safe = self.safe_options(opt)
```

```
        if 'forced' in safe:
```

```
            f = safe['forced']
```

```

        if f < 0 or f > 1:
            del safe['forced']

    if 'default' in safe:
        d = safe['default']
        if d < 0 or d > 1:
            del safe['default']

    if 'language' in safe:
        l = safe['language']
        if len(l) > 3:
            del safe['language']

    safe = self._codec_specific_parse_options(safe)

    optlist = ['-scodec', self.ffmpeg_codec_name]

    optlist.extend(self._codec_specific_produce_ffmpeg_list(safe))
    return optlist

class VideoCodec(BaseCodec):
    """
    Base video codec class handles general video options. Possible
    parameters are:
    * codec (string) - video codec name
    * bitrate (string) - stream bitrate
    * fps (integer) - frames per second
    * width (integer) - video width
    * height (integer) - video height
    * mode (string) - aspect preserval mode; one of:
        * stretch (default) - don't preserve aspect
        * crop - crop extra w/h
        * pad - pad with black bars
    * src_width (int) - source width
    * src_height (int) - source height
    Aspect preserval mode is only used if both source
    and both destination sizes are specified. If source
    dimensions are not specified, aspect settings are ignored.
    If source dimensions are specified, and only one
    of the destination dimensions is specified, the other one
    is calculated to preserve the aspect ratio.
    Supported video codecs are: null (no video), copy (copy directly
    from the source), Theora, H.264/AVC, DivX, VP8, H.263, Flv,
    MPEG-1, MPEG-2.
    """

    encoder_options = {
        'codec': str,
        'bitrate': int,
        'fps': int,
        'width': int,
        'height': int,
        'mode': str,
        'src_width': int,
        'src_height': int,
    }
}

```

```

def _aspect_corrections(self, sw, sh, w, h, mode):
    # If we don't have source info, we don't try to calculate
    # aspect corrections
    if not sw or not sh:
        return w, h, None

    # Original aspect ratio
    aspect = (1.0 * sw) / (1.0 * sh)

    # If we have only one dimension, we can easily calculate
    # the other to match the source aspect ratio
    if not w and not h:
        return w, h, None
    elif w and not h:
        h = int((1.0 * w) / aspect)
        return w, h, None
    elif h and not w:
        w = int(aspect * h)
        return w, h, None

    # If source and target dimensions are actually the same aspect
    # ratio, we've got nothing to do
    if int(aspect * h) == w:
        return w, h, None

    if mode == 'stretch':
        return w, h, None

    target_aspect = (1.0 * w) / (1.0 * h)

    if mode == 'crop':
        # source is taller, need to crop top/bottom
        if target_aspect > aspect: # target is taller
            h0 = int(w / aspect)
            assert h0 > h, (sw, sh, w, h)
            dh = (h0 - h) / 2
            return w, h0, 'crop=%d:%d:0:%d' % (w, h, dh)
        else: # source is wider, need to crop left/right
            w0 = int(h * aspect)
            assert w0 > w, (sw, sh, w, h)
            dw = (w0 - w) / 2
            return w0, h, 'crop=%d:%d:%d:0' % (w, h, dw)

    if mode == 'pad':
        # target is taller, need to pad top/bottom
        if target_aspect < aspect:
            h1 = int(w / aspect)
            assert h1 < h, (sw, sh, w, h)
            dh = (h - h1) / 2
            return w, h1, 'pad=%d:%d:0:%d' % (w, h, dh) # FIXED
        else: # target is wider, need to pad left/right
            w1 = int(h * aspect)
            assert w1 < w, (sw, sh, w, h)
            dw = (w - w1) / 2
            return w1, h, 'pad=%d:%d:%d:0' % (w, h, dw) # FIXED

```

```

    assert False, mode

def parse_options(self, opt):
    super(VideoCodec, self).parse_options(opt)

    safe = self.safe_options(opt)

    if 'fps' in safe:
        f = safe['fps']
        if f < 1 or f > 120:
            del safe['fps']

    if 'bitrate' in safe:
        br = safe['bitrate']
        if br < 16 or br > 15000:
            del safe['bitrate']

    w = None
    h = None

    if 'width' in safe:
        w = safe['width']
        if w < 16 or w > 4000:
            w = None

    if 'height' in safe:
        h = safe['height']
        if h < 16 or h > 3000:
            h = None

    sw = None
    sh = None

    if 'src_width' in safe and 'src_height' in safe:
        sw = safe['src_width']
        sh = safe['src_height']
        if not sw or not sh:
            sw = None
            sh = None

    mode = 'stretch'
    if 'mode' in safe:
        if safe['mode'] in ['stretch', 'crop', 'pad']:
            mode = safe['mode']

    ow, oh = w, h # FIXED
    w, h, filters = self._aspect_corrections(sw, sh, w, h, mode)

    safe['width'] = w
    safe['height'] = h
    safe['aspect_filters'] = filters

    if w and h:
        safe['aspect'] = '%d:%d' % (w, h)

    safe = self._codec_specific_parse_options(safe)

```



```

w = safe['width']
h = safe['height']
filters = safe['aspect_filters']

optlist = ['-vcodec', self.ffmpeg_codec_name]
if 'fps' in safe:
    optlist.extend(['-r', str(safe['fps'])])
if 'bitrate' in safe:
    optlist.extend(['-vb', str(safe['bitrate']) + 'k']) # FIXED
if w and h:
    optlist.extend(['-s', '%dx%d' % (w, h)])

    if ow and oh:
        optlist.extend(['-aspect', '%d:%d' % (ow, oh)])

if filters:
    optlist.extend(['-vf', filters])

optlist.extend(self._codec_specific_produce_ffmpeg_list(safe))
return optlist

class AudioNullCodec(BaseCodec):
    """
    Null audio codec (no audio).
    """
    codec_name = None

    def parse_options(self, opt):
        return ['-an']

class VideoNullCodec(BaseCodec):
    """
    Null video codec (no video).
    """
    codec_name = None

    def parse_options(self, opt):
        return ['-vn']

class SubtitleNullCodec(BaseCodec):
    """
    Null video codec (no video).
    """
    codec_name = None

    def parse_options(self, opt):
        return ['-sn']

class AudioCopyCodec(BaseCodec):
    """
    Copy audio stream directly from the source.

```

```

"""
codec_name = 'copy'

def parse_options(self, opt):
    return ['-acodec', 'copy']

class VideoCopyCodec(BaseCodec):
    """
    Copy video stream directly from the source.
    """
    codec_name = 'copy'

    def parse_options(self, opt):
        return ['-vcodec', 'copy']

class SubtitleCopyCodec(BaseCodec):
    """
    Copy subtitle stream directly from the source.
    """
    codec_name = 'copy'

    def parse_options(self, opt):
        return ['-scodec', 'copy']

# Audio Codecs
class VorbisCodec(AudioCodec):
    """
    Vorbis audio codec.
    @see http://ffmpeg.org/trac/ffmpeg/wiki/TheoraVorbisEncodingGuide
    """
    codec_name = 'vorbis'
    ffmpeg_codec_name = 'libvorbis'
    encoder_options = AudioCodec.encoder_options.copy()
    encoder_options.update({
        'quality': int, # audio quality. Range is 0-10(highest quality)
        # 3-6 is a good range to try. Default is 3
    })

    def _codec_specific_produce_ffmpeg_list(self, safe):
        optlist = []
        if 'quality' in safe:
            optlist.extend(['-qscale:a', safe['quality']])
        return optlist

class AacCodec(AudioCodec):
    """
    AAC audio codec.
    """
    codec_name = 'aac'
    ffmpeg_codec_name = 'aac'
    aac_experimental_enable = ['-strict', 'experimental']

    def _codec_specific_produce_ffmpeg_list(self, safe):
        return self.aac_experimental_enable

```

```

class FdkAacCodec(AudioCodec):
    """
    AAC audio codec.
    """
    codec_name = 'libfdk_aac'
    ffmpeg_codec_name = 'libfdk_aac'

class Ac3Codec(AudioCodec):
    """
    AC3 audio codec.
    """
    codec_name = 'ac3'
    ffmpeg_codec_name = 'ac3'

class FlacCodec(AudioCodec):
    """
    FLAC audio codec.
    """
    codec_name = 'flac'
    ffmpeg_codec_name = 'flac'

class DtsCodec(AudioCodec):
    """
    DTS audio codec.
    """
    codec_name = 'dts'
    ffmpeg_codec_name = 'dts'

class Mp3Codec(AudioCodec):
    """
    MP3 (MPEG layer 3) audio codec.
    """
    codec_name = 'mp3'
    ffmpeg_codec_name = 'libmp3lame'

class Mp2Codec(AudioCodec):
    """
    MP2 (MPEG layer 2) audio codec.
    """
    codec_name = 'mp2'
    ffmpeg_codec_name = 'mp2'

# Video Codecs
class TheoraCodec(VideoCodec):
    """
    Theora video codec.
    @see http://ffmpeg.org/trac/ffmpeg/wiki/TheoraVorbisEncodingGuide
    """
    codec_name = 'theora'

```

```

ffmpeg_codec_name = 'libtheora'
encoder_options = VideoCodec.encoder_options.copy()
encoder_options.update({
    'quality': int, # audio quality. Range is 0-10(highest quality)
    # 5-7 is a good range to try (default is 200k bitrate)
})

def _codec_specific_produce_ffmpeg_list(self, safe):
    optlist = []
    if 'quality' in safe:
        optlist.extend(['-qscale:v', safe['quality']])
    return optlist

class H264Codec(VideoCodec):
    """
    H.264/AVC video codec.
    @see http://ffmpeg.org/trac/ffmpeg/wiki/x264EncodingGuide
    """
    codec_name = 'h264'
    ffmpeg_codec_name = 'libx264'
    encoder_options = VideoCodec.encoder_options.copy()
    encoder_options.update({
        'preset': str, # common presets are ultrafast, superfast, veryfast,
        # faster, fast, medium(default), slow, slower, veryslow
        'quality': int, # constant rate factor, range:0(lossless)-51(worst)
        # default:23, recommended: 18-28
        # http://mewiki.project357.com/wiki/X264_Settings#profile
        'profile': str, # default: not-set, for valid values see above link
        'tune': str, # default: not-set, for valid values see above link
    })

    def _codec_specific_produce_ffmpeg_list(self, safe):
        optlist = []
        if 'preset' in safe:
            optlist.extend(['-preset', safe['preset']])
        if 'quality' in safe:
            optlist.extend(['-crf', safe['quality']])
        if 'profile' in safe:
            optlist.extend(['-profile', safe['profile']])
        if 'tune' in safe:
            optlist.extend(['-tune', safe['tune']])
        return optlist

class DivxCodec(VideoCodec):
    """
    DivX video codec.
    """
    codec_name = 'divx'
    ffmpeg_codec_name = 'mpeg4'

class Vp8Codec(VideoCodec):
    """
    Google VP8 video codec.
    """

```

```

    codec_name = 'vp8'
    ffmpeg_codec_name = 'libvpx'

class H263Codec(VideoCodec):
    """
    H.263 video codec.
    """
    codec_name = 'h263'
    ffmpeg_codec_name = 'h263'

class FlvCodec(VideoCodec):
    """
    Flash Video codec.
    """
    codec_name = 'flv'
    ffmpeg_codec_name = 'flv'

class MpegCodec(VideoCodec):
    """
    Base MPEG video codec.
    """
    # Workaround for a bug in ffmpeg in which aspect ratio
    # is not correctly preserved, so we have to set it
    # again in vf; take care to put it *before* crop/pad, so
    # it uses the same adjusted dimensions as the codec itself
    # (pad/crop will adjust it further if neccessary)
    def _codec_specific_parse_options(self, safe):
        w = safe['width']
        h = safe['height']

        if w and h:
            filters = safe['aspect_filters']
            tmp = 'aspect=%d:%d' % (w, h)

            if filters is None:
                safe['aspect_filters'] = tmp
            else:
                safe['aspect_filters'] = tmp + ',' + filters

        return safe

class Mpeg1Codec(MpegCodec):
    """
    MPEG-1 video codec.
    """
    codec_name = 'mpeg1'
    ffmpeg_codec_name = 'mpeg1video'

class Mpeg2Codec(MpegCodec):
    """
    MPEG-2 video codec.
    """

```

```

    codec_name = 'mpeg2'
    ffmpeg_codec_name = 'mpeg2video'

# Subtitle Codecs
class MOVTextCodec(SubtitleCodec):
    """
    mov_text subtitle codec.
    """
    codec_name = 'mov_text'
    ffmpeg_codec_name = 'mov_text'

class SSA(SubtitleCodec):
    """
    SSA (SubStation Alpha) subtitle.
    """
    codec_name = 'ass'
    ffmpeg_codec_name = 'ass'

class SubRip(SubtitleCodec):
    """
    SubRip subtitle.
    """
    codec_name = 'subrip'
    ffmpeg_codec_name = 'subrip'

class DVBSub(SubtitleCodec):
    """
    DVB subtitles.
    """
    codec_name = 'dvbsub'
    ffmpeg_codec_name = 'dvbsub'

class DVDSub(SubtitleCodec):
    """
    DVD subtitles.
    """
    codec_name = 'dvdsup'
    ffmpeg_codec_name = 'dvdsup'

audio_codec_list = [
    AudioNullCodec, AudioCopyCodec, VorbisCodec, AacCodec, Mp3Codec,
    Mp2Codec,
    FdkAacCodec, Ac3Codec, DtsCodec, FlacCodec
]

video_codec_list = [
    VideoNullCodec, VideoCopyCodec, TheoraCodec, H264Codec,
    DivxCodec, Vp8Codec, H263Codec, FlvCodec, Mpeg1Codec,
    Mpeg2Codec
]

```

```
subtitle_codec_list = [  
    SubtitleNullCodec, SubtitleCopyCodec, MOVTextCodec, SSA, SubRip, DVDSub,  
    DVBSub  
]
```

**ДОДАТОК Г**  
**СКРИПТ ОПТИМІЗАЦІЇ РОЗМІРУ ВІДЕОКОНТЕНТУ ПРИ**  
**ЗАВАНТАЖЕННІ НА ХМАРНЕ СЕРЕДОВИЩЕ**



```

# encoding: utf-8
import SocketServer
import json
import tqdm
import logging
import os
from pprint import pprint
from sh import ffprobe as _ffprobe, ffmpeg as _ffmpeg
from threading import Thread
from video_optimizer import cast_type, Dict

log = logging.getLogger(__name__)

def ffprobe(*args, **kwargs):
    return Dict(
        cast_type(
            json.loads(
                _ffprobe(
                    '-print_format',
                    'json',
                    *args,
                    **kwargs
                ).stdout
            )
        )
    )

def ffmpeg(*args, **kwargs):
    try:
        result = _ffmpeg(
            _iter=True,
            _err_to_out=True,
            *args, **kwargs
        )

        for line in result:
            print(line.strip())

    except Exception as e:
        print(e.stdout)
        print(e.stderr)
        raise

def create_handler(total):
    class ProgressHandler(SocketServer.StreamRequestHandler):
        bar_format = "{desc}{percentage:3.0f}% {bar} [{elapsed} ETA: {remaining}]"

        def handle(self):
            with tqdm.tqdm(total=total, unit='ms',
                bar_format=self.bar_format) as progressbar:
                old = 0

                while True:

```

```

        line = self.rfile.readline().strip()
        if not line:
            continue

        key, value = line.split("=")

        if key != "out_time_ms":
            continue

        value = cast_type(value)
        delta = value - old
        if delta < 1:
            continue

        progressbar.update(delta)

        old = value

    return ProgressHandler

def file_size_mb(fname):
    return os.stat(fname).st_size / 1024 / 1024.

def convert(input_file, output_file, profile):
    streams = ffprobe('-show_streams', input_file).get('streams')

    log.info("Found %d streams.", len(streams))
    log.debug("\n%s", pformat(streams))

    progress_server = SocketServer.TCPServer(
        ('', 0),
        create_handler(streams[0].duration * 1000000)
    )

    thread = Thread(target=progress_server.serve_forever)
    thread.daemon = True
    _, progress_port = progress_server.server_address

    thread.start()

    args = (
        '-progress',
        'tcp://localhost:{0}'.format(progress_port),
    )

    args += (
        '-loglevel',
        'panic',
        '-i', input_file,
        '-y',
    )

    args += tuple(profile(streams))

    args += (output_file,)

```

```
try:
    ffmpeg(*args)

    input_file_size = file_size_mb(input_file)
    output_file_size = file_size_mb(output_file)
    ratio = float(output_file_size)/float(input_file_size)

    log.info(
        "File optimized:\n\tOriginal size: %02fMb\n\tNew
size: %02fMb\n\tRatio: %03f",
        input_file_size,
        output_file_size,
        ratio,
    )
finally:
    progress_server.server_close()
```