

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається

Т.в.о. завідувача кафедри

_____ Сафонова С.О.

« ____ » _____ 2020 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження методів анотування відеопотоку та розробка
програмного додатку для анотування відео

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 123 “Комп’ютерна інженерія”

Науковий керівник роботи:

(підпис)

О.І.Рязанцев

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

(підпис)

А. М. Гамалій

(ініціали, прізвище)

Група:

КІ-18дм

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний
рівень “магістр”
Спеціальність 123 – “Комп'ютерна інженерія”
(шифр і назва)
Спеціалізація _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри комп'ютерної інженерії
к.т.н., доц. С.О. Сафонова
« _____ » _____ 2020 р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Гамалій Артуру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів анотування відеопотоку та розробка програмного додатку для анотування відео

керівник проекту (роботи) Рязанцев О.І., д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «11» 10 2019 р. № _____

2. Строк подання студентом роботи 15.01.2020

3. Вихідні дані до роботи Матеріали науково-дослідної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Аналіз предметної області та постановка задачі.

2. Дослідження методів анотування відеопотоку.

3. Розробка алгоритму анотування відеопотоку на основі одного з методів.

4. Дослідження результатів розробленого алгоритму анотування відеопотоку.

5. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|-------------------|---------------------|
| | | завдання видав | завдання прийняв |
| Охорона праці | <i>Критська Я.О.</i> | | |
| | | | |
| | | | |

7. Дата видачі завдання 06.09.2019

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи) | Строк виконання етапів проекту (роботи) | Примітка |
|-------|--|---|----------|
| 1 | Отримання завдання до магістерської роботи | <i>10.09.19-15.09.19</i> | |
| 2 | Аналіз завдання, огляд літератури | <i>16.09.19-25.09.19</i> | |
| 3 | Аналіз літератури з досліджуваної проблеми | <i>26.09.19-10.10.19</i> | |
| 4 | Аналіз технічних засобів | <i>11.10.19-20.10.19</i> | |
| 5 | Розробка методу | <i>21.10.19-06.11.19</i> | |
| 6 | Програмна реалізація | <i>17.11.19-06.12.19</i> | |
| 7 | Охорона праці | <i>07.12.19-15.12.19</i> | |
| 8 | Оформлення пояснювальної записки | <i>16.12.19-28.12.19</i> | |
| 9 | Підготовка презентації та доповіді | <i>03.01.20-13.01.20</i> | |

Студент _____

(підпис)

Гамалій А.М.

(прізвище та ініціали)

Науковий керівник _____

(підпис)

Рязанцев О.І.

(прізвище та ініціали)

АНОТАЦІЯ

Гамалій А.М. Дослідження методів анотування відеопотоку та розробка програмного додатку для анотування відео.

Метою магістерської роботи є дослідження методів анотування відеопотоків і реалізація запропонованого методу для створення спеціалізованого програмного засобу для анотації відеопотоків.

Результатом роботи є реалізація алгоритму анотування відео на основі кластеризації у вигляді програмного додатку.

Ключові слова: піксел, анотування, відео, обробка зображень, кластеризація.

АННОТАЦИЯ

Гамалий А.Н. Исследование методов аннотирования видеопотока и разработка программного приложения для аннотирования видео.

Целью магистерской работы является исследование методов аннотирования видеопотоков и реализация предложенного метода для создания специализированного программного средства для аннотации видеопотоков.

Результатом работы является реализация алгоритма аннотирования видео на основе кластеризации в виде программного приложения.

Ключевые слова: пиксел, аннотирования, видео, обработка изображений, кластеризация.

ABSTRACT

Hamalii A.N. Investigation of video summarization methods and development of video summarization software application.

The aim of certification diploma is to study the methods for video summarization and implementing the proposed method for creating specialized software for video summarization.

The result of the work is the implementation of a video summarization algorithm based on clustering as a software application.

Key words: pixel, summarization, video, image processing, clustering.

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень..... | 7 |
| Вступ..... | 8 |
| 1 Аналіз предметної області та постановка задачі | 9 |
| 1.1 Відеосумаризація..... | 9 |
| 1.2 Підходи до сумаризації відео | 12 |
| 1.2.1 Сумаризація на основі особливостей | 13 |
| 1.2.2 Сумаризація на основі технології | 16 |
| 1.2.3 Оцінка сумаризації | 18 |
| 1.3 Методи сумаризації відео | 19 |
| 1.4 Постановка задачі дослідження | 27 |
| 2 Розробка алгоритму сумаризації відео | 28 |
| 2.1 Відеосумаризація шляхом кластеризації | 28 |
| 2.2 Представлення відеокадру | 30 |
| 2.3 Кластеризація на основі представників | 33 |
| 2.4 Алгоритм Dynamic Time Warping..... | 37 |
| 2.5 Метрики оцінки для сумаризації | 43 |
| 3 Розробка програмного додатку для анотування відео | 49 |
| 3.1 Обґрунтування вибору програмного середовища | 49 |
| 3.1.1 Огляд технологій при розробці проекту на C#..... | 49 |
| 3.1.2 Огляд технологій при розробці проекту на JavaScript | 52 |
| 3.2 Програмна реалізація..... | 59 |
| 4 Охорона праці та безпека в надзвичайних ситуаціях..... | 61 |
| 4.1 Загальні питання з охорони праці..... | 61 |
| 4.2 Аналіз стану умов праці..... | 61 |
| 4.2.1 Вимоги до приміщень..... | 62 |
| 4.2.2 Вимоги до організації місця праці..... | 62 |
| 4.2.3 Навантаження та напруженість процесу праці..... | 63 |
| 4.3 Виробнича санітарія..... | 63 |
| 4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу..... | 63 |
| 4.3.2 Пожежна безпека..... | 65 |
| 4.3.3 Електробезпека..... | 65 |
| 4.4 Гігієнічні вимоги до параметрів виробничого середовища..... | 66 |

| | |
|---|----|
| | 6 |
| 4.4.1 Мікроклімат..... | 66 |
| 4.4.2 Освітлення..... | 66 |
| 4.4.3 Шум та вібрація, електромагнітне випромінювання..... | 68 |
| 4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій..... | 68 |
| 4.6 Екологія..... | 72 |
| Висновки до розділу 4..... | 72 |
| Перелік корисних посилань до розділу 4..... | 73 |
| Висновки | 74 |
| Перелік джерел посилань..... | 75 |
| Додаток А Лістинг «DtwMatrixPlot.xaml.cs» | 79 |
| Додаток Б Лістинг «DtwByVariablePlot.xaml.cs» | 86 |
| Додаток В Комп'ютерна презентація | 89 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HSV – відтінок, насиченість, співвідношення

PME – сприймальна енергія руху

MST – мінімальне сполучне дерево

SVD – сингулярна декомпозиція

ВСТУП

Мета магістерської роботи – провести аналіз існуючих методів анотування відеопотоків та розробити власний програмний додаток для анотування відео, який буде мати практичне застосування.

З швидким зростанням відеоконтенту сумаризація відео, яка фокусується на автоматичному виборі важливих і інформативних частин, стає все більш актуальною. Однак, проблема дуже складна через її суб'єктивність. Дослідження, які в основному покладаються на власні розроблені критерії, часто не можуть досягти таких результатів, які б задовольнили всіх. З метою забезпечення ефективного способу перегляду великої колекції відео, відеосумаризація привернула велику увагу в минулому десятилітті. Було запропоновано кілька підходів сумаризації відео, використовуючи знання про домен або навчання контрольованої моделі з маркуванням баз даних. Однак досі залишається актуальним завдання формулювання такої моделі, яка буде здатна впоратись з усім різноманіттям відеоконтенту без людського спостереження.

Проаналізувавши програмну частину атестаційного проекту було вирішено реалізувати його за допомогою мов програмування C# та JavaScript, оскільки вони містять у собі необхідні засоби та бібліотеки для успішного досягнення поставленої мети.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Відеосумаризація

Величезна популярність відео сайтів, таких як YouTube, Yahoo Video і соціальних мереж, таких як Facebook, Google+ стимулювала масове зростання відео контенту в Інтернеті. Щоб керувати зростаючою кількістю відеороликів в Інтернеті, а також інформацією від них, більше уваги приділялося відеосумаризації. Це механізм, який створює коротке прев'ю, щоб дати користувачам синтетичну і корисну візуальну анотацію відеоконтенту. Загалом, там представлені дві різні форми сумаризації відео: статична сумаризація і динамічна сумаризація відео. Статична сумаризація включає в себе набір ключових кадрів з відео і не має обмежень часу і послідовності. Динамічна сумаризація містить невелику частину відеороликів, об'єднаних в хронологічному порядку і більше схожа на коротшу версію оригінального відео. Крім того, мініатюра відео, яка є першим що користувач бачить при перегляді або пошуку відео, може вважатися особливим видом відеореклами на найвищому рівні абстракції, включаючи тільки один фрейм.

Сумаризація відео є складною проблемою через свою суб'єктивність - у користувачів є свої переваги в цьому. Проте, зі всієї безлічі можна виділити однакові переваги. Тому можна вибрати важливі і змістовні частини з відео, які можуть в основному задовольнити більшість користувачів.

Щоб вирішити цю проблему, деякі неконтрольовані підходи часто вибирають кадри з відео за допомогою вручну розроблених критеріїв, таких як візуальна увага, репрезентативність і важливість. Однак створені вручну критерії часто не можуть підходити для різноманітних відеороликів в Інтернеті. На відміну від неконтрольованих, контрольовані підходи вчили систему безпосередньо дізнатися зі створених людиною *summary*, як вибрати підмножину, щоб відповідати метрикам, отриманим від людини. Хоча іноді вони ефективні, вони поклалися в значній мірі на вказівки людини, які важко отримати програмно.

Насправді, люди дуже хороші в суммаризації інформації на словах. Коли люди дивляться відео, це звичайна справа для них, щоб підсумувати відеоконтент і ділитися один з одним словами, і те, що вони говорять, може розглядатися як зведення відео в текстовій формі. Різні люди можуть висловлювати різний зміст, але основна тема і семантичний сенс все ті ж. Кадри, які найбільш актуальні для загального вираження від людей дійсно «гідні підсумовування», так як вони відображають змістовну інформацію з

відео і представляють те, що стосується людей. Неможливо отримати текстові зведення відео безпосередньо від людей, але, як показано на рисунку 1.1, є деяка додаткова інформація (наприклад, навколишній текст, такий як заголовки, запити, опису, коментарі і т. д.), пов'язана з відео.

Додаткова інформація, що розглядається як непрямий зворотній зв'язок від людини, являє собою свого роду людино-кураторську семантику відеозмісту. Незважаючи на те, що він цінний для сумаризації відео, часто ігнорується.

Деякі недавні роботи спробували використовувати заголовки і запити, пов'язані з відео, щоб визначити важливість відеокадрів. Дехто намагався знаходити зображення в Інтернеті за назвою відео. Вони сподівалися отримати корисну візуальну інформацію від цих зображень, щоб знайти необхідні для сумаризації кадри з відеороликів. Інші безпосередньо зіставили візуальні особливості з фіксованим текстовим простором через лінійне перетворення, тому відеоескізи можуть бути обрані шляхом вимірювання їх відстаней до інформації в текстовому просторі. При цьому підході, критична проблема полягає в тому, щоб вибрати семантично значущі частини відео, які щільно корелюють з додатковою інформацією. Тому, як ефективно вимірювати смислову релевантність між відеокадрами і додатковою інформацією має велике значення. Однак попередні роботи не приділяли достатньої уваги цьому питанню, що призвело до проблеми.



(a) Відео і ключові кадри, які вибрані людським анотатором

Назва: Як поставити замок на велосипед ПРАВИЛЬНО!

Запит: Встановити замок на велосипед.

Опис відео: У цьому відео Аманда навчить нас правильно встановлювати замок на велосипед.

Коментар: Замість використання одного замка, можна використовувати два. Крадіві прийдеться різати зразу два замка.

(b) Стороння семантична інформація

Рисунок 1.1 - Приклад відео з Youtube і його додаткової семантичної інформації (a) являє собою відео і п'ять ключових кадрів, обраних людським анотатором, (b) показує

сторонню семантичну інформацію, таку як назва відео, користувальницький запит, опис відео, коментар користувача до відео в (а).

По-перше, існує як загальна інформація, так і спеціальна інформація в відео і оточуючих їх текстах, тоді як вони ігнорували те, що спеціальна інформація шкідлива для вимірювання семантичної релевантності. Як показано на рисунку 1.1, навіть якщо відео і його опис мають загальну інформацію, таку як «велосипед» і «замок», там все ще є деякі характеристики, які не можуть бути зкорельовані. Наприклад, «Правильний шлях!» - це текстова інформація, яку важко зафіксувати в відео. У той час як стіни і вулиці є візуальною інформацією, яка не може бути зображена як текст. Інтуїтивно, це загальна інформація, а не конкретна, допомагає нам зіставляти відповідні предмети з двох різних джерел. Отже, побудова нового прихованого підпростору, де може бути збережена тільки загальна інформація, є найкращим вибором для вимірювання семантичної релевантності.

По-друге, втрата загальної інформації в прихованому підпросторі також призведе до погіршення продуктивності при зіставленні відеокадрів і додаткової інформації. Наприклад, також, як показано на рисунку 1.1, є «велосипед, замок і вулиця», зображена в відеокадрі, і є «велосипед, замок і люди», описані в заголовку відео. При виділенні відеокадра і заголовка для прихованого підпростору без будь-яких інших обмежень може бути тільки часткове кінцеве поле «велосипед» (або «замок»). Ця неповна загальна інформація може привести до неповного виміру подібності. Отже, зберігаючи більш повну загальну інформацію (як «велосипед» і «блокування») в прихованому підпросторі, відбувається більш надійний вимір семантичної релевантності.

Сумаризація відео може бути розглянута з різних точок зору. Нижче будуть приведені найбільш загальні підходи - предметно-орієнтована сумаризація, контрольована і неконтрольована.

Предметно-орієнтована відеосумаризація - може допомогти визначити важливі кадри відео. Наприклад, спортивні відеоролики містять канонічні сцени, такі як «біг до бази» в бейсболі і «тачдаун» в футболі; ці кадри можуть використовуватися для генерації деяких моментів. Аналогічним чином можна використовувати траєкторії, щоб підсумувати тактичну інформацію в футбольних іграх. У відеоспостереженні більшість кадрів містять стаціонарний фон, і, таким чином, можна узагальнити їх на короткий огляд.

Контрольована сумаризація: багато робіт було запропоновано в вимірі важливості кадру під наглядом. У відео від першої особи можна узагальнити вивчаючи важливі об'єкти. Щоб передбачити цікавий для нас кадр, низькорівневі, високорівневі і просторово-часові функції були об'єднані для навчання лінійної регресійної моделі. Точно

так же важливість кадру вимірювали попередньо підготовленим тематичним бінарним класифікатором SVM або SVM ранжувальником. Крім того, з невеликою кількістю міток, була створена ієрархічна модель для створення відеопідсумку, що містить об'єкти інтересів.

Неконтрольована сумаризація - це неконтрольована відеосумаризація, яка не вимагає знань про домени або позначені кадри, але замість цього шукає візуальну релевантність або додаткові ресурси для визначення важливості кадру. Один популярний підхід зменшує візуальну надмірність шляхом навчання словника базових кадрів, або виконує ієрархічний кластерний аналіз. Інші роботи досліджували увагу людини під час перегляду відео в порядку для захоплення перцепційної якості відеороликів для вибору основних моментів. Кілька відео можна сумаризувати, використовуючи набір обраних ключових кадрів. Ще одна недавня тенденція полягає в тому, щоб підсумувати відео з онлайн-зображеннями, такими як набір зображень з канонічними видами або потік фотографій, які робляться послідовно. Такі методи генерують підсумок ключових кадрів з використанням кореляцій між відео кадрами і колекцією зображень.

1.2 Підходи до сумаризації відео

Відео являє собою мультимедійну послідовність зображень, яка також може містити звук. З метою обробки, довге відео може розбиватися на невеликі сегменти на основі сцен або кадрів. Збір семантичних і тимчасово пов'язаних груп елементів відео, які передають значення більш високого рівня, називаються сценою. Існує безліч методів сумаризації відео, які засновані на розкладанні рівня сцени. Кадр визначається як послідовність дій, знятих однією камерою без будь-яких великих змін в візуальному контенті. Він являє собою фізичну концепцію, і зазвичай використовується в методах, які засновані на кадрованих границях. Границя кадру визначається різкими і поступовими змінами в кадрах. І «представницький» кадр з кожної сцени буде обраний для побудови підсумку всього відео.

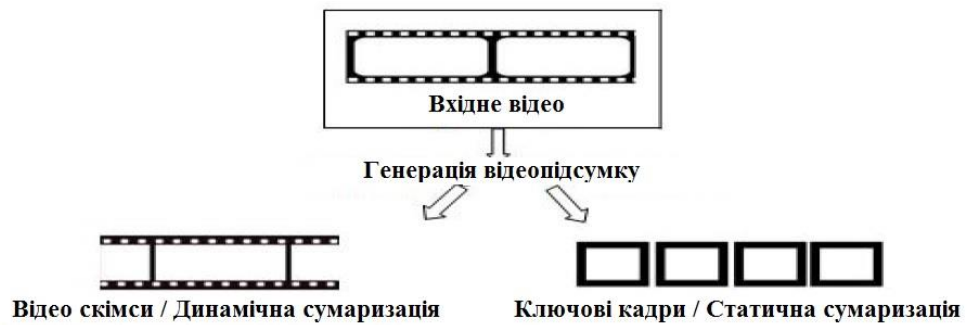


Рисунок 1.2 – Типи відеосумаризації

Існує безліч методів суммаризації відео з різною основою класифікації. Однак, існуючі методи можуть бути класифіковані на основі використовуваних функцій: сумаризація на основі низько- і високорівневих особливостей (Low Level Feature based Video Summarization and High Level Feature based Video Summarization), одно- і багатокomпонентна сумаризація (Single Feature based Video Summarization and Multiple Feature based Video Summarization); на основі використовуваної техніки - кластерна і некластерна сумаризація (Clustering based Video Summarization і Non-Clustering based Video Summarization), і нові методи. Нижче описується класифікація методів.

1.2.1 Сумаризація на основі особливостей

Залежно від типу і кількості функцій, які використовуються для визначення представлення кадру і візуальної ідентифікації важливо, щоб методи сумаризації відео можна було класифікувати, як показано нижче.

1) Сумаризація на основі низькорівневих особливостей: методи, які були використані раніше для відеозйомки, були засновані на низькорівневих особливостях, таких як колір, текстура, рух для визначення візуально важливої і релевантної інформації з відео. Найбільш часто використовувана функція низького рівня є колір. Його легко обчислити. Гістограми кольору отримані для кожного кадру і засновані на перетині або різниці гістограм, і можна порівнювати кадри. Гістограма зображення/кольор відноситься до масової функції інтенсивності і обчислюється шляхом підрахунку кількості пікселів одного кольору. Хоча колірна модель RGB є найбільш часто використовуваною моделлю, колірна модель HSV має близькість до сприйняття людини, тому вона більш краща, ніж RGB. Кольорові гістограми нечутливі до малих рухів камери і легко обчислюються. Під час порівнюючи два кадри за допомогою кольорових гістограм, можна зіткнутися з

ситуацією, коли два кадри повністю різні, але гістограми однакові через аналогічний колірний розподіл кадрів. Текстура також є широко використовуваною функцією низького рівня. Існує багато методів для витягу текстур, в основному для аналізу текстури використовується вейвлет-трансформація.

Рух вважається однією з важливих функцій для захоплення візуально цікавих елементів. Найбільш поширені методи для обчислення руху - це гістограми руху і аналіз оптичного потоку. Гістограми руху аналогічні гістограмам кольору. Аналіз оптичного потоку є найпростішим і поширеним методом обчислення руху. Аналіз оптичного потоку заснований на яскравості, що обмежує можливість застосування методу до навколишнього середовища, на яку не впливають зміни освітлення. Перевага використання низькорівневих функцій полягає в тому, що їх легко обчислити, що робить ці функції основними для додатків, які вимагають швидкої реакції і в яких сцени не складні. Але існує багато складних сцен, що мають важливу інформацію, які залишаються непоміченими, якщо використовуються функції низького рівня. Таким чином, зі стрімко ростучим рівнем технологій якість відео і складність сцен також збільшилася і, таким чином, перемістила фокус з низькорівневих функцій на високорівневі функції для ідентифікації візуально важливого вмісту з відео.

2) Сумаризація на основі високорівневих особливостей: функції низького рівня легко обчислюються, але вони не близькі до семантики, однак такі високорівневі функції, як розпізнавання об'єктів, виявлення подій, виявлення жестів, розпізнавання осіб, розпізнавання емоцій і т. д. допомагає краще зрозуміти зміст відео. Багато роботи зроблено в області виявлення подій в спортивних відео для створення коротких оглядів про деякі спеціальні події в грі, таких як гол у футболі, бій в боротьбі і т. д. Моделі пропонувані в таких випадках навчаються розпізнавати такі події. Одні методи використовують емоції і визначають семантично важливий зміст. В інших статичні зведення створюються для відеороликів, знятих ношеними камерами. Об'єктно-керований підхід використовується для визначення важливого вмісту, в якому він використовує особливості області та функції об'єкта для обчислення релевантності вмісту фрейма. Високорівневі підходи сумаризації відео дуже корисні для додатків, які дуже чутливі до вмісту. Незважаючи на свої переваги, високорівневі методи сумаризації, засновані на характеристиках, займають багато часу і мають високу складність обчислень.

3) Однокомпонентна сумаризація. Найпростіші методи сумаризації відео використовують єдину функцію зазвичай або колір, або рух як єдиний дескриптор. Використання одного компонента спрощує завдання обчислення і економить час.

Більшість раніше виконаних робіт було засновано тільки на колірних моделях. Існує багато досліджень, які використовують рух як єдину особливість, щоб знайти відповідний зміст в відео. За аналогією з колірною гістограмою, були також запропоновані гістограми руху. Тіло в русі спочатку прискорюється, а потім сповільнюється. Використовуючи цей закон, і з припущенням, що конкретна точка, де рух починає сповільнюватися - це точка інтересу, а трикутник між прискоренням і уповільненням є енергією руху. З цим припущенням була розроблена модель PME (Perceived Motion Energy). У моделі PME малюнок руху моделюється, і після цього витягуються ключові кадри. Методи сумаризації відео на основі окремих функцій є обчислювально простими і легкими, але вони можуть не завжди забезпечувати точні результати.

4) Сумаризація на основі декількох компонентів: зі збільшенням складності вмісту відео було усвідомлено, що одного дескриптора недостатньо для оцінки важливості та актуальності змісту. Були запропоновані методи, які використовують відразу кілька компонентів. Одні з них засновані на уявленні трьох ознак для гістограм кольору кадру, гістограм напрямку кромки і текстурної статистики. Інші агрегували кореляцію колірних каналів RGB, моменти інерції і колірні гістограми з використанням адаптивної формули для генерації відеопідсумку. Комбінуючи кілька ознак, деякі з методів використовували формулу, яка є лінійною функцією. Але є багато реальних ситуацій, коли не всі функції однаково важливі і можуть не представляти для нас інтерес. Методи, в яких кожній функції присвоюється певна вага, заснована на її відносній важливості. Одні використовували СММ (модель випуклих сумішей) для обчислення ваги різних візуальних дескрипторів, тоді як інші привласнювали зумовлені ваги різних функцій. Ці ваги були емпірично визначені. Моделі уваги користувачів - це обчислювальна модель, що імітує візуальну модель людини, заснована на тому, що людське око чутливе як до колірного контрасту, так і до руху, тому ці моделі об'єднують ці функції для обчислення кривих уваги. Використання декількох функцій для оцінки значущості змісту робить підхід сумаризації більш ефективним. За допомогою кількох функцій досліджують і використовують вміст з кожного аспекту, але в той же час здійсненність методу слід оцінювати заздалегідь, тому що в деяких випадках недоцільно використовувати кілька функцій для оцінки.

1.2.2 Сумаризація на основі технології

Традиційні підходи до сумаризації відео засновані на методах кластеризації, але існують деякі некластерні методи, які також можуть використовуватися для сумаризації відео. Обидві категорії описані нижче.

1) Сумаризація відео на основі виявлення знімків: відео є безперервною послідовністю зображень, які іноді діляться на невеликі кадри, засновані на ідентифікації його меж. Границя кадру в основному визначається або різкими змінами у змісті відео або поступовими (зникнення) змінами змісту. Після виявлення кадрів на основі деяких функцій, в якості «представника» кадру вибирається рамка з кадру. Всі репрезентативні рамки з кадрів сприяють сумаризації відео. Існують деякі методи, які використовують різниці кадрів (колірні гістограми) для виявлення їх кордонів для подальшого відбору репрезентативних кадрів. Відмінності в кадрах зазвичай використовуються для виявлення кордонів кадрів. Відмінності в кадрах можна обчислити, використовуючи будь-яку функцію, але найбільш часто використовується функція, яка легко обчислюється, є колірна гістограма.

Алгоритми виявлення кордонів кадру прості в реалізації, але вони мають обмежену застосовність. Ці алгоритми залежать від порогових значень. Якщо задано дуже маленьке значення, це призведе до великої кількості кадрів і, отже, призведе до тривалої сумаризації. Кращі результати алгоритму виявлення кордонів кадру можна отримати на професійно створених відео, але вони не можуть бути використані для створених користувачами відео. Призначені для користувача відеоролики мають безперервний характер і не мають різких змін змісту.

2) Сумаризація на основі кластеризації. Основна ідея кластерних підходів полягає в тому, щоб розглядати кожен кадр як точку і застосовуючи алгоритм кластеризації, представники кластеру вибираються в якості ключових кадрів. Існує багато алгоритмів кластеризації - k-means, k-medoids, кластеризація спектрального типу, кластеризація MST (мінімальне сполучне дерево) і т. д., які були використані в літературі з метою сумаризації відео. У методах вилучення ключових кадрів на основі кластеризації (статична відеосумаризація) зазвичай генеруються ключові кадри, які не перебувають в хронологічному порядку, що часто створює семантичний розрив для розуміння послідовності подій. Таким чином, для вирішення цієї проблеми необхідно вживати заходів, спрямованих на збереження часової послідовності відео.

3) Некластерні підходи до сумаризації: існує безліч методів, заснованих на визначенні порогових значень для вибору ілюстративних кадрів відео. Наприклад, метод

заснований на пороговому значенні для послідовного вибору ключових кадрів опису. Сума абсолютних відмінностей гістограм кадрів і кадри, для яких сума абсолютної різниці більше визначеного порогу, будуть обрані в якості ключових кадрів. Ефективність методів, заснованих на порогових значеннях, в значній мірі визначається граничним значенням. Якщо поріг значення дуже низький, тоді кількість ключових кадрів буде великою. Більш того, щоб емпірично визначити порогове значення, повинні бути сформовані набори з усіма типами даних. Існують реалізації, в яких були використані методи, засновані на різністній основі на основі кривизни. Існують методи, засновані на різниці кадрів на основі пікселів. Хоча легко обчислити кадр на основі пікселів, але цей підхід не дає надійних результатів при зміні кута камери. Пізніше з'ясувалося, що можливе вирішення цієї проблеми можливо поділ кадру на блоки і замість використання заснованих на пікселях кадрів повинні бути розраховані відмінності блоків. Деякі використовували візуальну модель для з'ясування візуально значимого вмісту. У описаній вище моделі уваги користувача, обчислюються статичні динамічні і семантичні криві уваги. Отримана глобальна крива уваги необхідна для сумаризації відео. У цьому підході криві глобальної кривої уваги розглядаються як ключові кадри, які представляють собою візуально значущий зміст відео. Однією з найбільших переваг моделі уваги користувача є те, що вона не вимагає повного розуміння візуального контенту. Вона фіксує когнітивні функції, відповідальні за візуалізацію різної інформації. Деякі з доведених фактів і висновків можна використовувати для моделювання візуальної уваги. Хоча є багато переваг методів візуалізації відео на основі візуальної привабливості, але є деякі недоліки цих моделей. Вони складні, так як вони аналогічні візуальній системі людини, крім того, існують різні моделі уваги, які складні для розрахунку загального візуально привабливого вмісту. Ці моделі уваги створюють складність. Однак, ця складність може зменшуватися за допомогою паралельної обробки.

Алгоритм SIFT (Scale Invariant Feature Transform) не чутливий до зміни освітлення, масштабу і шуму. Деякі використовували SIFT для вилучення ключових точок кадрів, а потім представляли кожен кадр за допомогою гістограм точок SIFT, межі кадру були виявлені, а потім ключові рамки були витягнуті з кожного кадру. Перевага алгоритму SIFT полягає в тому, що він не схильний до шуму, зміни освітлення, повороту і масштабуванню (збільшення або зменшення). Але, незважаючи на ці переваги, головним недоліком є його обчислювальна складність, крім того, він має потребу в навчанні системи. Ці недоліки роблять цей алгоритм менш ймовірним для вибору для додатків реального часу.

Наведені підходи до сумаризації відео були в основному засновані на понятті використання тільки функції зображення. Але є й інші, не візуальні джерела інформації, такі як аудіо, час, текстові дані з сайтів соціальних мереж і т. д., які в поєднанні з даними зображення можуть бути дуже корисні для вилучення візуально важливого вмісту відео. Хоча вивчення з декількох джерел звучить цікаво, але в цих техніках є багато перешкод. Доступність даних без зображень і вплив шуму на звукові дані, зняті в громадських місцях, збільшують обчислювальну складність. Методика сегментації спортивного відео, заснована на подіях, яка використовує текстові дані для веб-трансляції разом з тимчасовими мітками подій.

1.2.3 Оцінка сумаризації

Отримані відеоматеріали повинні оцінюватися для підтвердження їх переваг над іншими методами. Однак ця область відеоаналізу не має стандартних показників ефективності і методів оцінки. Відсутність стандартних методів робить процес більш складним. Оцінка сумаризації відео є дуже суб'єктивним завданням, але деякі методи існують. Всі методи можна класифікувати як:

1) Суб'єктивна оцінка. Суб'єктивна форма оцінки сумаризації має дуже велику залежність від людей. Найосновнішим підходом була візуальна оцінка підсумків, але через відсутність практичності підходу були розроблені деякі інші методи. Суб'єктивна оцінка була згенерована користувачами, які переглядали відео. Група користувачів, яким було показано відео, а потім відео-анотація, була обрана з різних верств і вікових груп, щоб уникнути упередженості. Група користувачів зазвичай коментує відео, а анотації порівнюються з сумаризацією, щоб оцінити їх. Крім цього, є деякі інші параметри для оцінки, такі як інформативність і приємність вмісту автоматично згенерованого відео-резюме.

2) Об'єктивна оцінка. Без будь-яких об'єктивних результатів процедури оцінки стає важко оцінити автоматично згенерований відеопідсумок. Існують деякі методи, які намагаються використовувати об'єктивні методи оцінки; дуже поширеною і широко використовуваною технікою є CUS (Порівняння відеопідсумків користувачів). CUS передбачає втручання людини, але в той же час він дає і кількісні результати. У CUS група користувачів генерують відеопідсумок; відеопідсумки, згенеровані користувачем, порівнюються з автоматично згенерованими. Кадри зі згенерованих користувачем і

автоматично згенерованих відео порівнюються шляхом обчислення відстані Манхеттена між ними або з використанням різниці колірних гістограм.

Недавні досягнення в області відеоаналітики зумовили необхідність автоматичної сумаризації відео. Було представлено багато методів в цій області. Відзначається, що не всі методи сумаризації підходять до кожної ситуації. Деякі з методів гарні для додатків реального часу, так як вони обчислювально прості і швидкі; деякі методи підходять для додатків, які вимагають точних результатів незалежно від часу, витраченого на підготовку зведення (наприклад, спостереження). Кожен метод має свої переваги і недоліки. Так само варто відзначити, що відсутні стандартні методи оцінки, що значно ускладнює якісну оцінку методу.

1.3 Методи сумаризації відео

Сумаризація відео є механізмом для генерації короткого відеопідсумку, який може бути або послідовністю нерухомих зображень (ключових кадрів), або рухомих зображень (відео). Відео можна сумаризувати двома різними способами, які полягають в наступному:

1) Сумаризація відео на основі ключового кадру.

Вони також називаються репрезентативними кадрами, R-кадрами, а набір складається з набору виділених зображень, витягнутих з основного джерела. Нижче наводяться деякі проблеми, через які слід дотримуватися обережності при реалізації алгоритму на основі ключового кадру:

- резервування: в якості ключового кадру вибираються кадри з незначною різницею;
- при різних змінах контенту складно зробити кластеризацію.

2) Сумаризація на основі відеозображення.

Це також називається абстракцією рухомого зображення, що рухається сюжетом або вільною послідовністю. Початкове відео сегментується на різні частини, що є відеокліпом з меншою тривалістю. Кожен сегмент з'єднаний або розрізом, або поступовим ефектом. Трейлер фільму - найкращий приклад для цього підходу.

Сумаризація відео являє собою зведення, яка представляє собою абстрактний вид вихідної відеопослідовності і може використовуватися для перегляду і пошуку відео. Зазвичай це являє собою набір ключових кадрів. Для вибору ключових кадрів можуть бути використані різні методи

Використовуючи трикутну модель сприймаємої енергії руху (PME), в відео моделюються рухомі моделі. Рамки точок прискорення руху і уповільнення руху обрані в якості ключових кадрів. Процес вибору ключового кадру є гранично вільним і швидким, а виділені ключові кадри є репрезентативними.

У візуальному алгоритмі дескрипторів кадру є три візуальні функції: гистограма кольору, статистика вейвлетів і гистограма напрямку, які використовуються для вибору ключових кадрів. Методи подібності обчислюються для кожного дескриптора і утворюють міру різниці кадрів. Ступінь відновлення кадру, якість стиснення використовуються для оцінки сумаризації відео.

У моделі Motion Attention Model кадри виявляються з використанням розподілу кольору і коефіцієнта покриття краю, які збільшують точність виявлення кадру. Ключові кадри витягуються з кожного кадру з використанням моделі уваги руху. Тут перший і останній кадр кожного кадру розглядаються як ключові, а решта витягуються шляхом прийняття Motion Attention Model. Ці ключові кадри потім групуються, і значення пріоритету обчислюється шляхом оцінки енергії руху і колірної варіація знімків.

В алгоритмі множинних візуальних дескрипторів (Multiple Visual Descriptor) ключові кадри вибираються шляхом побудови накопичувального графіка для значень різниці кадрів. Кадри на різкому схилі вказують на значну візуальну зміну; отже, вони обрані і включені в підсумковий summary.

Метод фокусування руху (Motion focusing method) фокусується на одному русі постійної швидкості і вирівнює відеокадри, фіксуючи сфокусований рух в статичну ситуацію. Згенерований підсумок має всі рухомі об'єкти і включає просторову інформацію про рух. Віднімання фону і мінімальне скорочення в основному використовуються при фокусуванні руху.

У Camera Motion і Object Motion відео сегментується з використанням камер, заснованих на русі: панорамування, масштабування, і фіксація. Остаточний вибір ключових кадрів з кожного з цих сегментів витягується на основі значення достовірності розробленого для масштабування, панорамування і стійких сегментів.

Сумаризація відеозображень на основі ключових кадрів працює з кадрами, тому першим кроком є отримання кадрів від вихідної послідовності відео. На другому кроці витягнуті відеокадри є кластер з надмірним контентом. Вибір ключових кадрів виконується на третьому кроці.

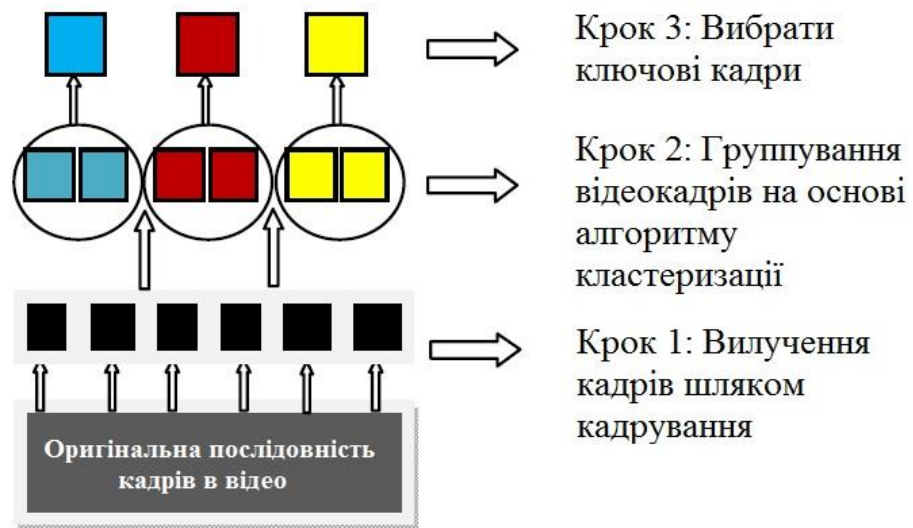


Рисунок 1.3 – Вибір ключових кадрів при сумаризації на основі ключових кадрів

Сумаризація відео на основі ключових кадрів може бути класифікована трьома різними способами:

1) Класифікація на основі вибірки. Цей метод вибирає ключові кадри рівномірно або випадковим чином без вибірки, не розглядаючи відеоконтент. Підсумок, створений цими методами, не представляє всіх частин відео і може викликати деяку надмірність ключових кадрів з схожим вмістом.

2) Класифікація на основі сегментації сцени. Цей метод витягує ключові фрейми з використанням виявлених сцен. Сцена включає в себе всі частини з семантичним зв'язком в відео або в одному і тому ж місці, або в один і той же час. Недоліком цих методів є отриманий підсумок, який не враховує тимчасове положення кадрів.

3) Класифікація на основі сегментації кадрів. Цей метод витягує адаптовані ключові кадри для відеоконтенту. Вони витягають перше зображення в якості знятих ключових кадрів або початок і кінець кадру. Ці методи ефективні для зміни стаціонарного кадру і невеликого вмісту, але вони не забезпечують адекватного уявлення кадру з сильними рухами. Нижче наведені різні методи вилучення ключових кадрів:

- Сумаризація відео шляхом кластеризації з використанням евклідової відстані. Цей метод заснований на видаленні надлишкових відеокадрів з майже однаковим контентом. Як і багато інших підходів, весь відеоматеріал спочатку кластеризується в вузли, кожен з яких містить кадри аналогічного візуального змісту. Представляючи кожен кластер з його найбільш представницьким кадром, виходить набір ключових кадрів, який потім підсумовує цю послідовність. Процедура для цього методу показана на рисунку 1.4.



Рисунок 1.4 – Сумаризація шляхом кластеризації з використанням евклідової відстані

- Perceived Motion Energy Model (PME). Рухи є найпомітнішою особливістю при поданні дій або подій в відео і, таким чином, повинна бути функція визначення ключових кадрів за допомогою цього. Трикутна модель сприймання енергії руху моделює рух в відео, витягує ключові кадри на основі цієї моделі. PME представляє собою комбіновану метрику інтенсивності руху та виду руху з великим упором на домінуючий рух відео. Середня величина $Mag(t)$ векторів руху в усьому кадрі обчислюється, як описано у формулі 1.1.

$$Mag(t) = \frac{\left(\frac{\sum (MixFEN_{i,j}(t))}{N} + \frac{\sum (MixBEN_{i,j}(t))}{N} \right)}{2}, \quad (1.1)$$

де $MixFEN_{i,j}(t)$ являє вектори руху вперед, а $MixBEN_{i,j}(t)$ являє вектори зворотного руху, N – кількість макроблоків у кадрі.

Відсоток домінуючого напрямку руху (t) визначено в формулі 1.2:

$$\alpha(t) = \frac{\max(AH(t, k), k \in [1, n])}{\sum_n^{k=1} AH(t, k)}, \quad (1.2)$$

де (t, k) є гістограмою кута з n ячейками.

РМЕ B -кадру обчислюється як $P(t) = MAg(t) \cdot \alpha(t)$. Ці значення РМЕ кадрів будуються як представляючі послідовність трикутника руху. Кадри в точці повороту прискорення руху і уповільнення руху вибираються в якості ключових кадрів. Процес вибору кадру є граничним, вільним і швидким. Тут спочатку відеопослідовність сегментується в кадри, використовуючи метод порівняння *twiп*. Ключові кадри вибираються на основі шаблонів руху всередині кадрів. Для кадрів з моделлю руху трикутна модель використовується для вибору ключового кадру, тоді як для знімків без моделі руху, перший кадр вибирається як ключовий. Має задовільну швидкість для спортивних і розважальних відеороликів, так як в них набагато більше дій в порівнянні з домашнім і новинним відео.

- Дескриптори візуальних кадрів. G. Ciocca і R. Schettini представили алгоритм з трьома візуальними особливостями: колірною гістограмою, вейвлет статистикою і гістограмою напрямку краю, які використовуються для вибору ключових кадрів. Міри подібності розраховуються для кожного дескриптора і об'єднуються для формування міри різниці кадрів. Відстань між двома кольоровими гістограмами d_{H} обчислюється:

$$d_H(H_t, H_{t+1}) = 1 - \sum_{j=0}^{63} \min(H_t(j), H_{t+1}(j)), \quad (1.3)$$

Різниця між двома гістограмами напрямку краю d_D обчислюється з використанням евклідової відстані в разі двох статистик вейвлетів d_W ,

$$d_D(D_t, D_{t+1}) = \sqrt{\sum_j^{71} (D_t(j) - D_{t+1}(j))^2}, \quad (1.4)$$

$$d_W(W_t, W_{t+1}) = \sqrt{\sum_j^{19} (W_t(j) - W_{t+1}(j))^2}. \quad (1.5)$$

Ці відмінності об'єднуються, щоб сформувати остаточну міру різниці кадрів d_{HWD} :

$$d_{HWD} = d_H \cdot d_W + d_W \cdot d_D + d_D \cdot d_H. \quad (1.6)$$

Ці значення різниці використовуються для побудови кривої кумулятивних різниць кадрів, яка описує, як візуальний контент кадрів змінюється. Визначаються точки високої кривизни і витягуються два ключових кадри послідовності. Наступні якості використовуються для оцінки сумаризованого відео:

1. Точність: міра визначається як напів-гаусдорфова відстань.
2. Ступінь відновлення знімка (SRD): використовується відповідний алгоритм інтерполяції кадрів; повинні вміти відновити всю послідовність з набору ключових кадрів.
3. Співвідношення стиснення (CR): CR визначається як відношення кількості ключових кадрів і загальної кількості кадрів в послідовності відео.

- Motion Attention Model. I. C. Chang використовував цю модель для виявлення кадрів. У цій моделі кадри виявляються з використанням розподілу кольорів і коефіцієнту покриття краю, який підвищує точність виявлення.

Ключові кадри витягуються з кожного знімка використовуючи модель уваги руху. Тут перший і останній кадр кожного кадру розглядаються як ключові, а інші витягуються шляхом прийняття моделі уваги руху. Ці ключові кадри потім групуються і пріоритет їх значення обчислюється шляхом оцінки енергії руху і зміни кольору знімків. Енергія руху визначається як (1.7):

$$TMA \left(S_i = \sum_{f_j \in S_i} MA(f_i \times \log \left(\sum_{f_j \in S_i} MA(f_j) \right)) \right), \quad (1.7)$$

де $T(S_i)$ означає суму уваги руху значення кадру i . І зміна енергії (EMC) рахується як:

$$EMC(S_i) = TMA(S_i) \times CF(S_i), \quad (1.8)$$

де $CF(S_i)$ означає загальну кількість кадрів, які мають значну зміну інтенсивності в знімку i . Пріоритет значення кадру:

$$PV = e^{\left(\frac{EMC(S_i)}{\sum_{S_i \in C_j} EMC(S_i)} \right)}. \quad (1.9)$$

Більш високе значення PV означає, що цей знімок більш важливий для цього кластера, і кадр стане основним елементом кластера.

- Multiple Visual Descriptor Features. Chitra A.D використовували ті ж візуальні функції, що і G. Сіосса, а також одну додаткову функцію - зважений стандарт відхилення.

Зображення у відтінках сірого зосереджено на розкладанні дискретного вейвлет-сигналу L -рівня. На кожному i -му рівні ($i = 1 \dots L$) є докладними зображеннями LH , HL , HH і апроксимаційним зображенням на рівні L . Стандартне відхилення для всіх цих зображень і вектор характеристик зваженого стандартного відхилення визначено як (1.10):

$$f = \left\{ \begin{array}{l} \sigma_1^{LH}, \sigma_1^{HL}, \sigma_1^{HH}, \frac{1}{2} \sigma_2^{LH}, \sigma_1^{HL} \dots \\ \frac{1}{2^{L-1}} \sigma_L^{LH}, \frac{1}{2^{L-1}} \sigma_L^{HL}, \frac{1}{2^{L-1}} \sigma_L^{HH}, \frac{1}{2^{L-1}} \sigma^A, \mu^A \end{array} \right\}. \quad (1.10)$$

Ключові кадри вибираються шляхом побудови кумулятивного графіка для значень різниці кадрів. Різкий нахил графіка вказує на значну візуальну зміну; тому ці кадри вибираються і включаються до підсумку. І ключові кадри, відповідні середнім точкам між кожною парою точок послідовної кривизни, розглядаються як репрезентативні кадри. Алгоритм тестується на навчальній відеопослідовності і порівнюється з I -кадром, отриманим з Cue Video, і метод дає кращий результат.

- Фокусування руху. Songcong запропонував метод фокусування руху. Цей метод витягує ключові фрейми і генерує зведені для відеоспостереження за доріжкою. Цей метод фокусується на одному русі постійної швидкості і вирівнює відеокадри фіксуючи сфокусований рух в статичній ситуації. Зведення створюється з використанням всіх рухомих об'єктів і включає просторову інформацію та інформацію про рух. Метод починається з вирахування фону для вилучення рухомого переднього плану для кожного кадру. У цьому методі віднімання фону об'єднується з мінімальним розрізом, щоб отримати гладку сегментацію об'єктів переднього плану. Функція маркування f позначає кожен піксель i як передній план $f_i = 1$ або фон $f_i = 0$. Завдання маркування вирішується зведенням до мінімуму енергію Гіббса:

$$E(f) = \sum_{i \in V} E_1(f_i) + \lambda \sum E_2(f_i, f_j), \quad (1.11)$$

де E_1 і E_2 дорівнюють:

$$E_1(1) = \begin{cases} 0 & d_i > k_i^1 \\ k_i^1 - d_i k_i^2 < d_i < k_i^1, & \\ \inf & d_i < k_i^1 \end{cases} \quad (1.12)$$

$$E_1(0) = \begin{cases} 0 & d_i > k_i^3 \\ d_i - k_i^3 k_i^3 < d_i < k_i^1. & \\ \inf & d_i < k_i^1 \end{cases} \quad (1.13)$$

де $E_2 = (fi - fj)$, di є різницею між поточним фреймом і гаусовим значенням для i -того пікселя і $k^t t = 1, 2, 3$ є порогами для i -того пікселя.

Виділення ключового кадру і зведена генерація зображень виконуються за допомогою двох кроків. Первісна розмітка виконується з результатами сегментації переднього плану. Метод жадібного пошуку використовується для визначення ключових кадрів, які збільшують охоплення переднього плану на зображенні мозаїки. По-друге, загасання переноситься шляхом згладжування ключових кадрів для генерації сумаризованого зображення. Зведення не тільки відображає всі об'єкти в сфальцьованому русі, але також забезпечує тимчасовий і просторовий зв'язок.

- Рух камери і рух об'єкта. Jiebo Luo та ін. запропонували метод вилучення ключових кадрів з персональних відеокліпів. Ключові кадри витягуються з персонального відео, де контент необмежений і відсутні встановлені структури. Основа для вилучення ключових кадрів заснована на русі камери та об'єкта. Відео сегментоване на основі методів руху камери: панорамування, збільшення, зменшення і виправлення. Ключові кадри вибираються з кожного з цих сегментів. Для масштабування фокус знаходиться в кінці руху, коли об'єкт найближче. Для визначення фіксованого сегмента вибирається середина кадру сегмента або рамки, в якому максимальний рух об'єкта. Остаточний вибір ключових кадрів з кожного з цих сегментів витягуються на основі значення довіри, сформульованого для масштабування, панорамування і стійкості сегмента. Глобальна довірча функція $dpan$: $dpan = \alpha_1 dspat + \alpha_2 dknow$ з $\alpha_1 + \alpha_2 = 1$, $dspat$ - імовірнісна функція кумулятивних зсувів камери, а $dknow = G(\mu + \sigma)$ - гаусівська функція, із зазначенням локального мінімуму і стандартного відхилення, обчисленого з перекладу кривої.

Сумаризація відео грає важливу роль у багатьох відеододатках. Немає ніякого загальноприйнятого методу, доступного для відеосумаризації, який дає кращу

продуктивність у всіх типах відео. Сумаризація повністю залежить від обраного методу. Семантичне розуміння і представлення - найбільші проблеми, які необхідно вирішити, включаючи відмінності в відео і сприйнятті людини. Залежно від змін у змісті відео витягуються різні ключові кадри. Оскільки для сумаризації необхідно обробляти ключові кадри, важливий контент не можна пропускати.

1.4 Постановка задачі дослідження

Метою магістерської роботи є проведення аналізу методів анотування відеопотоку та розробка програмного додатку для анотування відео.

Для цього необхідно вирішити наступні питання:

- провести порівняльний аналіз існуючих методів анотування відео;
- розробити алгоритм анотування відео на основі одного з методів;
- реалізувати алгоритм анотування відео у вигляді програмного додатку.

2 РОЗРОБКА АЛГОРИТМУ СУМАРИЗАЦІЇ ВІДЕО

2.1 Відеосумаризація шляхом кластеризації

Огляд запропонованого методу показаний на рисунку 2.1. Тут вхід являє собою кількість відеокадрів, а вихід - це розкадровка, що складається з репрезентативних кадрів. Метод включає чотири етапи: попередня вибірка, представлення кадру, кластеризація і генерація результатів.

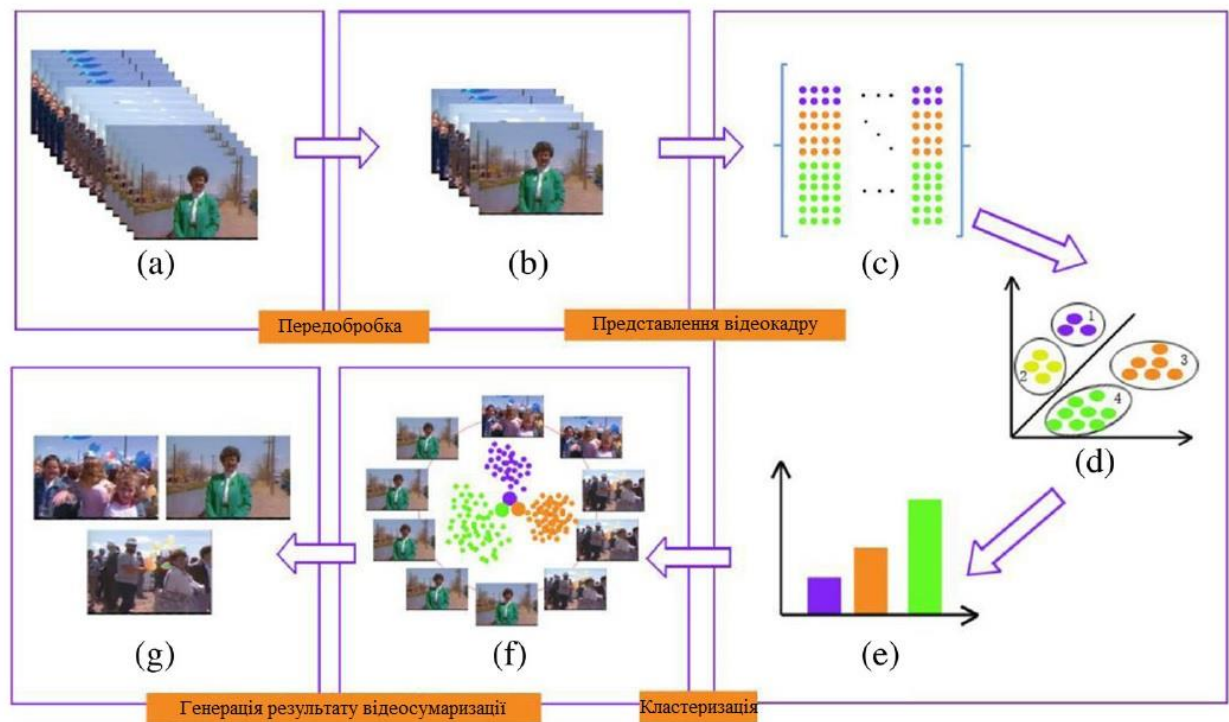


Рисунок 2.1 – Сумаризація відео шляхом кластеризації

Як правило, відео мають значну надмірність, і десятки кадрів мають однакову візуальну інформацію. Таким чином, багато пов'язаних з відео методів застосовували попередню вибірку перед обробкою, щоб зменшити кількість кадрів, що може значно скоротити обчислювальні складності і привести до економії часу. У нашому підході попередня вибірка відео також виконується для отримання кадрів-кандидатів $S = [S_1, S_2, \dots, S_M]$, де M - загальна кількість кадрів-кандидатів. В цьому етапі витяг ключового кадру виконується для всіх вхідних кадрів спочатку, а потім непотрібні кадри будуть стерті для подальшої вибірки.

Метод вилучення ключового кадру заснований на сингулярній декомпозиції (SVD). H^t представляє колірний простір Hue-Saturation-Value (HSV) поточного кадру в момент часу t . Три гістограми h_H , h_S і h_V , довжини: l_H , l_S і l_V , відповідно, побудовані для трьох колірних каналів H^t . Визначаємо функцію x^t , яка змінюється в часі як:

$$x^t = [h_H h_S h_V]. \quad (2.1)$$

Тоді довжина вектора x^t $L = l_H + l_S + l_V$. Встановлюємо матрицю $N \times L$ для кожного кадру при часі $t > N$, як описано нижче:

$$X^t = \begin{pmatrix} x^t \\ x^{t-1} \\ \dots \\ x^{t-N+1} \end{pmatrix}, \quad (2.2)$$

де $t = N, N+1, \dots, T-1, T$, N - розмір вікна, а T - кількість вхідних кадрів. X^t описує матрицю ознак під час поточного кадру в момент часу t і попередні кадри $N-1$. Після цього обчислення SVD виконується для матриці X^t , як показано в рівнянні (2.3):

$$X^t = U \Sigma V^t, \quad (2.3)$$

де U - матриця безлічі вихідних ортогональних сингулярних векторів, V^t - матриця безлічі вхідних ортогональних сингулярних векторів, а Σ - матриця сингулярних значень, діагональні елементи поміщаються в порядку убутання значущості. Наприклад, якщо q_1, q_2, \dots, q_N є діагональними елементами з Σ , то q_1 є найбільшим сингулярним значенням. r^t визначається як ранг X^t і його можна обчислити за кількістю сингулярних значень, які перевищують певний користувачем поріг τ . Якщо ранг X^t більше рангу його попередньої матриці X^{t-1} , це означає, що візуальний контент поточного кадру відрізняється від попереднього. В результаті поточний кадр буде обраний в якості ключового кадру.

Відео зазвичай має багато непотрібних кадрів. На рисунку 2.2 показаний зразок непотрібних кадрів, де (a) являє собою чорну рамку, (b) і (c) - межі шотів. Насправді, (b) є різким переходом, а (c) - поступовим переходом. Для сумаризації відео, метою якого є захоплення суті відеоконтенту, визначаємо чорні рамки і межі шотів як непотрібні кадри. Зміст цих непотрібних кадрів не повинен потрапити в результат сумаризації. Тому видалимо їх на етапі попередньої вибірки.



Рисунок 2.2 – Приклад неважливих кадрів у відео. (a) - чорна рамка, (b) і (c) - межі шотів

2.2 Представлення відеокадру

Після попередньої вибірки вибирається кількість кадрів-кандидатів $S = [S_1, S_2, \dots, S_M]$. Потім набір BoW (bag of word) [24] застосовується для подання кожного кадру-кандидата. В принципі, BoW моделювання має три етапи: витяг функції, генерація кодової книги і представлення гістограм.

Оскільки останнім часом було встановлено, що місцева функція є більш представницькою, застосовуємо одну з класичних локальних функцій - масштабне інваріантне перетворення ознак (SIFT) в якості дескриптора. Надаємо огляд генерації уявлення відеокадру в алгоритмі, який буде показано нижче.

Алгоритм описує процедуру подання відеокадру. По-перше, здобуємо SIFT функції на кожному кадрі-кандидаті, в результаті маємо велику кількість дескрипторів SIFT. По-друге, кількість типових функцій вибирається серед них як кодова книга, використовуючи алгоритм k-середніх. Після цього можемо створити гістограму для кожного кадру-кандидата відповідно до їх функціонального розподілу кодової книги. Нарешті, кожен відеокадр позначається як гістограма.

Алгоритм виглядає наступним чином. На вході маємо $S = [S_1, S_2, \dots, S_M]$: M кадр-кандидати цільового відео, L : розмір кодової книги. В кінці отримуємо $z^q = [z_1, z_2, \dots, z_L]$: представлення q -того кадра-кандидата.

1. Отримати цікаві для нас моменти з кожного кадра-кандидата S_i і зібрати всі локальні дескриптори, щоб скомпрометувати матрицю $F = [f_1, f_2, \dots, f_N]$.

2. Згенерувати кодову книгу $C = [c_1, c_2, \dots, c_N]$, виконавши алгоритм кластеризації k -середніх по F .

3. Отримати кожне представлення кадра-кандидата z^q відповідно до його локального розподілу функцій в кодової книзі:

- для кожного кадру-кандидата S_i обчислювати схожість між усіма його витягнутими дескрипторами і кожним кодовим словом c_i .

- для кожного кодового слова c_i , обчислити загальну кількість дескрипторів, які схожі з ним як z_i .

З усіма кадрами кандидатів, представлених моделлю BoW, наступний крок методу розділяє всі кадри-кандидати на кластери. Використовується метод VRHDPS заснований на HDPS, в якому не потрібно вказувати кількість кластерів, оскільки метод кластеризації HDPS покладається на ідею про те, що кластерні центри характеризуються більш високою щільністю, чим їх сусіди, і на відносно великій відстані від точок з більш високою щільністю.

Однак, деякі характеристики сумаризації відео не були розглянуті в HDPS. Тому тут розглянуто алгоритм кластеризації VRHDPS, який більш підходить до задачі сумаризації відео. У процедурі кластеризації VRHDPS необхідно обчислити дві величини для кожної точки: її локальної щільності і відстань від точок з більшою щільністю. Локальна щільність визначається як:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \quad (2.4)$$

де

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}, \quad (2.5)$$

де ρ_i позначає щільність i -ої точки, d_{ij} - відстань між точкою i і точкою j , а d_c - відстань відсічення.

В принципі, ρ_i - це число точок, близьких до d_c , до точки i . δ_i вимірюється обчислюючи мінімальну відстань між точкою i і будь-якими іншими точками з більшою щільністю:

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}). \quad (2.6)$$

Для точки з найбільшою щільністю зазвичай ідентифікуємо:

$$\delta_i = \max_{j: \rho_j > \rho_i} (d_{ij}). \quad (2.7)$$

Нарешті, кластерними центрами повинні бути ті точки з високою щільністю і аномально великими δ_i . Після того, як всі кластерні центри були розпізнані, інші точки прив'язані до того ж скупчення їх найближчого сусіда з високою щільністю. Оскільки завдання для всіх точок завершено, то процедура кластеризації закінчена. У пропонуваному методі кластеризації VRHDPS необхідно обчислити дві величини на основі відстані, тому він має низьку обчислювальну складність. Крім того, цей алгоритм кластеризації є стабільним. Як тільки вхідні відстані даних точок підтверджені, результати кластеризації більше не зміняться. В експериментальній частині результати підтверджують ці переваги.

Коли використовуємо алгоритм кластеризації VRHDPS для кластерів-кандидатів, він може бути детально розділений на три етапи. Ці три етапи описані в такий спосіб:

Крок 1. Обчислити евклідові відстані між кожними двома кадрами-кандидатами.

Крок 2. Обчислити локальну щільність відповідно до рівняння (2.4) для кожного кадра-кандидата.

Крок 3. Отримати мінімальну відстань відповідно до рівняння (2.6) для кожного кадра-кандидата.

У вищенаведених кроках використовується модель BoW для подання відеокадра і алгоритм кластеризації для вибору основних кадрів і видалення надмірності. У VRHDPS запропонована нова стратегія генерації результату сумаризації відео, яка робить алгоритм кластеризації більш здатним захоплювати суть відеоконтенту.

Насправді важливі деякі особливості відеозйомки. Наприклад, в відеосумаризації треба звернути увагу на ізольовані точки, оскільки вони представляють унікальні рамки. У методі кластеризації HDPS показано, що кластерний центр можна вибрати залежно від $\gamma = \rho * \delta$. Графік γ відсортований в порядку убутання, кластерні центри - це точки, які вище тієї точки, коли графік починає розвиватися аномально. Однак $\gamma = \rho * \delta$ вважає, що два параметри однаково важливі. В крайньому випадку точки з високою щільністю і малою мінімальною відстанню матимуть однакові γ як точки з низькою щільністю і великим мінімумом. Фактично в відеосумаризації більш важливі точки з більш низькою щільністю і відносно великою відстанню - більш підходять до сумаризації. Однак, була запропонована нова стратегія $\gamma = \alpha * (\rho * \delta) + (1 - \alpha) * \delta$ для створення кластерних центрів, які дозволяють кадрам з меншою схожістю, але великою відстанню, розглядається як результат. Крім того, α знаходиться в діапазоні від 0 до 0,5. Фактично, порівняно ефективність VRHDPS з методами HDPS при роботі з завданням сумаризації відео і результат показує, що VRHDPS більш здатний справлятися із завданнями сумаризації відео, ніж HDPS.

2.3 Кластеризація на основі представників

Для оцінки подібності двох багатовимірних часових рядів, індукованих відеорядами, можна скористатися модифікацією популярного методу динамічної часової деформації (динамічної трансформації часової шкали), який отримав широке поширення для оцінки відстані між послідовностями різної довжини. Суть його полягає в наступному. Введемо в розгляд дві багатовимірних послідовності $X = \{x(1), x(2), \dots, x(k), \dots, x(M)\}$ і $Y = \{y(1), y(2), \dots, y(l), \dots, y(N)\}$, $N \neq M$, де $x(k)$, $y(l)$ або вектори $x(k) = (x_1(k), x_2(k), \dots, x_i(k), \dots, x_n(k))$, $y(l) = (y_1(l), y_2(l), \dots, y_i(l), \dots, y_n(l))$ в деякому просторі $\Omega \subset \mathbb{R}^n$ ознак $x(k) = \{x_{i_1 i_2}(k)\}$, $y(l) = \{y_{i_1 i_2}(l)\}$ або матриці (фрагменти відеокадрів) в просторі сигналів $x(k) = \{x_{i_1 i_2}(k)\}$, $y(l) = \{y_{i_1 i_2}(l)\}$, $x(k), y(l) \in \mathbb{R}^{h \times v}$.

Далі в розгляд вводиться матриця $(N \times M)$ відстаней в прийнятій локальній (поелементній) метриці з елементами $d(x(k), y(l))$, $k = 1, 2, \dots, N$, $l = 1, 2, \dots, M$ між усіма елементами аналізованих послідовностей. На підставі цієї матриці будується «викривлений шлях» у вигляді ряду відстаней

$W = \{w_1, w_2, \dots, w_q, \dots, w_L\}$, $w_q = d(x(k), y(l))_q$, $\max\{N, M\} \leq L \leq M + N - 1$, $q = 1, 2, \dots, L$, який і визначає подібність між і на основі акумулятивної відстані (2.8):

$$D(k, l) = d(x(k), y(l)) + \min\{D(k, l-1), D(k-1, l), D(k-1, l-1)\}. \quad (2.8)$$

Результуючий «викривлений шлях» може бути представлений у вигляді $DTW(X, Y) = D(N, M)$, який, взагалі кажучи, являє собою міру подібності, але не метрику.

Як відстань між елементами X і Y зазвичай використовується евклідова метрика:

$$d(x(k), y(l)) = \|x(k) - y(l)\|_2, \quad (2.9)$$

яка при обробці фрагментів відеокадрів набуває вигляду метрики норми Фробеніуса:

$$d(x(k), y(l)) = (Sp(x(k) - y(l))(x(k) - y(l))^T)^{\frac{1}{2}}. \quad (2.10)$$

У ситуаціях, коли оброблювані дані зашумлені і мають викиди, доцільно використання манхеттенської метрики, що володіє робастними властивостями.

При цьому (2.9) відповідає вираз:

$$d(x(k), y(l)) = \sum_{i=1}^n |x_i(k) - y_i(l)|, \quad (2.11)$$

а для (2.10) маємо:

$$d(x(k), y(l)) = \sum_{i_1=1}^n \sum_{i_2=1}^v |x_{i_1}(k) - y_{i_2}(l)|. \quad (2.12)$$

Звернемося тепер власне до задачі кластеризації. Введемо в розгляд набір багатовимірних часових рядів $X_1, X_2, \dots, X_q, \dots, X_Q$, які повинні бути згруповані в m кластерів. Передбачається, що кожен з кластерів містить різну кількість спостережень $N_1, N_2, \dots, N_q, \dots, N_Q$, відповідно. Безпосередня кластеризація вихідних послідовностей представляється малоефективною, оскільки розрахунок (2.8) на основі динамічного програмування на довгих лавах, продукованих відеоданими, пов'язаний з високою

обчислювальною складністю. Крім того, в опрацюванні послідовності, як правило, нестационарні, тобто різні сегменти можуть ставитися до різних класів.

У подібних ситуаціях можна скористатися «віконним» підходом, коли кожен ряд розбивається на ділянок-«вікон», в результаті чого виникає набір нових рядів, які розглядаються далі як незалежні послідовності, що підлягають кластеризації. При цьому, якщо вікна одного ряду потрапляють в різні кластери, що свідчить про нестационарності вихідної послідовності, і отримані сегменти можуть класифікуватися як незалежні один від одного сигнали.

У подібних ситуаціях можна скористатися «віконним» підходом, коли кожен ряд X_q , $q=1,2,\dots,Q$ розбивається P на ділянок-«вікон», в результаті чого виникає набір нових рядів $WX_{q1}, X_{q2}, \dots, WX_{qp}, \dots, WX_{qp}$, $p=1,2,\dots,P$, які розглядаються далі як незалежні послідовності, що підлягають кластеризації. При цьому, якщо вікна WX_{qp} одного ряду X_q потрапляють в різні кластери, що свідчить про нестационарності вихідної послідовності, і отримані сегменти можуть класифікуватися як незалежні один від одного сигнали.

Таким чином, в розгляд вводиться QP сигналів WX_{qp} , кожен з яких містить N_{pq} відліків, а кінцевою метою є розбиття цих рядів в режимі самонавчання на однорідні у сенсі (2.8) - (2.12) класи.

Серед відомих алгоритмів кластеризації найбільш широкого поширення набули методи, засновані на прототипах-центроїдах, завдяки простоті обчислювальних моделей і наочності отриманих результатів. Однак ці методи мало придатні до кластеризації багатовимірних часових рядів, які продукуються відеопотоками, оскільки оброблювані образи у відомих підходах мають однакову розмірність, а в нашому випадку основний фактор - різні довжини ряду: WX_{qp} має N_{pq} елементів.

У подібних ситуаціях кращими можуть виявитися підходи, засновані на використанні зразків-представників, де замість прототипу-центроїда, що обчислюється, вибирається один з векторів (матриць) - образ, наявних в оброблюваній вибірці $WX_{11}, \dots, X_{1p}, \dots, WX_{21}, \dots, WX_{qp}, \dots, WX_{QP}$, що містить QP часових сегментів. Розглянемо процедуру, яка базується на модифікації.

Робота алгоритму починається з вибору початкових представників $WX_{qp}^1(0), WX_{qp}^2(0), \dots, WX_{qp}^m(0)$, де m - задане число кластерів. Як $WX_{qp}^1(0)$ вибирається елемент вибірки найбільш віддалений від усіх інших, тобто

$$DTW(WX_{qp}^1(0), WX_{qp}(0)) > DTW(WX_{rs}, WX_{qp}), \forall q, p \in \{1, 2, \dots, Q\}, \\ \forall r, s \in \{1, 2, \dots, P\}.$$

Далі $WX_{qp}^1(0)$ тимчасово виключається з вибірки і вибирається другий представник, як і раніше, найбільш віддалений від залишених у вибірці елементів, тобто

$$DTW(WX_{qp}^1(0), WX_{qp}(0)) > DTW(WX_{qp}^2(0), WX_{qp}(0)) > DTW(WX_{rs}, WX_{qp}). \quad (2.13)$$

Ця процедура вибору повторюється до формування початкових образів: $DTW(WX_{qp}^1(0), WX_{qp}(0)) > \dots > DTW(WX_{qp}^m(0), WX_{qp}(0)) > DTW(WX_{rs}, WX_{qp})$.

Дана процедура вибору початкових образів ефективна, якщо вихідні дані не містять аномальних викидів. В іншому випадку початкові представники можуть бути обрані випадковим чином, подібно до того, як це трапляється в жадних алгоритмах кластеризації, наприклад, в класичному алгоритмі k-середніх. Стосовно до обробки відеопотоків подібні викиди можуть являти собою, наприклад, часові відрізки вставленої рекламної продукції, кластеризація навколо яких семантично безглузда. Інакше кажучи, попередня обробка структурованої вибірки $WX_{11}, \dots, X_{1p}, \dots, WX_{21}, \dots, WX_{qp}, \dots, WX_{Qp}$ набуває особливого значення.

На другому етапі залишені $QP - m$ образів розподіляються по кластерам, згідно співвідношення:

$$WX_{qp} \in Cl_j, \quad \text{якщо}$$

$$DTW(WX_{qp}, WX_{qp}^j(0)) < DTW(WX_{qp}, WX_{qp}^l(0)), \forall j \neq l \in \{1, 2, \dots, m\}.$$

Таким чином, всі наявні образи групуються в околицях кожного з початкових представників $WX_{qp}^j(0)$.

У кожній з сформованих груп на третьому етапі розраховується новий представник, в якості якого вибирається спостереження з мінімальною сумарною відстанню до всіх точок вихідного кластера. Для уточненого представника справедливі співвідношення:

$$WX_{qp} = WX_{qp}^j(1), \text{якщо}$$

$$\sum DTW(WX_{qp} \in Cl_j, WX_{qp}^j(1)) < \sum DTW(WX_{qp} \in Cl_j, WX_{rs} \in Cl_j)$$

2.4 Алгоритм Dynamic Time Warping

Алгоритм динамічного часового деформування (DTW) є добре відомим алгоритмом в багатьох областях. Використовується в багатьох областях: інтелектуальний аналіз даних і кластеризація часових рядів (пошук по базах даних часових рядів), комп'ютерний зір і комп'ютерна анімація, вирівнювання послідовностей, хімія, музика і обробка сигналів.

Алгоритм DTW заслужив свою популярність, будучи надзвичайно ефективним, оскільки міра подібності серії часу, яка мінімізує ефекти зсуву і спотворення в часі, дозволяючи «еластичну» трансформацію часових рядів, щоб виявити аналогічні форми з різними фазами. З огляду на два тимчасові ряди $X = (x_1, x_2, \dots, x_N), N \in \mathbb{N}$ і $Y = (y_1, y_2, \dots, y_M), M \in \mathbb{N}$, представлений послідовності значень (або кривих, представлених послідовностями вершин) DTW дає оптимальне рішення в $O(MN)$ часу, яке може бути покращено далі за допомогою різних методів.

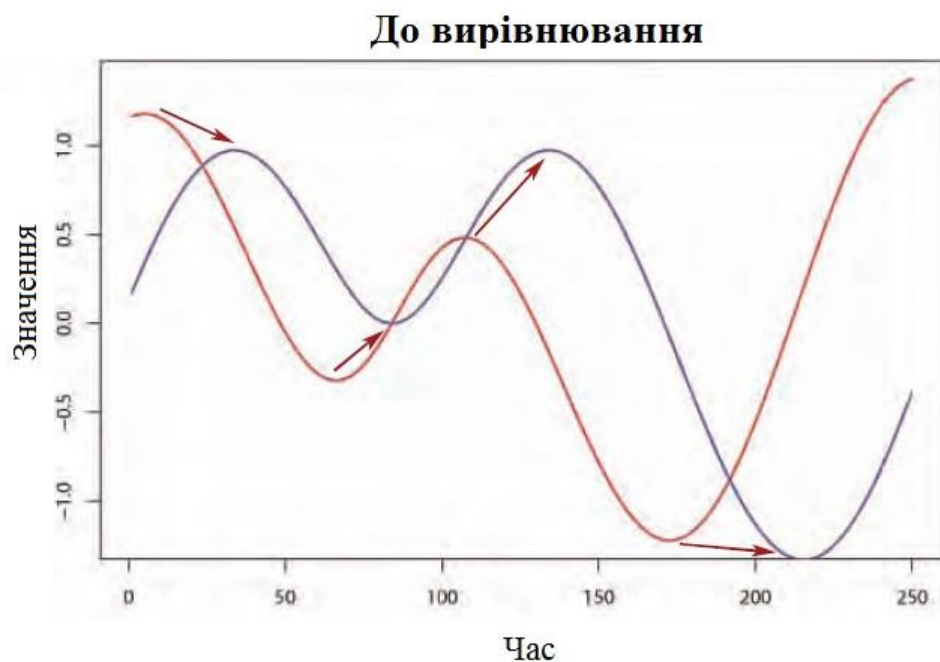


Рисунок 2.3 – Неопрацьований часовий ряд. Стрілки вказують на бажане вирівнювання

Єдиним обмеженням на послідовності даних полягає в тому, що вони повинні бути відібрані на рівновіддалені точки в часі (цю проблему можна вирішити шляхом повторної вибірки). Якщо послідовності приймають значення з будь-якого простору функцій Φ ніж

в порядку порівняння двох різних послідовностей $X, Y \in \Phi$ потрібно використовувати локальну відстань, яка визначається як:

$$d: \Phi \times \Phi \rightarrow \mathbb{R} \geq 0. \quad (2.14)$$

Інтуїтивно d має невелике значення, коли послідовності схожі і велике значення якщо вони різні. Оскільки алгоритм динамічного програмування лежить в ядрі, DTW зазвичай називають цю функцію відстані «функцією вартості», і завдання оптимального вирівнювання послідовностей стає завданням впорядкування всіх точок послідовності шляхом мінімізації функції вартості (або відстані).

Алгоритм починається з побудови матриці відстаней $C \in \mathbb{R}^{N \times M}$, що представляє всі попарні відстані між X і Y . Ця матриця відстаней, називається локальною матрицею витрат для вирівнювання двох послідовностей X і Y :

$$C_l \in \mathbb{R}^{N \times M}: c_{i,j} = \|x_i - y_j\|, i \in [1:N], j \in [1:M]. \quad (2.15)$$

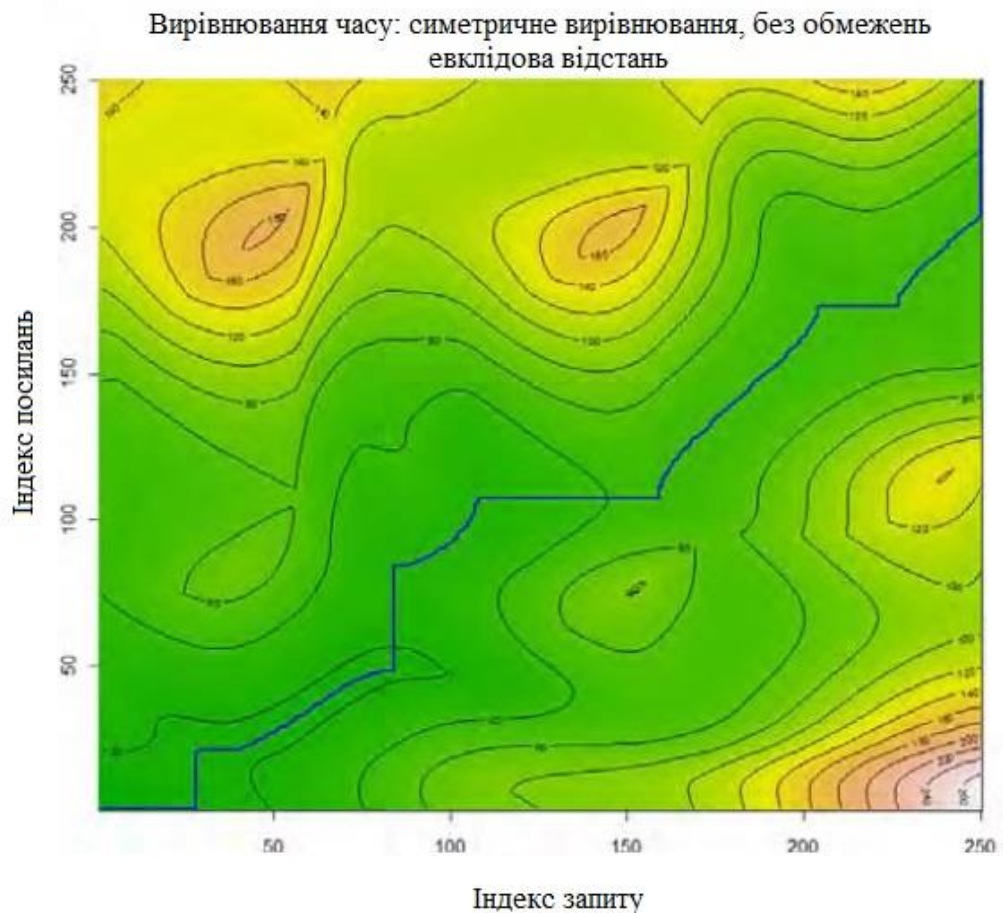


Рисунок 2.4 - Вирівнювання часових рядів, схема розрахунку вартості.

Як тільки побудована локальна матриця витрат, алгоритм вирівнює шлях який проходить через «недорогі» області – «долини» на матриці витрат (рис. 2.4).

Цей шлях вирівнювання (або шлях деформації) визначає відповідність елементу $x_i \in X$ рівним $y_j \in Y$ після границі умови, яка присвоює перший і останній елементи X і Y один одному (рис 2.5).

Формально кажучи, шлях вирівнювання, побудований DTW, являє собою послідовність точки $p = (p_1, p_2, \dots, p_K)$ з $p_l = (p_i, p_j) \in [1:N] \times [1:M]$ для $l \in [1:K]$ який повинен відповідати таким критеріям:

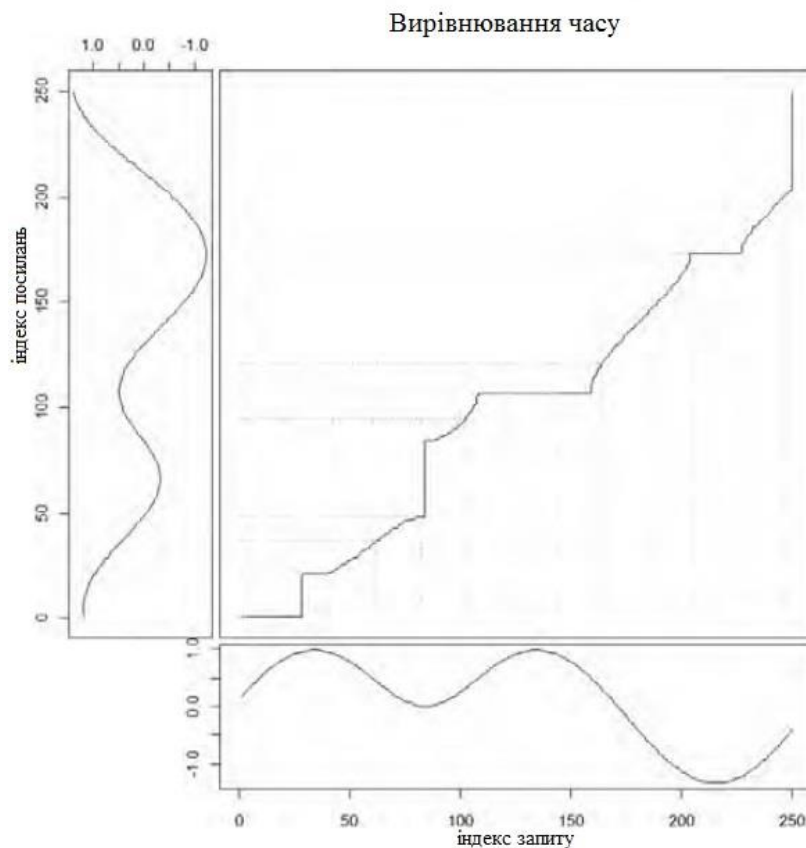


Рисунок 2.5 - Оптимальний часовий ряд вирівнювання траєкторії, показаний на рисунку 2.3

1) Гранична умова: $p_1 = (1,1)$ і $p_k = (M, N)$ початкова і кінцева точки шляху деформації повинні бути першими і останніми точками вирівняних послідовностей.

2) Умова монотонності: $n_1 \leq n_2 \leq \dots \leq n_k$ і $m_1 \leq m_2 \leq \dots \leq m_k$. Ця умова зберігає порядок часу в точках.

3) Умова розміру кроку: цей критерій обмежує шлях деформації від довгих переходів (зрушення в часі) при вирівнюванні послідовностей. Хоча ця умова буде більш

детально обговорюватися далі, зараз буде використовуватися умова основного кроку, сформульована як $p_{l+1} - p_l \in \{(1,1), (1,0), (0,1)\}$.

Функція вартості, пов'язана з траєкторією деформування, спектру локальної матриці витрат (який представляє всі попарні відстані) буде:

$$c_p(X, Y) = \sum_{l=1}^L c(x_{n_l}, y_{m_l}). \quad (2.16)$$

Шлях деформації, який має мінімальну вартість, пов'язану з вирівнюванням, називається оптимальним шляхом деформації. Будемо називати цей шлях P^* .

Дотримуючись визначенню оптимального шляху поділу, щоб визначити його, необхідно перевірити всі можливі шляхи деформації між X і Y , які можуть мати обчислювальну складність через експоненціальне зростання числа оптимальних шляхів, так як довжини X і Y лінійно зростає. Щоб подолати цей виклик, DTW використовує алгоритм, заснований на динамічному програмуванні зі складністю тільки $O(MN)$.

Компонент динамічного програмування алгоритму DTW використовує DTW функцію відстані:

$$DTW(X, Y) = c_{p^*}(X, Y) = \min \{c_p(X, Y), p \in P^{N \times M}\}, \quad (2.17)$$

де $P^{N \times M}$ - безліч всіх можливих шляхів деформації і будує матрицю сукупної вартості або глобальну матрицю витрат D , яка визначається наступним чином:

$$1) \text{ Перший рядок: } D(1, j) = \sum_{k=1}^j c(x_1, y_k), j \in [1, M].$$

$$2) \text{ Перша колонка: } D(i, 1) = \sum_{k=1}^i c(x_k, y_1), i \in [1, N].$$

3) Інші елементи:

$$D(i, j) = \min \{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + c(x_i, y_j), i \in [1, N], j \in [1, M].$$

Часова вартість побудови цієї матриці дорівнює $O(NM)$.

Щоб підвищити продуктивність і налаштувати чутливість «наївного» DTW були запропоновані різні модифікації цього методу. Далі будуть описані основні модифікації, такі як умови розміру кроку, етап зважування і глобальні обмеження шляху.

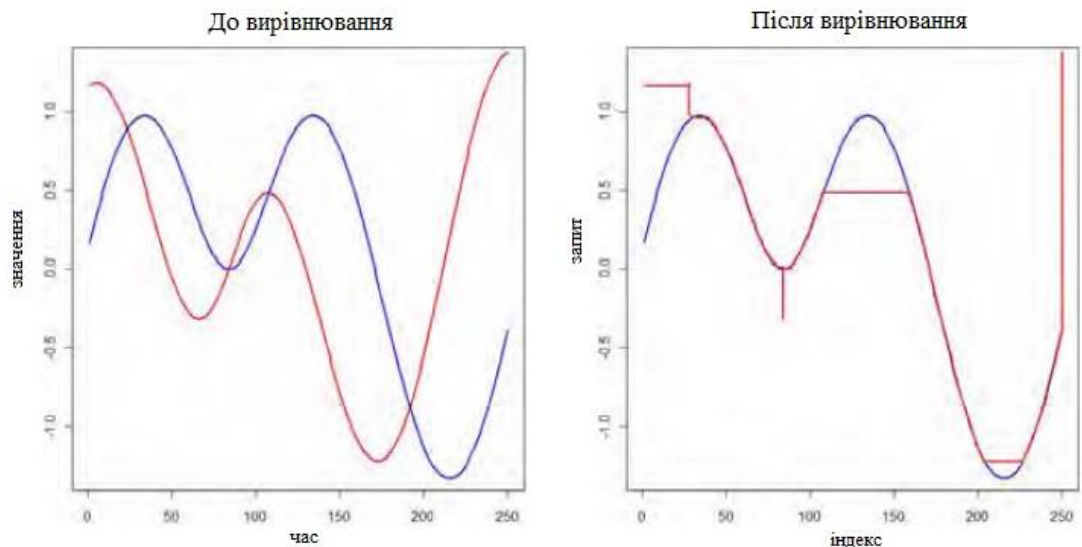


Рисунок 2.6 - Оптимальне вирівнювання часових рядів по рисунку 2.3.

Функція кроку (обмеження нахилу). Коли між часовими рядами немає відмінностей, шлях деформації співпадає з діагональною лінією $i = j$, але коли відмінності між тимчасовими рядами збільшуються, шлях деформації відхиляється більше від діагональної лінії шляхом зіставлення аналогічної осі часу флуктуації.

У той час як DTW знаходить оптимальне вирівнювання часового ряду, іноді він, як правило, створює нереалістичну відповідність за допомогою декількох часових функцій, поєднуючи дуже короткі функції з одним із серії до довгих функцій у другому часовому ряду.

Щоб уникнути таких явищ, шлях деформації необхідно обмежити на кожному кроці. Ці обмеження будуть між декількома послідовними точками на шляху деформації.

Наприклад, після переміщення в однаковому напрямку горизонтально для k послідовних точок деформований шлях не дозволяє продовжувати в тому ж напрямку до кроку l в діагональному напрямку (рис 2.7).

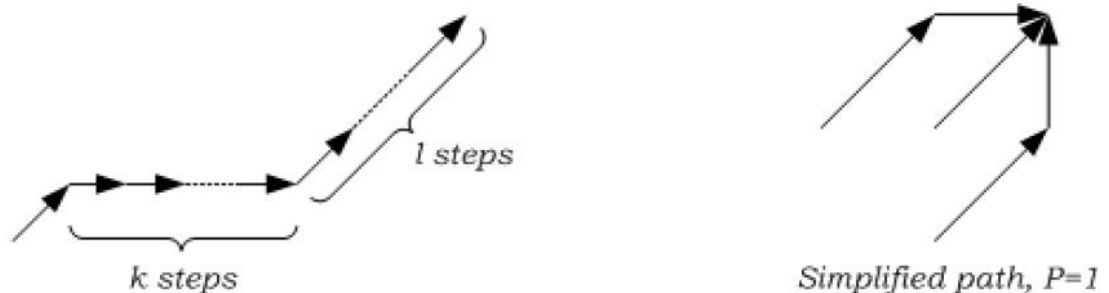


Рисунок 2.7 – Обмеження функції

Як було показано раніше, обчислювальна вартість алгоритму DTW дорівнює $O(NM)$ і алгоритм вимагає зберігання для двох матриць розміру $N \times M$.

З метою поліпшення обчислювальних витрат і оптимізації чутливості DTW до обмежень функції кроку були введені глобальні обмеження.

Смуга Сакое-Чіба з'явилася в паралелограмі Ітакури як глобальне обмеження шляху, показана на рисунку 2.8, так і визначення безлічі точок доступних для вирівнювання DTW тільки з незатінених областей. Смуга Сакое-Чіба проходить по головній діагоналі ($i = j$) і обмежує діапазон деформації, який визначається шириною $R \in \mathbb{N}$ (рис.2.9)

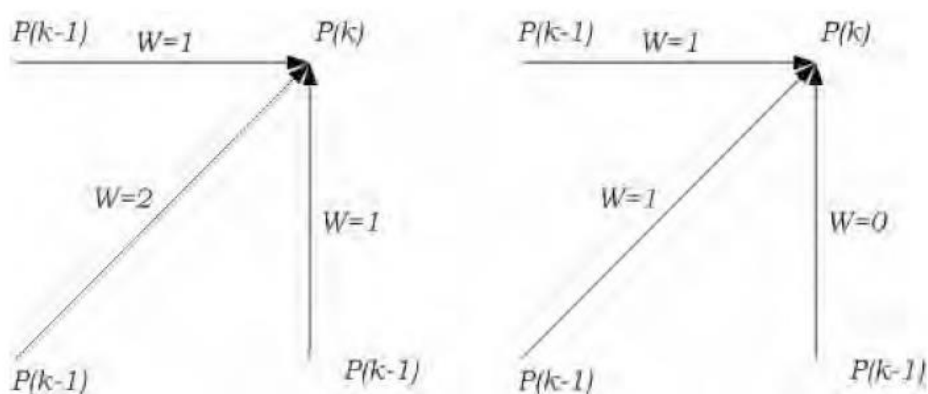


Рисунок 2.8 - Вагові коефіцієнти $w(k)$ для симетричних (ліворуч) і асиметричних (праворуч) форм

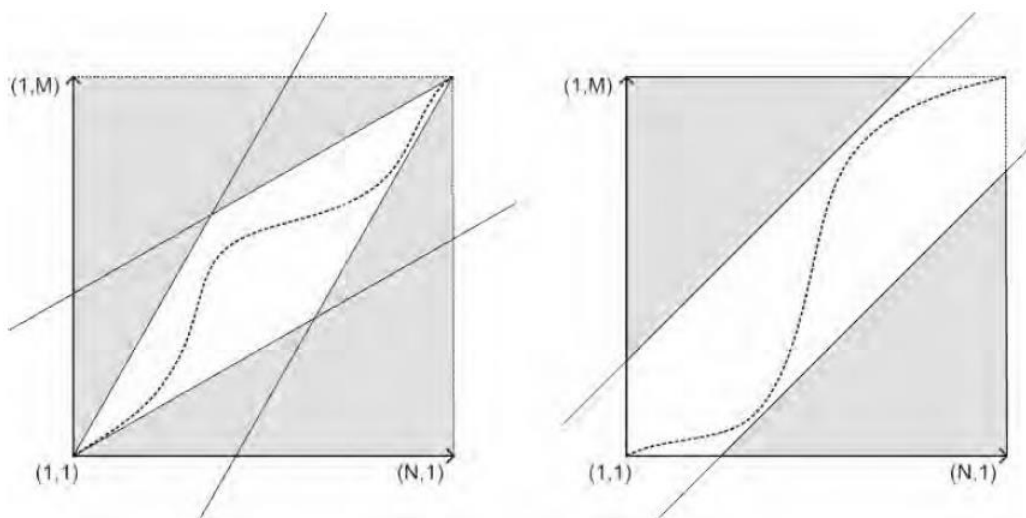


Рисунок 2.9 - Глобальні обмеження шляху, ліворуч: паралелограм Ітакури, праворуч: група Сакое-Чіба

Параметр R задає довжину вертикального вектора від основної діагоналі до верхньої межі смуги і симетрично довжину горизонтального вектора від головної діагоналі до нижньої межі смуги:

$$|i(k) - m| \leq R, |j(k) - m| \leq R, \quad (2.18)$$

де m - відповідна координата на головній діагоналі k і показує k -ту точку траєкторії деформації. Паралелограм Ітакури також обмежує діапазон деформації і за допомогою параметра $S \in \mathbb{R}_{\geq 1}$.

Хоча і було показано скорочення обчислювального алгоритму DTW, іншим популярним методом прискорення DTW є апроксимація часового ряду за допомогою масштабування.

Масштабування також відомо як зменшення розмірності і цільове зменшення довжин N і M вхідних часових рядів через укрупнення. Це може бути досягнуто шляхом понижування дискретизації нижнього проходу або лінійним наближенням. Підхід полягає в тому, що після досягнення певного рівня укрупнення вирівнювання стає абсолютно марним.

Добре відомо, що прямий пошук подібності в часових рядах баз даних обчислювально коштовний. Як засіб для швидкого звернення до запитів бази даних була придумана «індексація часових рядів». По суті будь-який з часових рядів довжини n можна розглядати як кортеж в n -вимірному просторі. Як тільки такий простір індексується, пошук зазвичай займає лінійний час, виконуючи прості порівняння.

У той час як «прокляття розмірності» робить пряме індексування цього простору, ідея полягає в поліпшенні індексування за допомогою застосування деякого методу зменшення розмірності, який перетворює довгі часові ряди n елементів в простір з k розмірами, де $k \ll n$. Розкладання SVD, перетворення Фур'є (і аналогічне дискретне косинусне перетворення), розкладання вейвлетів, багатовимірне масштабування, методи випадкової проекції, FastMap (і інші) є часто використовуваними та робочими методами.

2.5 Метрики оцінки для сумаризації

Стислість, охоплення контенту і видалення надмірності є ключовими аспектами якісного методу відеосумаризації. Для відео стислість відноситься до компактного подання його візуального змісту. Коротке представлення гарантує, що вміст відео добре представлено без будь-яких непотрібних деталей. Заочне забезпечення гарантує, що

користувач зможе переглядати відеопідсумок в найменший можливий час і зрозуміти суть відео. Можна швидко вирішити, чи є відеопідсумок корисним чи ні. З іншого боку, охоплення контенту відноситься до аспекту уявлення значного вмісту відео, так що важливі частини можуть не враховуватися.

Було застосовано стандартизовану методику вирізу і аналіз тимчасового графа для максимізації охоплення контенту і зменшення дублікатів відеоданих. Аналогічно, для автоматичної генерації використовувався механізм триангуляції Делоне. Це було плідно, оскільки людська взаємодія не була потрібна, а значимий візуальний контент був досягнутий за меншу кількість кадрів. Вони змодельовали кожен кадр відео як графік просторово-часових відносин для визначення того, яка сцена сприятлива для скімінгу. Це було зроблено після визначення меж сцени. Також була отримана загальна довжина кожної відеосцени [32].

Це дало значне охоплення відеоконтенту. Було проведено три механізми оцінки для визначення обсягу автоматичних зведень з аудіо, слайдами і відео-презентаціями. Результати показали, що використання ключових моментів в скімінгу було досить ефективним. Три способи були застосовані для отримання відеопідсумку. Спочатку відео було сегментоване для вибору кандидатів. Потім була дана функція несхідства, що позначає просторово-часове відношення. Після цього кандидат-шот був складений в спрямований граф і був отриманий найдовший шлях. Це забезпечило хороше охоплення контенту в підсумку [33].

Було застосовано розкладання сингулярного значення для генерації підсумку відео. Надмірність була видалена і збережено значний контент. Вони зобразили методи для автоматичної генерації графічних відеороликів, які мали велике охоплення контенту. З рамок кадру створюється мінімальне сполучне дерево (MST). Адаптивний поріг розраховується окремо для кожного кадру на основі середнього і стандартного відхилення краю ваги МСД. Щільність вузла - це кількість кадрів, що лежать всередині диска, радіус який дорівнює обчисленому порогу. Жадібний метод використовується для вибору фреймів зі списку за допомогою максимальної щільності. Рамки, представлені вибраними ключовими кадрами, виключаються зі списку. Це забезпечує те, що найбільш підходящі представники обрані в якості ключових кадрів. Легко побачити, що вибрані ключові кадри забезпечують повне охоплення кадру [25].

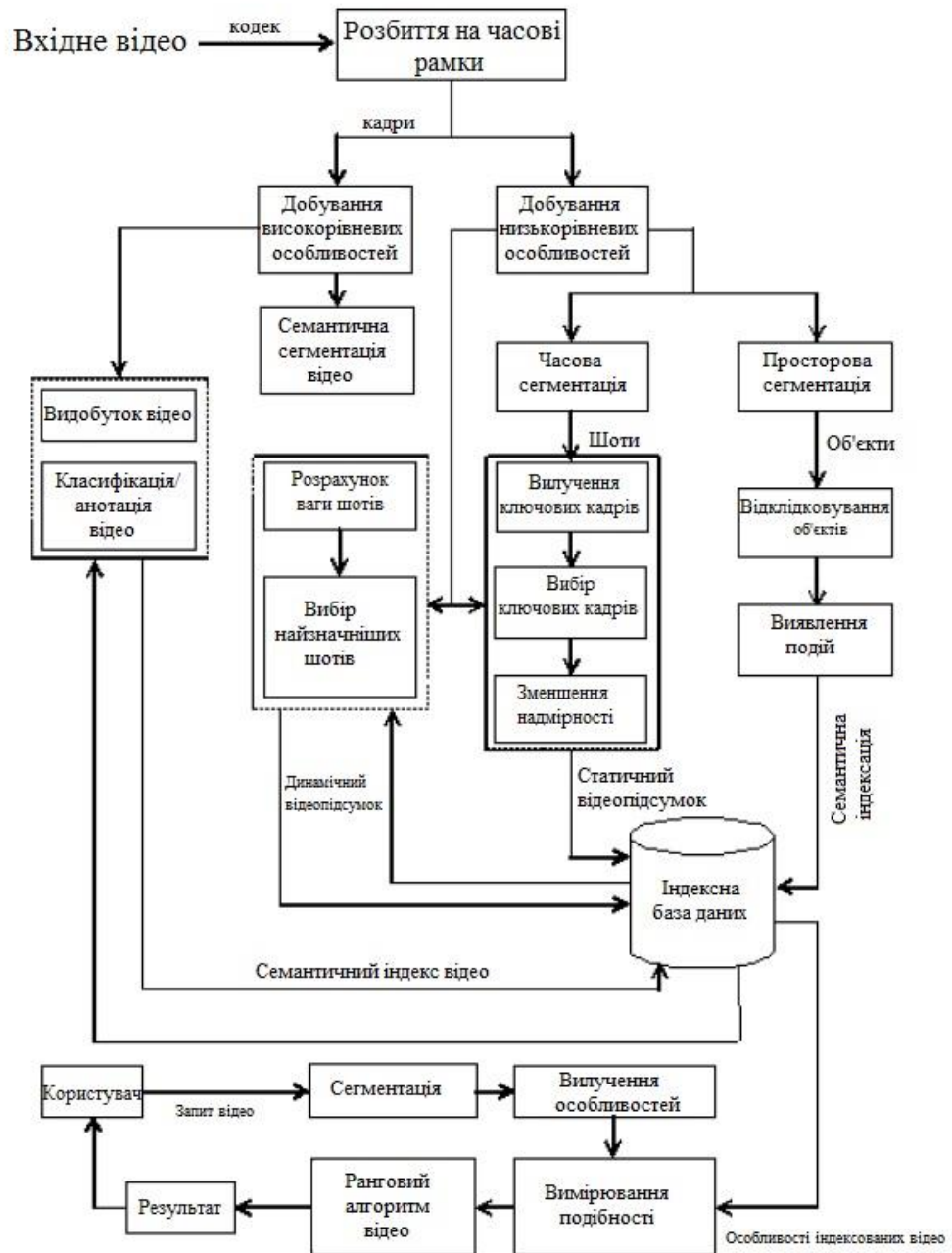


Рисунок 2.10 - Кроки для сумаризації, індексування та вилучення відео

Було виконано охоплення, яке визначене як кількість візуально схожих кадрів, представлених обраним ключовим кадром. Отже, охоплення може бути обчислене за такою формулою:

$$\text{покриття} = \frac{\text{кількість_представлених_кадрів}}{\text{загальна_кількість_кадрів}} \quad (2.19)$$

Також було запропоновано охоплення на основі функцій, структурованих кадром з унікальних пул точок, створених з композиційних кадрів. Спочатку всі функції є частиною набору $K_{uncovered}$. Охоплення кадру обчислюється за формулою:

$$C = \eta(K_{uncovered} \cap FP_i). \quad (2.20)$$

Надмірність кадру визначається:

$$R = \eta(K_{covered} \cap FP_i). \quad (2.21)$$

де $\eta(X)$ - потужність безлічі X , а FP_i - безліч точок ознак, які містять не менше одного кадру, аналогічного кадру-кандидату. Таким чином, охоплення - це ще одна метрика, яка показує якість відеосумаризації.

Деякі адаптували підхід вирівнювання послідовностей, названий Smith-Waterman алгоритмом, щоб видалити візуальну надмірність. Це було зроблено шляхом виявлення аналогічного відеосегмента. Деякі використовували два різних механізми для скорочення надмірності. По-перше, підхід семантичної кластеризації на основі знімків був використаний для видалення надлишкових відеопослідовностей. Друга методика була заснована на іншому алгоритмі кластеризації - ієрархічній агломерації (НА). Обидві методики добре зарекомендували себе при усуненні надмірності. Дехто використовував механізм ієрархічної агломерації для виявлення дворівневої надмірності. Також деякі використовували метрику надмірності, щоб визначити, скільки надлишку присутні в сумаризованому відеоконтенті. Більш того, сумаризація на основі розкладання сингулярного значення (SVD) була представлена в деяких роботах, щоб визначити ступінь візуальних змін. Видалення надмірності з використанням дескрипторів SURF і GIST також застосовується в [25].

Оцінка системи сумаризації відео є важливим завданням, оскільки вона дозволяє використовувати різні суматори для порівняння. Механізм оцінки також вказує на ефективність системи сумаризації. Спочатку вихідні дані, отримані від відеооб'єктиву, називаються «Згенерована система» (SGS). Щоб оцінити якість цього відеопідсумку, SGS порівнюється з «основною істиною». Це очевидно, що істина повинна бути якомога ближче до людського сприйняття. Отже, основна істина створена набором користувачів і називається «Підсумок користувача». Підсумок, згенерований користувачами, об'єднується разом для формування підсумку користувача (FUS). Підсумована інформація

користувача (US) створюється групою користувачів. Ступінь перекриття між FUS і SGS відображає ефективність відеопідсумку. Відгук (recall) та точність (precision) обчислюється таким чином:

$$recall = \frac{\eta(FUS \cap SGS)}{\eta(FUS)}, \quad (2.22)$$

$$precision = \frac{\eta(FUS \cap SGS)}{\eta(SGS)}, \quad (2.23)$$

де FUS - набір кадрів в підсумку користувачів; SGS - набір фреймів в підсумку, який згенерувала система, η - кількісний числівник безлічі X .

Точність і відгук, пов'язані з системою, є суперечливими з тієї причини, що одна має тенденцію до зменшення, коли інша збільшується. Конструкція системи повинна бути такою, щоб вона максимізувала і те й інше. Оцінка F1 є кульмінацією як відгуку, так і точності і обчислюється як середнє значення гармонік з двох оцінок. Вона забезпечує узгоджену міру для визначення загальної ефективності інформації пошукової системи. Для обчислення оцінки F1 використовується такий вираз:

$$F_1 = 2 \frac{precision * recall}{precision + recall}. \quad (2.24)$$

Оцінка F1 змінюється в діапазоні $[0, 1]$, де оцінка 1 вказує на кращу ефективність системи.

Три нових показника були запропоновані (Mundur, 2006) для визначення якості сумаризації. Фактор значущості означає важливість вмісту, що представляє собою кластер кадрів. Значення i -го кластера дається як:

$$Фактор_значущості(i) = \frac{C_i}{\sum_{j=1}^k C_j}, \quad (2.25)$$

де C_i - загальне число кадрів в i -м кластері, k - загальна кількість кластерів.

Коефіцієнт перекриття визначає загальну значимість кадрів, які перекриваються, знайдених в двох відеопідсумках. Іншими словами, обчислюємо сукупну значимість тих

кластерів, які мають загальний ключовий кадр із загальним підсумком істини. Це важливий показник для порівняння двох відеопідсумків. Цей коефіцієнт обчислюється як:

$$\text{Коефіцієнт_перекриття} = \frac{\sum_{p \in \text{загальні_кластери_ключових_кадрів}} C_p}{\sum_{j=1}^k C_j}, \quad (2.26)$$

Більш високе значення коефіцієнта перекриття означає кращий репрезентативний підсумок щодо істини. Коефіцієнт стиснення для відео позначає розмір зведення по відношенню до оригінального розміру відео. Він визначається як:

$$\text{Коефіцієнт_стиснення} = \frac{\text{Кількість_ключових_кадрів_підсумку}}{\text{Загальна_кількість_ключових_кадрів}}. \quad (2.27)$$

3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ДЛЯ АНОТУВАННЯ ВІДЕО

3.1 Обґрунтування вибору програмного середовища

Для якісної програмної реалізації додатку було прийнято рішення використовувати дві мови програмування: JavaScript та C#. Були обрані комбінації технологій, які доповнюють ці мови програмування. Далі проаналізуємо переваги та недоліки використання тієї чи іншої мови програмування.

3.1.1 Огляд технологій при розробці проекту на C#

C# - це об'єктно-орієнтована мова програмування, яка дозволяє розробнику створювати різноманітні безпечні і надійні додатки, які працюють на платформі .NET Framework. Можна використовувати C# для створення клієнтських додатків Windows, веб-служб XML, розподілених компонентів, клієнт-серверних додатків, додатків баз даних і багато іншого.

Visual C# надає вдосконалений редактор коду, зручні дизайнери користувальницького інтерфейсу, інтегрований відладчик і безліч інших інструментів, щоб спростити розробку додатків на основі мови C# і .NET Framework.

Синтаксис у C# дуже виразний, але він також простий і легкий у вивченні, тому його можна швидко засвоїти для виконання роботи. Синтаксис C# буде миттєво впізнаваний будь-якою людиною, яка знайома з іншими популярними мовами програмування.

Розробник, який знає Java, C++, як правило, може почати продуктивно працювати і на C# протягом дуже короткого часу. Синтаксис C# спрощує багато складностей C++ і надає потужні функції, такі як типи значень з нульовим значенням, перерахування, делегати, лямбда-вирази і прямий доступ до пам'яті, які не знайдені в Java. C# підтримує загальні методи і типи, які забезпечують підвищену безпеку і продуктивність типів, і ітератори, які дозволяють розробникам класів-колекцій визначати призначені для користувача ітераційні поведінки, які прості у використанні клієнтським кодом. Вирази LINQ (Language-Integrated Query) роблять строго типізований запит першокласною мовною конструкцією.

Як об'єктно-орієнтована мова C# підтримує концепції інкапсуляції, успадкування та поліморфізму. Всі змінні і методи, включаючи метод Main, точку входу додатку,

інкапсулюються всередині визначень класів. Клас може успадковувати безпосередньо від одного батьківського класу, але він може реалізувати будь-яку кількість інтерфейсів.

Методи, які скасовують віртуальні методи в батьківському класі, вимагають ключового слова перевизначення як спосіб уникнути випадкового перевизначення. У C# структура схожа на легкий клас; це тип, що виділяється стеком, який може реалізувати інтерфейси, але не підтримує спадкування.

На додаток до цих базових об'єктно-орієнтованих принципів, C# полегшує розробку програмних компонентів за допомогою декількох інноваційних мовних конструкцій, включаючи наступні:

- інкапсульовані сигнатури методів називаються делегатами, які включають повідомлення про події типу безпеки;
- властивості, які служать в якості аксесорів для приватних змінних-членів;
- атрибути, які надають декларативні метадані про типи під час виконання;
- коментарі до документації по вбудованим XML документах;
- Language-Integrated Query (LINQ), який надає вбудовані можливості запитів в безлічі джерел даних.

Якщо потрібно взаємодіяти з іншим програмним забезпеченням, таким як COM-об'єкти або власні DLL-бібліотеки Win32, можна зробити це на C# за допомогою процесу під назвою «Interop». Цей процес дозволяє програмам C# робити майже все, що може зробити власний додаток на C++. C# навіть підтримує покажчики і концепцію «небезпечного» коду для тих випадків, коли прямий доступ до пам'яті є абсолютно критичним.

Процес складання C# простий в порівнянні з C і C++ і більш гнучкий, ніж в Java. Окремих файлів заголовків немає, і не потрібно, щоб методи і типи оголошувалися в певному порядку. Вихідний файл C# може визначати будь-яку кількість класів, структур, інтерфейсів і подій.

C# - програми працюють на платформі .NET Framework, що є невід'ємним компонентом Windows, який включає в себе віртуальну систему виконання, звану CLR (common language runtime), і уніфікований набір бібліотек класів.

CLR - це комерційна реалізація Microsoft загальноприйнятою мовною інфраструктурою (CLI) міжнародного стандарту, який є основою для створення середовищ виконання і розробки, в яких мови і бібліотеки працюють разом.

Вихідний код, написаний на C#, компілюється в проміжну мову (IL), яка відповідає специфікації CLI. Код IL і його ресурси, такі як растрові зображення і рядки, зберігаються

на диску в виконуваному файлі, як правило, з розширенням .exe або .dll. Збірка містить маніфест, який надає інформацію про типи збірки, версії і вимоги безпеки.

Коли програма C# виконується, збірка завантажується в середу CLR, яка може виконувати різні дії на основі інформації в маніфесті. Потім, якщо вимоги безпеки дотримані, CLR виконує компіляцію якраз в часі для перетворення IL-коду в власні машинні інструкції. CLR також надає інші послуги, пов'язані з автоматичним збором сміття, обробкою виключень і управлінням ресурсами. Код, який виконується CLR, іноді називають «керованим кодом» на відміну від «некерованого коду», який компілюється на рідній машинній мові, орієнтований на певну систему. На наступному рисунку показані відносини часу компіляції і часу виконання файлів вихідного коду C#, бібліотек класів .NET Framework, збірок і середовища CLR.

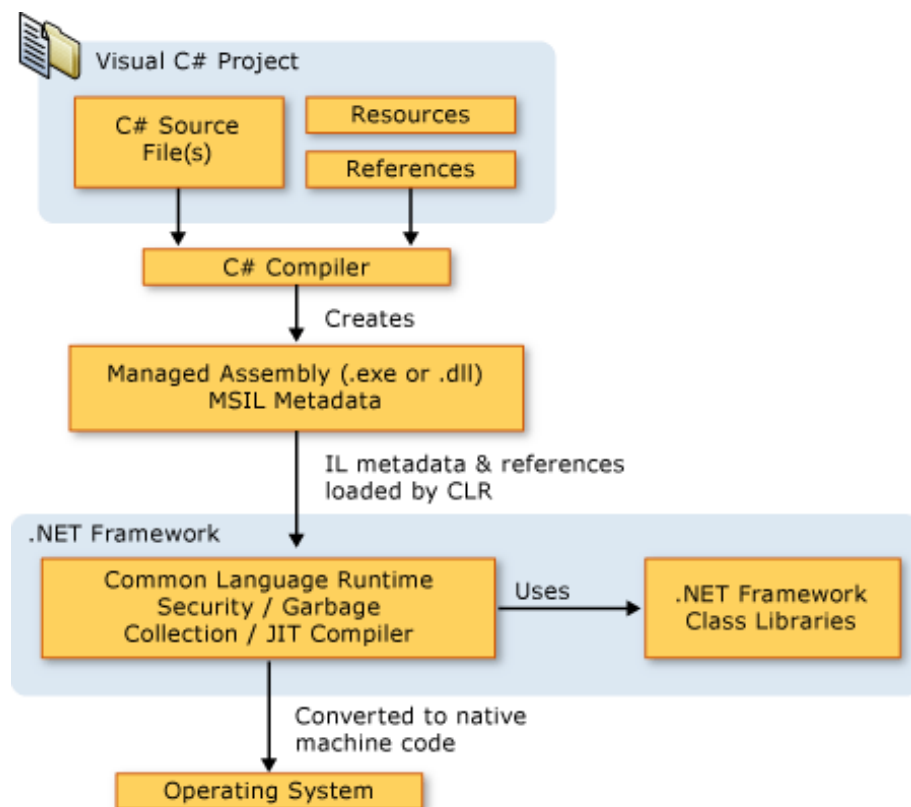


Рисунок 3.1 – Час компіляції та час виконання вихідного коду C#

Мовна інтероперабельність є ключовою особливістю .NET Framework. Оскільки IL-код, створений компілятором C#, відповідає специфікації Common Type Specification (CTS). IL-код, створений на C#, може взаємодіяти з кодом, який був згенерований з .NET-версій Visual Basic, Visual C++ або будь-якої з більш ніж 20 інших CTS-сумісних мов. Одна збірка може містити кілька модулів, написаних на різних мовах .NET, і типи можуть посилатися один на одного так само, як якщо б вони були написані на одній мові.

На додаток до служб часу виконання .NET Framework також включає велику бібліотеку з більш ніж 4000 класів, організованих в простір імен, які надають широкий спектр корисних функцій для всього: від введення і виведення файлів до маніпуляції з рядками, аналізу XML, до елементів управління Windows Forms. Типовий додаток C # широко використовує бібліотеку класів .NET Framework, щоб справлятися з повсякденними справами.

3.1.2 Огляд технологій при розробці проекту на JavaScript

JavaScript - мультипарадигмена мова програмування. Підтримує об'єктно-орієнтований, імперативний і функціональний стилі. Є реалізацією мови ECMAScript.

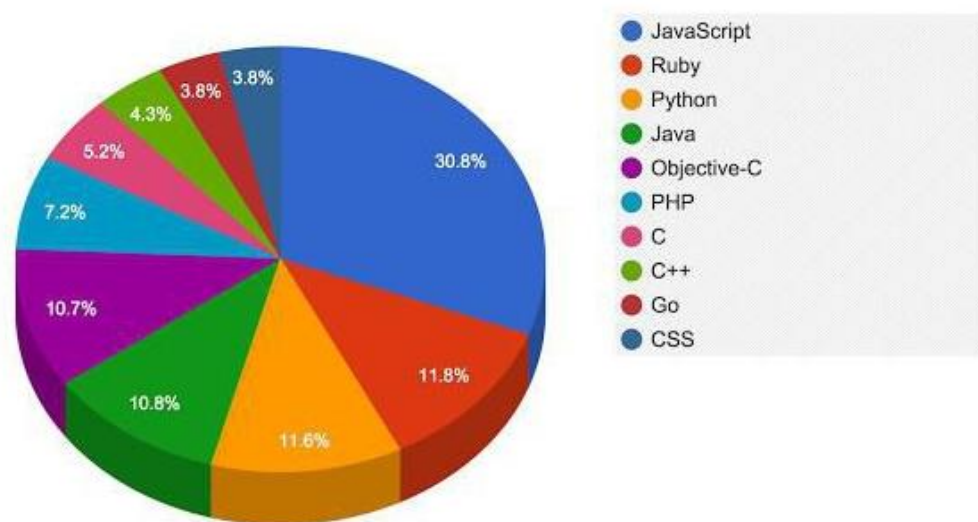


Рисунок 3.2 – Популярність мов програмування в проектах Github

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістів. Мовою JavaScript не володіє будь-яка компанія або організація, що відрізняє її від ряду мов програмування, використовуваних в веб-розробці.

JavaScript є об'єктно-орієнтованою мовою, але використання в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними клас-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, карінг, анонімні функції, замикання - що додає мові додаткову гнучкість.

Незважаючи на схожий з Сі синтаксис, JavaScript в порівнянні з мовою Сі має корінні відмінності:

- об'єкти з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

У мові відсутні такі корисні речі, як:

- стандартна бібліотека: зокрема, відсутній інтерфейс програмування додатків по роботі з файловою системою, управління потоками введення-виведення, базових типів для бінарних даних;
- стандартні інтерфейси до веб-серверів і баз даних;
- система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх.

Структурно JavaScript можна представити у вигляді об'єднання трьох чітко помітних частин:

- ядро (ECMAScript);
- об'єктна модель браузера (Browser Object Model або BOM);
- об'єктна модель документа (Document Object Model або DOM).

Якщо розглядати JavaScript в відмінних від браузера середовищах, то об'єктна модель браузера і об'єктна модель документа можуть не підтримуватися.

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа. На противагу цьому ряд авторів знаходять BOM і DOM тісно взаємопов'язаними.

ECMAScript не є браузерною мовою і в ній не визначаються методи введення і виведення інформації. Це, скоріше, основа для побудови скриптових мов. Специфікація ECMAScript описує типи даних, інструкції, ключові і зарезервовані слова, оператори, об'єкти, регулярні вирази, не обмежуючи авторів похідних мов в розширенні їх новими складовими.

Об'єктна модель браузера - браузерно-специфічна частина мови, що є прошарком між ядром і об'єктною моделлю документа. Основне призначення об'єктної моделі браузера - управління вікнами браузера і забезпечення їх взаємодії. Кожне з вікон браузера представляється об'єктом window, центральним об'єктом DOM. Об'єктна модель браузера на даний момент не стандартизована, однак специфікація знаходиться в розробці WHATWG і W3C.

Крім управління вікнами, в рамках об'єктної моделі браузера, браузерами зазвичай забезпечується підтримка наступних сутностей:

- управління кадрами;
- підтримка затримки у виконанні коду і зациклення з затримкою;
- системні діалоги;
- управління адресою відкритої сторінки;
- управління інформацією про браузери;
- управління інформацією про параметри монітора;
- обмежене управління історією перегляду сторінок;
- підтримка роботи з HTTP cookie.

Об'єктна модель документа - інтерфейс програмування додатків для HTML і XML-документів. Згідно DOM, документ (наприклад, веб-сторінка) може бути представлений у вигляді дерева об'єктів, що володіють рядом властивостей, які дозволяють виробляти з ним різні маніпуляції:

- генерація і додавання вузлів;
- отримання вузлів;
- зміна вузлів;
- зміна зв'язків між вузлами;
- видалення вузлів.

JavaScript використовується в клієнтській частині веб-додатків: клієнт-серверних програм, в якому клієнтом є браузер, а сервером - веб-сервер, що мають розподілену між сервером і клієнтом логіку. Обмін інформацією з веб-додатком відбувається по мережі. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є кросплатформеними сервісами.

JavaScript використовується в AJAX, популярному підході до побудови призначених для користувача інтерфейсів веб-додатків, що полягає в «фоновому» асинхронному обміні даними браузера з веб-сервером. В результаті, при оновленні даних веб-сторінка не перезавантажується повністю і інтерфейс веб-додатку стає швидше, ніж це відбувається при традиційному підході (без застосування AJAX).

Comet - широке поняття, яке описує механізм роботи веб-додатків, що використовують постійні HTTP-з'єднання, що дозволяє веб-серверу відправляти дані браузеру без додаткового запиту з боку браузера. Для таких додатків використовуються технології, безпосередньо підтримувані браузерами. Зокрема, в них широко використовується JavaScript.

JavaScript широко використовується в браузерних операційних системах. Так, наприклад, вихідний код IndraDesktop WebOS на 75% складається з JavaScript, код браузерної операційної системи IntOS - на 70%. Частка JavaScript у вихідному коді eyeOS - 5% , однак і в рамках цієї операційної системи JavaScript грає важливу роль, беручи участь в візуалізації на клієнті і є необхідним механізмом для комунікування клієнта і сервера.

Призначені для користувача скрипти в браузері - це програми, написані на JavaScript, що виконуються в браузері користувача при завантаженні сторінки. Вони дозволяють автоматично заповнювати форми, переформатовувати сторінки, приховувати небажаний вміст і вбудовувати бажаний для відображення вміст, змінювати поведінку клієнтської частини веб-додатків, додавати елементи керування на сторінку і т. д.

Для управління призначеними для користувача скриптами в Mozilla Firefox використовується розширення Greasemonkey; Opera і Google Chrome надають засоби підтримки користувальницьких скриптів і можливості для виконання ряду скриптів Greasemonkey.

JavaScript використовується для написання прикладного ПЗ. Наприклад, 16,4% вихідного коду Mozilla Firefox написано на JavaScript.

Google Chrome OS в якості прикладного ПЗ використовує веб-додатки. В оточенні робочого столу GNOME є можливість створювати на JavaScript програми, які оперують з бібліотеками GNOME за допомогою Gjs, Seed.

JavaScript також знаходить застосування в якості скриптові мови доступу до об'єктів додатків. Платформа Mozilla (XUL/Gecko) використовує JavaScript. Серед сторонніх продуктів, наприклад, Java, починаючи з версії 6, містить вбудований інтерпретатор JavaScript на базі Rhino. Сценарії JavaScript підтримуються в таких додатках Adobe, як Adobe Photoshop, Adobe Dreamweaver, Adobe Illustrator і Adobe InDesign.

Стандартизація JavaScript зажадала піти від проблем з торговельними марками, тому стандарт ECMA 262 називає мову ECMAScript, три редакції якої були опубліковані з початку роботи над ним в листопаді 1996 року.

Objective-J - це строга компактна надмножина JavaScript, в якій до JavaScript додано:

- традиційне спадкування;
- динамічна диспетчеризація методів в стилі Smalltalk/Objective-C;
- псевдостатична типізація.

VBScript від Microsoft, подібно JavaScript, може виконуватися на стороні клієнта в веб-сторінках. VBScript має синтаксис, похідний від Visual Basic і підтримується тільки в Internet Explorer.

JSON або об'єктна нотація JavaScript, це формат обміну даними загального призначення, певний як підмножина JavaScript.

Scheme також є родинною JavaScript мовою, оскільки обидві представляють великі можливості функціонального програмування: JavaScript є динамічною мовою, підтримує гнучкі масиви, може легко симулювати s-вирази, має підтримку лямбда-виразів.

Загальною помилкою є те, що JavaScript аналогічна або тісно пов'язана з Java, це не так. Обидві мови мають C-подібний синтаксис, є об'єктно-орієнтованими і як правило широко використовуються в клієнтських веб-додатках.

З важливих відмінностей можна відзначити:

- Java реалізує ООП підхід, заснований на класах, JavaScript – на прототипах;
- Java має статичну типізацію, JavaScript - динамічну типізацію;
- Java завантажується з скомпільованого байт-коду; JavaScript інтерпретується безпосередньо з файлу (але часто з непомітною JIT-компіляцією).

У деяких мовах програмування існують засоби підтримки взаємодії з JavaScript-кодом.

Для PHP є пакет HTML Javascript, що надає інтерфейс створення простих JavaScript-програм. Відповідний пакет для Tcl називається :: javascript. Він надає команди для створення коду HTML і JavaScript. Пакет для Perl Data :: JavaScript дозволяє переносити структури даних Perl в JavaScript-код

На сьогоднішній день підтримку JavaScript забезпечують сучасні версії всіх найбільш часто використовуваних браузерів. В Internet Explorer, Opera, Mozilla Firefox, Safari, Google Chrome є повна підтримка третьої редакції ECMA-262. При цьому в Mozilla Firefox зроблена спроба здійснення підтримки четвертої редакції специфікації, а першим браузером, в якому з'явилася неповна підтримка специфікації 3.1, з'явився Internet Explorer 8.

Допущені розробниками популярних браузерів помилки в реалізації специфікації, як правило, незначні. Станом на листопад 2009 року об'єктна модель документа має більш обмежену підтримку.

На думку творця мови, підтримка в Internet Explorer компанією Microsoft одного з існуючих і застосованих в інших браузерах швидких двигунів JavaScript здатна привести до появи додатків, що працюють з тривимірною графікою, написаних на JavaScript 3D-ігор, використання JavaScript в задачах, в яких раніше застосовувалася технологія Adobe Flash.

Регресійне тестування відповідності браузерів третьої редакції специфікації ECMA-262 може здійснюватися за допомогою розробленого Google інструменту для тестування відповідності специфікації ECMAScript Sputnik, що включає більше п'яти тисяч варіантів тестування і отримав назву по імені російської команди Google, а також написаної Юрієм Зайцевим оболонки sputniktests-webrunner або сервісу Google. Варіанти тестування, що входять до Sputnik оновлюються в зв'язку з виходом п'ятої редакції специфікації ECMA-262, відображаючи зміни в порівнянні з її попередньою редакцією.

ECMAScript 5 Conformance Suite являє собою набір тестів, випущений Microsoft під ліцензією BSD, для перевірки відповідності реалізації мови ECMAScript його п'ятої редакції специфікації. Станом на 12 березня 2010 року в пакеті налічувалося 1236 варіантів тестування, він мав версію 0.2 альфа і кількість завантажень за три місяці становило 178. Для перевірки коректності реалізацій JavaScript є набір тестів JavaScript Test Suite, випущених Mozilla.

JavaScript дозволяє потенційним авторам шкідливого коду запускати його на будь-якому комп'ютері мережі: для цього досить відкрити на ньому веб-сторінку.

JavaScript-програми виконуються в «пісочниці», в якій вони можуть виконувати тільки обмежене коло дій, а не завдання програмування загального призначення (наприклад, створення файлів, робота з сокетом). Для JavaScript-коду застосовується політика спільного походження, відповідно до якої скрипт, вбудований в сторінку, не може отримати доступ до ряду властивостей об'єктів іншої сторінки (зокрема, до більшості властивостей об'єкта document) при відміні в протоколі, хост і номер порту цих сторінок.

Крім цього, розробники браузерів вносять додаткові обмеження у відповідь на що має місце зловживання. Так з'явилася, зокрема, заборона на відкриття вікна, розмір одного боку якого менше ста пікселів.

Загальна проблема, що стосується JavaScript - міжсайтовий скриптинг або XSS, порушення політики загального походження. Уразливості XSS мають місце в ситуаціях, коли зловмисник має можливість помістити скрипт на сторінку, яка відображається. В цьому випадку скрипт отримує доступ до сайту з правами цього користувача, що в ряді випадків відкриває можливість відсилання конфіденційної інформації, здійснення

небажаних транзакцій. Уразливості XSS також відбуваються через помилки, допущені розробниками браузерів.

Іншим типом міжсайтової уразливості є підробка міжсайтових запитів або CSRF. Вона полягає в можливості сайту зловмисника змусити браузер користувача здійснити небажану дію на цільовому сайті (наприклад, банківський переказ грошей). Така можливість є, якщо цільовий сайт покладається тільки на HTTP cookie або запити авторизації. У цьому випадку запити, початі кодом сайту зловмисника, виконуються так само як запити користувача, якщо він авторизований на цільовому сайті. Одним із засобів захисту від CSRF є здійснення аутентифікації при будь-якому запиті, який призводить до незворотних наслідків. Також може допомогти аналіз HTTP referer.

Розробники клієнтських додатків, незалежно від того, використовують вони JavaScript чи ні, повинні усвідомлювати, що останні можуть перебувати під контролем зловмисників. Тому будь-яка перевірка на стороні клієнта може бути обійдена, JavaScript може бути як запущений, так і ні. Код, який зазнавав обфускації може стати об'єктом зворотньої розробки; дані форми можуть бути послані на сервер, минаючи валідацію, здійснювану за допомогою JavaScript; скрипти можуть бути відключені частково, тому, наприклад, надійний захист від збереження зображень за допомогою JavaScript здійснити не можна; надзвичайно необачно впроваджувати пароль в JavaScript, виконуваний на клієнті, де він може бути знайдений зловмисником.

JavaScript надає інтерфейс до широкого спектру можливостей браузера, деякі з яких можуть містити помилки, що призводять, наприклад, до переповнення буфера. Це дозволяє писати скрипти, що призводять до виконання довільного коду на користувальницькій системі.

Подібні помилки виявлялися у часто використовуваних браузерів, включаючи Mozilla Firefox, Internet Explorer, Safari. При виявленні потенційно небезпечних помилок в браузері і наявності відомостей про реалізовані експлойти фірми-виробники і експерти з безпеки рекомендують відключати JavaScript до виходу патча.

Модулі, такі як плеєри, Macromedia Flash і ряд ActiveX компонент, доступних за умовчанням в Internet Explorer, можуть також містити помилки, експлуатовані за допомогою JavaScript, що вже траплялося раніше.

Розширення Mozilla Firefox не ізольовані один від одного: одне розширення може виправляти інше, що може використовуватися зловмисниками. На конференції SecurityByte & Owasp AppSec Asia 2009 Роберто Сагг Ліверані і Нік Фрімен продемонстрували три експлойта в популярних розширеннях Firefox, завантажені з сайту більше 30 мільйонів разів.

3.2 Програмна реалізація

Розроблений програмний додаток виконує наступні основні функції:

- завантаження власного відеоролика;
- сумаризація відеоролика;
- збереження результату роботи програми;
- побудова діаграми динамічної трансформації часової шкали.

Інтерфейс розробленого програмного додатку можна побачити на рисунку 3.3. У формі додатку розташоване вікно для виводу графіку алгоритму DTW (дод. А та Б).

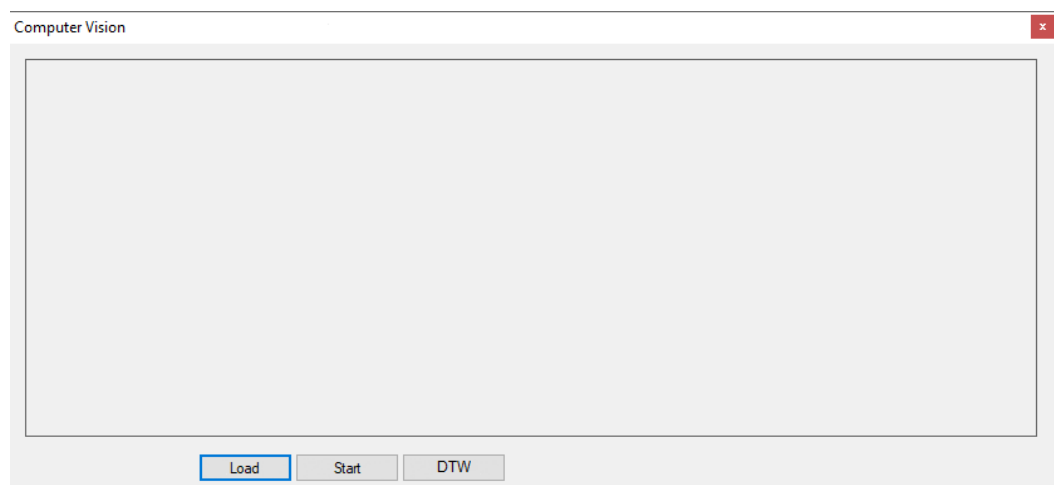


Рисунок 3.3 – Інтерфейс розробленої програми

До додаткових функцій можна віднести відкриття користувацького відеоролику.

При натисненні на кнопку «Load» завантажуються відеоролик. При натисненні кнопки «Start» починається сумаризація відео. При завершенні сумаризації відеоролику він зберігається у відповідну папку та стає можливою побудова графіку, як зображено на рисунку 3.4.

При роботі з програмним додатком користувачеві, у першу чергу, необхідно завантажити відеоролик, скориставшись кнопкою «Load».

У процесі роботи з програмою користувач може експериментувати з різними зображеннями у різних кольорних моделях та з різним числом кластерів.

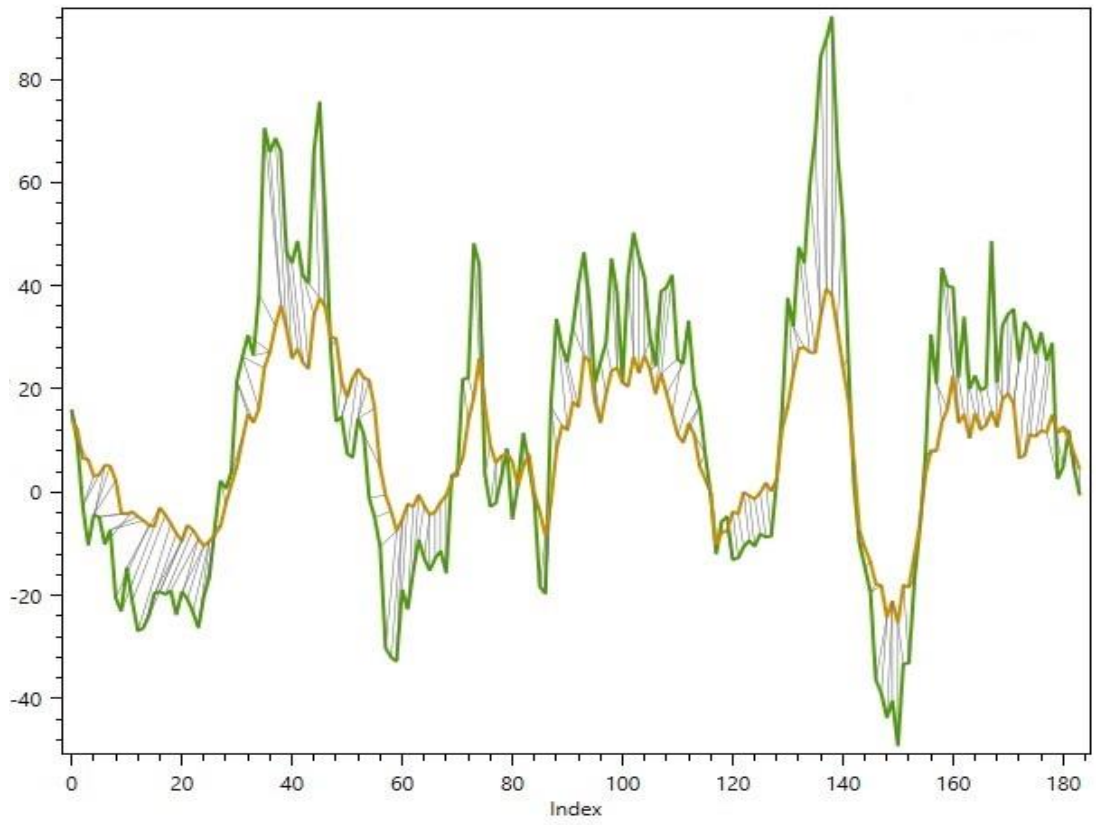


Рисунок 3.4 – Результат побудови діаграми динамічної трансформації часової шкали

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Завданням даної роботи було дослідження методів анотування відео потоків та створення спеціалізованого програмного засобу для анотації відео потоків. Так як в процесі виконання завдань магістерської роботи використовувалося комп'ютерне обладнання, то аналіз потенційно небезпечних і шкідливих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка.

В даному розділі проведено аналіз потенційних небезпечних та шкідливих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики. Розглянуті питання впливу на навколишнє природне середовище діяльності за темою магістерської роботи.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави і особистої відповідальності.

4.2 Аналіз стану умов праці

Робота проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

| Найменування | Значення |
|-----------------------|----------|
| Довжина, м | 5 |
| Ширина, м | 5 |
| Висота, м | 2.8 |
| Площа, м ² | 25 |
| Об'єм, м ³ | 70 |

Згідно з [2] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [3] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Приміщення знаходиться на другому поверсі трьох поверхової будівлі і має об'єм 70 м³, площу – 25 м². Обладнано одне місце праці укомплектоване ПК.

Таблиця 4.2 - Характеристики робочого місця

| Найменування параметра | Фактичне значення | Нормативне значення |
|--|-------------------|---------------------|
| Висота робочої поверхні, мм | 700 | 680 800 |
| Висота простору для ніг, мм | 750 | не менше 600 |
| Ширина простору для ніг, мм | 550 | не менше 500 |
| Глибина простору для ніг, мм | 700 | не менше 650 |
| Висота поверхні сидіння, мм | 500 | 400 500 |
| Ширина сидіння, мм | 450 | не менше 400 |
| Глибина сидіння, мм | 470 | не менше 400 |
| Висота поверхні спинки, мм | 400 | не менше 300 |
| Ширина опорної поверхні спинки, мм | 400 | не менше 380 |
| Радіус кривини спинки в горизонтальній площині, мм | 400 | 400 |
| Відстань від очей до екрану дисплея, мм | 750 | 700 800 |

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви програм тривалістю 15 хв. через кожен годину роботи.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [6] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи

із застосуванням ЕОМ з ВДТ і ПП.

Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220\text{В} \pm 5\%$;
- робочий струм $I=2\text{А}$;
- споживана потужність $P=350\text{ Вт}$.

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [3].

Таблиця 4.3 – Аналіз небезпечних і шкідливих факторів

| Небезпечні і шкідливі виробничі фактори | Джерела факторів (види робіт) | Кількісна оцінка | Нормативні документи |
|---|---|------------------|----------------------|
| фізичні | | | |
| - підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини | -//- | 4 | [4] |
| - недостатність природного світла | порушення умов праці (вимог до приміщень) | 2 | [5] |
| - недостатнє освітлення робочої зони | порушення гігієнічних параметрів виробничого середовища | 3 | [5] |
| психофізіологічні: | | | |
| - нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових) | - пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи | 4 | [6] [3] |
| - фізичні (статичне – сидіння) | порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота) | 2 | [6] [3] |

4.3.2 Пожежна безпека

Небезпека розвитку пожежі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання.

Згідно ДБН В.2.5-28:2018 [5] таке приміщення, площею 25 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому можливо встановлення автоматичної пожежної сигналізації із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві

рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення мікроклімату для робочого місця відповідають ДСН 3.3.6.042-99 [2] (табл. 4.4):

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

| Період року | Категорія робіт | Температура С ⁰ | Відносна вологість % | Швидкість руху повітря, м/с |
|-------------|-----------------|----------------------------|----------------------|-----------------------------|
| Холодна | легка-1 а | 22 - 24 | 40 – 60 | 0,1 |
| Тепла | легка-1 а | 23 - 25 | 40 – 60 | 0,1 |

У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату. Дане приміщення обладнане системою опалення, кондиціонування повітря. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

Рівні позитивних і негативних іонів у повітрі мають відповідати ДСН 3.3.6.042-99 [2].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [5]. Джерелом природного освітлення є

сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6 \text{ м}^2$ кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 25 \text{ м}^2$;

Z – поправочний коефіцієнт світильника ($Z=1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, зумовлений як роботою системного блоку, клавіатури, так і друкуванням на принтері, а також зовнішніми чинниками, коливається у межах 50–65 дБА ДСН 3.3.6.042-99 [2].

Віброізоляцію можливо здійснювати за допомогою спеціальної прокладки під системний блок, яка послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.042-99 [2].

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) *Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:*

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-втяжну вентиляцію або кондиціонування повітря.

2) *Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:*

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;
- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;
- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводиться робота відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho=40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{\text{розр}}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{\text{розр.в}}$, і горизонтальних $\rho_{\text{розр.г}}$, Ом·м за формулою:

$$\rho_{\text{розр.}} = \psi \cdot \rho, \quad (4.4)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{\text{розр.в}}=1,7$ і горизонтальних $\rho_{\text{розр.г}}=5,5$ Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача $R_{\text{в}}$, Ом, за формулою (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left(\ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.5)$$

де $l_{\text{в}}$ – довжина вертикального заземлювача (для труб – 2–3 м; $l_{\text{в}}=3$ м);

$d_{\text{ст}}$ – діаметр стержня (для труб – 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad (4.6)$$

де $h_{\text{в}}$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді $t = 0,8 + \frac{3}{2} = 2,3$

м

$$R_{\text{в}} = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання $\eta_{\text{в}}$:

$$n = \frac{2 \cdot R_{\text{в}}}{R_{\text{д}}} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_B = 0,57$ (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти за $L_B = 3$ м);

n_B – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_Γ , Ом:

$$R_\Gamma = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_\Gamma}, \quad (4.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

h_Γ – глибина закладання горизонтальних заземлювачів (0,5 м);

l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_\Gamma = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c , відповідно до необхідної кількості вертикальних заземлювачів n_B .

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B \cdot R_\Gamma}{R_B \cdot \eta_c + R_\Gamma \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.18)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} < 4$ Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

4.6 Екологія

Діяльність за темою магістерської роботи, а саме: дослідження методів анотування відео потоків та створення спеціалізованого програмного засобу для анотації відео потоків в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [7], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [8], Законом України «Про відходи» [9].

В процесі діяльності з виконання дипломного проектування виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

Висновки до розділу 4

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики. Розглянуті заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій. У розділі «Екологія» розглянуті питання впливу на навколишнє природне середовище діяльності за темою магістерської роботи.

Перелік корисних посилань до розділу 4

1. Закон України «Про охорону праці». Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> - 10.14.1992 р.
2. Державні санітарні норми. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/va042282-99> - 01.02.1999 р.
3. Державні санітарні правила і норми. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> - 10.12.1998 р.
4. Державний стандарт України. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» - Режим доступу: <http://epicentre.co.ua/dstu/doc28522.html> - 01.07.2016 р.
5. Державні будівельні норми. ДБН В.2.5-28:2018 «Природне і штучне освітлення» - Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf> - 03.10.2018
6. Нормативно-правовий акт з охорони праці. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> - 14.02.2018 р.
7. Закон України «Про охорону навколишнього природного середовища» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/1264-12> - 26.06.1991 р.
8. Закони України «Про охорону навколишнього природного середовища» - Режим доступу - <https://zakon.rada.gov.ua/laws/show/4004-12> - 24.02.1994 р.
9. Закон України «Про відходи» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80> – 05.03.1998 р.

ВИСНОВКИ

З метою забезпечення ефективного способу перегляду великої колекції відео, відеосумаризація привернула велику увагу в минулому десятилітті. Було запропоновано кілька підходів сумаризації відео, використовуючи знання про домен або навчання контрольованої моделі з маркуванням баз даних. Однак досі залишається актуальним завдання формулювання такої моделі, яка буде здатна впоратись з усім різноманіттям відеоконтенту без людського спостереження.

У результаті виконання роботи був проведений аналіз існуючих методів анотування відеопотоків та розроблений власний програмний додаток для анотування відео, який буде мати практичне застосування.

Для цього були вирішені наступні питання:

- проведений порівняльний аналіз існуючих методів анотування відео;
- розроблений алгоритм анотування відео на основі кластеризації;
- реалізований алгоритм анотування відео у вигляді програмного додатку.

Проаналізувавши програмну частину, проект був реалізований за допомогою мов програмування C# та JavaScript, оскільки вони містять у собі необхідні засоби та бібліотеки для успішного досягнення поставленої мети.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Diverse sequential subset selection for supervised video summarization [Електронний ресурс] – режим доступу до ресурсу: <https://papers.nips.cc/paper/5413-diverse-sequential-subset-selection-for-supervised-video-summarization.pdf>
- 2) A generic framework of user attention model and its application in video summarization [Електронний ресурс] – режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-3-540-92892-8_48
- 3) Summarizing web videos using titles [Електронний ресурс] – режим доступу до ресурсу: https://www.cvfoundation.org/openaccess/content_cvpr_2015/papers/Song_TVSum_Summarizing_Web_2015_CVPR_paper.pdf
- 4) Multi-task deep visualesemantic embedding for video thumbnail selection [Електронний ресурс] – режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s11704-016-5438-1>
- 5) Formulating team sport video summarization as a resource allocation problem [Електронний ресурс] – режим доступу до ресурсу: https://www.researchgate.net/publication/235747981_Formulating_Team-Sport_Video_Summarization_as_a_Resource_Allocation_Problem
- 6) Towards scalable summarization of consumer videos via sparse dictionary selection [Електронний ресурс] – режим доступу до ресурсу: <https://papers.nips.cc/paper/7724-fast-greedy-algorithms-for-dictionary-selection-with-generalized-sparsity-constraints.pdf>
- 7) Discovering important people and objects for egocentric video summarization [Електронний ресурс] – режим доступу до ресурсу: http://vision.cs.utexas.edu/projects/egocentric/egocentric_cvpr2012.pdf
- 8) Category-specific video summarization [Електронний ресурс] – режим доступу до ресурсу: https://hal.inria.fr/hal01022967/PDF/video_summarization.pdf
- 9) Video summarization using shot segmentation and local motion estimation [Електронний ресурс] – режим доступу до ресурсу: <http://www.rroij.com/open-access/a-study-on-video-summarization-techniques.pdf>
- 10) Video summarization by clustering using euclidean distance [Електронний ресурс] – режим доступу до ресурсу: https://www.researchgate.net/publication/266032463_Video_Summarization_Using_Clustering

11) A Study on Keyframe Extraction Methods for Video Summary [Электронный ресурс] – режим доступа до ресурсу: http://www.ijircce.com/upload/2014/february/7I_A%20Study.pdf

12) An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram [Электронный ресурс] – режим доступа до ресурсу: https://www.researchgate.net/profile/Afdel_Karim/publication/270882684_Video_Shot_Detection_and_Key_Frame_Extraction_Using_Faber_Shoulder_DWT_and_SVD/links/54b6c0b30cf2bd04be328349/Video-Shot-Detection-and-Key-Frame-Extraction-Using-Faber-Shoulder-DWT-and-SVD.pdf

13) A video summarization approach based on the emulation of bottom-up mechanisms of visual attention [Электронный ресурс] – режим доступа до ресурсу: <https://www.springerprofessional.de/a-video-summarization-approach-based-on-the-emulation-of-bottom-/11990910>

14) Fast Summarization of User Generated Videos: Exploiting Semantic [Электронный ресурс] – режим доступа до ресурсу: <https://ieeexplore.ieee.org/document/7436667>

15) Key-frame Extraction by Analysis of Histograms of Video Frames Using Statistical Methods [Электронный ресурс] – режим доступа до ресурсу: https://www.researchgate.net/publication/284358522_Key-frame_Extraction_by_Analysis_of_Histograms_of_Video_Frames_Using_Statistical_Methods

16) Simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing [Электронный ресурс] – режим доступа до ресурсу: <https://ieeexplore.ieee.org/document/4712476>

17) Distinctive image features from scale-invariant key-points [Электронный ресурс] – режим доступа до ресурсу: <https://books.google.com.ua/books?id=97QebyNxyaYC&pg=PA180&lpg=PA180&dq=Lowe+D.G.+%E2%80%9CDistinctive+image+features+from+scale-invariant+key-points.+Int+J+Comput+Vision>

18) Clustering by fast search and find of density peaks. Science [Электронный ресурс] – режим доступа до ресурсу: <https://arxiv.org/ftp/arxiv/papers/1501/1501.04267.pdf>

19) Efficient visual search of videos cast as text retrieval [Электронный ресурс] – режим доступа до ресурсу: <https://www.di.ens.fr/~josef/publications/sivic09a.pdf>

20) Dynamic programming algorithm optimization for spoken word recognition [Электронный ресурс] – режим доступа до ресурсу: https://www.irit.fr/~Julien.Pinquier/Docs/TP_MABS/res/dtw-sakoe-chiba78.pdf

21) Curve matching, time warping, and light elds: New algorithms for computing similarity between curves [Электронный ресурс] – режим доступа до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.6645&rep=rep1&type=pdf>

22) Hand shape classification using DTW and LCSS as similarity measures for vision-based gesture recognition system [Электронный ресурс] – режим доступа до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.719.9144&rep=rep1&type=pdf>

23) A simple approximation for dynamic time warping search in large time series database [Электронный ресурс] – режим доступа до ресурсу: https://link.springer.com/chapter/10.1007/11875581_101

24) Keyframe-based video summarization using Delaunay clustering [Электронный ресурс] – режим доступа до ресурсу: <https://link.springer.com/article/10.1007/s00799-005-0129-9>

25) Video Shot Segmentation Using Spatio-Temporal Fuzzy Hostility Index and Automatic Threshold [Электронный ресурс] – режим доступа до ресурсу: https://www.researchgate.net/publication/259322984_Video_Shot_Segmentation_Using_Spatio-Temporal_Fuzzy_Hostility_Index_and_Automatic_Threshold

26) Video summarization using singular value decomposition [Электронный ресурс] – режим доступа до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.2342&rep=rep1&type=pdf>

27) Modeling the shape of the scene: A holistic representation of the spatial envelope [Электронный ресурс] – режим доступа до ресурсу: <http://cvcl.mit.edu/Papers/IJCV01-Oliva-Torralba.pdf>

28) Sequence alignment for redundancy removal in video rushes summarization [Электронный ресурс] – режим доступа до ресурсу: <http://www.eurecom.fr/en/publication/2575/download/mm-dumoem-081027.pdf>

29) Rushes summarization by IRIM consortium: redundancy removal and multi-feature fusion [Электронный ресурс] – режим доступа до ресурсу: <https://hal.archives-ouvertes.fr/hal-00347902/document>

30) A video summarization tool using two-level redundancy detection for personal video recorders [Электронный ресурс] – режим доступа до ресурсу: <https://ieeexplore.ieee.org/document/4560124>

31) Efficient video indexing scheme for content-based retrieval [Электронный ресурс] – режим доступа до ресурсу: https://link.springer.com/chapter/10.1007/978-3-540-87536-9_87

32) Automatic video summarization by graph modeling [Электронный ресурс] – режим доступа до ресурсу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.1272&rep=rep1&type=pdf>

33) Auto-summarization of audio-video presentations [Электронный ресурс] – режим доступа до ресурсу: http://www.cs.sfu.ca/CourseCentral/820/ii/material/source/papers/autosummarization_acm_99.pdf

ДОДАТОК А

Лістинг “DtwMatrixPlot.xaml.cs”

```
1      using System;
2      using System.Linq;
3      using System.Windows;
4      using System.Windows.Controls;
5      using OxyPlot;
6      namespace NDtw.Visualization.Wpf
7      {
8          public partial class DtwMatrixPlot : UserControl
9          {
10             public DtwMatrixPlot()
11             {
12                 InitializeComponent();
13             }
14             public static DependencyProperty DtwProperty =
15                 DependencyProperty.Register(
16                     "Dtw",
17                     typeof (IDtw),
18                     typeof (DtwMatrixPlot),
19                     new FrameworkPropertyMetadata(null, (d, e) =>
20 ((DtwMatrixPlot)d).OnDataChanged()));
21             public IDtw Dtw
22             {
23                 get { return (IDtw) GetValue(DtwProperty); }
24                 set { SetValue(DtwProperty, value); }
25             }
26             public static DependencyProperty DrawDistanceProperty =
27                 DependencyProperty.Register(
28                     "DrawDistance",
29                     typeof (bool),
30                     typeof (DtwMatrixPlot),
31                     new FrameworkPropertyMetadata(false, (d, e) =>
32 ((DtwMatrixPlot)d).OnDataChanged()));
33             public bool DrawDistance
34             {
35                 get { return (bool)GetValue(DrawDistanceProperty); }
36                 set { SetValue(DrawDistanceProperty, value); }
```



```

76         for (int j = 0; j < yLength; j++)
77             maxMatrixValue = Math.Max(maxMatrixValue,
78 Double.IsPositiveInfinity(matrixValues[i][j]) ? 0 :
79 matrixValues[i][j]);
80         for (int i = 0; i < xLength; i++)
81             for (int j = 0; j < yLength; j++)
82                 {
83                     var value = matrixValues[i][j];
84                     var isValuePositiveInfinity =
85 Double.IsPositiveInfinity(value);
86                     var intensityBytes =
87 isValuePositiveInfinity ? new byte[] { 0, 0, 0 } :
88 GetFauxColourRgbIntensity(value, 0, maxMatrixValue);
89                     //var intensityByte = (byte)(255 -
90 Math.Floor(255 * intensity));
91                     plotModel.Annotations.Add(new
92 PolygonAnnotation
93                     {
94                         Points =
95                             new[]
96                             {
97                                 new DataPoint(i - 0.5, j -
98 0.5), new DataPoint(i + 0.5, j - 0.5),
99                                 new DataPoint(i + 0.5, j +
100 0.5), new DataPoint(i - 0.5, j + 0.5),
101                             },
102                         StrokeThickness = 0,
103                         Selectable = false,
104                         Layer = AnnotationLayer.BelowAxes,
105                         Fill = OxyColor.FromArgb(255,
106 intensityBytes[0], intensityBytes[1], intensityBytes[2]),
107                     });
108                 }
109             for (int i = 0; i < 30; i++)
110                 {
111                     var intensityBytes =
112 GetFauxColourRgbIntensity(i, 0, 29);
113

```

```

114         plotModel.Annotations.Add(new
115     RectangleAnnotation
116         {
117             MinimumX = -39,
118             MaximumX = -25,
119             MinimumY = -i - 6,
120             MaximumY = -i - 5,
121             Selectable = false,
122             Fill = OxyColor.FromArgb(255,
123     intensityBytes[0], intensityBytes[1], intensityBytes[2])
124         });
125     }
126     plotModel.Annotations.Add(new TextAnnotation
127     {
128         Position = new DataPoint(-24, -5),
129         HorizontalAlignment =
130     HorizontalTextAlign.Left,
131         VerticalAlignment = VerticalTextAlign.Middle,
132         StrokeThickness = 0,
133         Text = "0"
134     });
135     plotModel.Annotations.Add(new TextAnnotation
136     {
137         Position = new DataPoint(-24, -34),
138         HorizontalAlignment =
139     HorizontalTextAlign.Left,
140         VerticalAlignment = VerticalTextAlign.Middle,
141         StrokeThickness = 0,
142         Text = String.Format("{0:0.00}",
143     maxMatrixValue),
144     });
145     }
146     var matrixPathSeries = new LineSeries("Path")
147     {
148         StrokeThickness = 1,
149         Color = OxyColors.Red,
150     };
151     for (int i = 0; i < dtwPath.Length; i++)

```

```

152         matrixPathSeries.Points.Add(new
153 DataPoint(dtwPath[i].Item1, dtwPath[i].Item2));
154         plotModel.Series.Add(matrixPathSeries);
155         var seriesMatrixScale = (xLength + yLength) * 0.05;
156         for (int variableIndex = 0; variableIndex <
157 Dtw.SeriesVariables.Length; variableIndex++)
158         {
159             var variableA =
160 Dtw.SeriesVariables[variableIndex];
161             var variableASeries = variableA.OriginalXSeries;
162             var variableB =
163 Dtw.SeriesVariables[variableIndex];
164             var variableBSeries = variableB.OriginalYSeries;
165             var minSeriesA = variableASeries.Min();
166             var maxSeriesA = variableASeries.Max();
167             var normalizedSeriesA = variableASeries.Select(x
168 => (x - minSeriesA) / (maxSeriesA - minSeriesA)).ToList();
169             var matrixSeriesA = new
170 LineSeries(variableA.VariableName);
171             for (int i = 0; i < normalizedSeriesA.Count; i++)
172                 matrixSeriesA.Points.Add(new DataPoint(i, (-1
173 + normalizedSeriesA[i]) * seriesMatrixScale - 1 - seriesMatrixScale *
174 (variableIndex + 1)));
175             plotModel.Series.Add(matrixSeriesA);
176             var minSeriesB = variableBSeries.Min();
177             var maxSeriesB = variableBSeries.Max();
178             var normalizedSeriesB = variableBSeries.Select(x
179 => (x - minSeriesB) / (maxSeriesB - minSeriesB)).ToList();
180             var matrixSeriesB = new
181 LineSeries(variableB.VariableName);
182             for (int i = 0; i < normalizedSeriesB.Count; i++)
183                 matrixSeriesB.Points.Add(new DataPoint(
184 normalizedSeriesB[i] * seriesMatrixScale - 1 - seriesMatrixScale *
185 (variableIndex + 1), i));
186             plotModel.Series.Add(matrixSeriesB);
187         }
188         plotModel.Axes.Add(new
189 LinearAxis(AxisPosition.Bottom, "          Series A") { Maximum =
190 Math.Max(xLength, yLength), PositionAtZeroCrossing = true});

```

```

191         plotModel.Axes.Add(new LinearAxis(AxisPosition.Left,
192 "           Series B") { Maximum = Math.Max(xLength, yLength),
193 PositionAtZeroCrossing = true });
194
195         MatrixPlot.Model = plotModel;
196     }
197     /// <summary>
198     /// Generate heatmap color
199     /// </summary>
200     /// <remarks>Thanks to Eddie Yee Tak Ma:
201 http://eddiema.ca/2011/01/21/c-sharp-heatmaps/ </remarks>
202     public static byte[] GetFauxColourRgbIntensity(double
203 val, double min, double max)
204     {
205         byte r = 0;
206         byte g = 0;
207         byte b = 0;
208         val = (val - min) / (max - min);
209         if (val <= 0.2)
210         {
211             b = (byte)((val / 0.2) * 255);
212         }
213         else if (val > 0.2 && val <= 0.7)
214         {
215             b = (byte)((1.0 - ((val - 0.2) / 0.5)) * 255);
216         }
217         if (val >= 0.2 && val <= 0.6)
218         {
219             g = (byte)(((val - 0.2) / 0.4) * 255);
220         }
221         else if (val > 0.6 && val <= 0.9)
222         {
223             g = (byte)((1.0 - ((val - 0.6) / 0.3)) * 255);
224         }
225         if (val >= 0.5)
226         {
227             r = (byte)(((val - 0.5) / 0.5) * 255);
228         }
229         return new byte[] { r, g, b };

```

```

230         }
231         /// <summary>
232         /// Generate heatmap color (grayscale)
233         /// </summary>
234         /// <remarks>Thanks to Eddie Yee Tak Ma:
235 http://eddiema.ca/2011/01/21/c-sharp-heatmaps/ </remarks>
236         public static byte[] GetGrayscaleRgbIntensity(double val,
237 double min, double max)
238         {
239             byte y;
240             val = (val - min) / (max - min);
241             y = (byte)((1.0 - val) * 255);
242             return new [] { y, y, y };
243         }
244         private const double AspectRatio = 1;
245         protected override void
246 OnRenderSizeChanged(SizeChangedEventArgs sizeInfo)
247         {
248             //keep aspect ratio
249             if (sizeInfo.NewSize.Width / sizeInfo.NewSize.Height
250 > AspectRatio)
251             {
252                 MatrixPlot.Width = sizeInfo.NewSize.Height *
253 AspectRatio;
254                 MatrixPlot.Height = MatrixPlot.Width /
255 AspectRatio;
256             }
257             else
258             {
259                 MatrixPlot.Height = sizeInfo.NewSize.Width /
260 AspectRatio;
261                 MatrixPlot.Width = MatrixPlot.Height *
262 AspectRatio;
263             }
264         }
265     }
266 }

```

ДОДАТОК Б

Лістинг “DtwByVariablePlot.xaml.cs”

```

1      using System;
2      using System.Windows;
3      using System.Windows.Controls;
4      using OxyPlot;
5      using System.Linq;
6      namespace NDtw.Visualization.Wpf
7      {
8          public partial class DtwByVariablePlot : UserControl
9          {
10             public DtwByVariablePlot()
11             {
12                 InitializeComponent();
13             }
14             public static DependencyProperty DtwProperty =
15                 DependencyProperty.Register(
16                 "Dtw",
17                 typeof (IDtw),
18                 typeof(DtwByVariablePlot),
19                 new FrameworkPropertyMetadata(null, (d, e) =>
20 ((DtwByVariablePlot)d).OnDtwChanged()));
21             public IDtw Dtw
22             {
23                 get { return (IDtw) GetValue(DtwProperty); }
24                 set { SetValue(DtwProperty, value); }
25             }
26             public void OnDtwChanged()
27             {
28                 var dtwPath = Dtw.GetPath();
29                 var xLength = Dtw.XLength;
30                 var yLength = Dtw.YLength;
31                 var cost = Dtw.GetCost();
32                 var costNormalized = Dtw.GetCost() /
33 Math.Sqrt(xLength * xLength + yLength * yLength);
34                 var plotModel = new PlotModel(String.Format("Dtw norm
35 by length: {0:0.00}, total: {1:0.00}", costNormalized, cost));

```

```

36
37         for (int variableIndex = 0; variableIndex <
38 Dtw.SeriesVariables.Length; variableIndex++)
39         {
40             var variableA =
41 Dtw.SeriesVariables[variableIndex];
42             var variableASeries =
43 variableA.GetPreprocessedXSeries();
44             var variableB =
45 Dtw.SeriesVariables[variableIndex];
46             var variableBSeries =
47 variableB.GetPreprocessedYSeries();
48
49             var axisTitleAndKey = String.Format("Value
50 {{0}}", variableA.VariableName);
51             plotModel.Axes.Add(new
52 LinearAxis(AxisPosition.Left, axisTitleAndKey) { Key =
53 axisTitleAndKey, PositionTier = variableIndex });
54             var plotSeriesA = new
55 LineSeries(variableA.VariableName) { YAxisKey = axisTitleAndKey };
56             for (int i = 0; i < xLength; i++)
57                 plotSeriesA.Points.Add(new DataPoint(i,
58 variableASeries[i]));
59             var plotSeriesB = new
60 LineSeries(variableB.VariableName) { YAxisKey = axisTitleAndKey };
61             for (int i = 0; i < yLength; i++)
62                 plotSeriesB.Points.Add(new DataPoint(i,
63 variableBSeries[i]));
64             var plotSeriesPath = new LineSeries("Dtw")
65             {
66                 YAxisKey =
67 axisTitleAndKey,
68                 StrokeThickness =
69 0.5,
70                 Color =
71 OxyColors.DimGray,
72             };
73             for (int i = 0; i < dtwPath.Length; i++)
74             {

```

```
75             plotSeriesPath.Points.Add(new
76 DataPoint(dtwPath[i].Item1, variableASeries[dtwPath[i].Item1]));
77             plotSeriesPath.Points.Add(new
78 DataPoint(dtwPath[i].Item2, variableBSeries[dtwPath[i].Item2]));
79             plotSeriesPath.Points.Add(new
80 DataPoint(double.NaN, double.NaN));
81         }
82         plotModel.Series.Add(plotSeriesA);
83         plotModel.Series.Add(plotSeriesB);
84         plotModel.Series.Add(plotSeriesPath);
85     }
86     plotModel.Axes.Add(new
87 LinearAxis(AxisPosition.Bottom, "Index"));
88     Plot.Model = plotModel;
89 }
90 }
91 }
```


Додаток В

Комп'ютерна презентація

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНИКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

МАГІСТЕРСЬКА РОБОТА

Дослідження методів анування відеопотоку та розробка програмного додатку для
анування відео

Виконав:
студент групи КІ-18дм

Гамалій А.М.

Керівник проекту:
д.т.н., професор

Рязанцев О.І.

Рисунок В.1- Слайд №1

СУМАРИЗАЦІЯ ВІДЕО




Рисунок В.2- Слайд №2

ПОСТАНОВКА ЗАДАЧІ

Метою цієї роботи є проведення аналізу методів анотування відеопотоку та розробка програмного додатку для анотування відео.

Для цього необхідно вирішити наступні питання:

- ◆ провести порівняльний аналіз існуючих методів анотування відео;
- ◆ розробити алгоритм анотування відео на основі одного з методів;
- ◆ реалізувати алгоритм анотування відео у вигляді програмного додатку .

Рисунок В.3- Слайд №3

СУМАРИЗАЦІЯ ВІДЕО

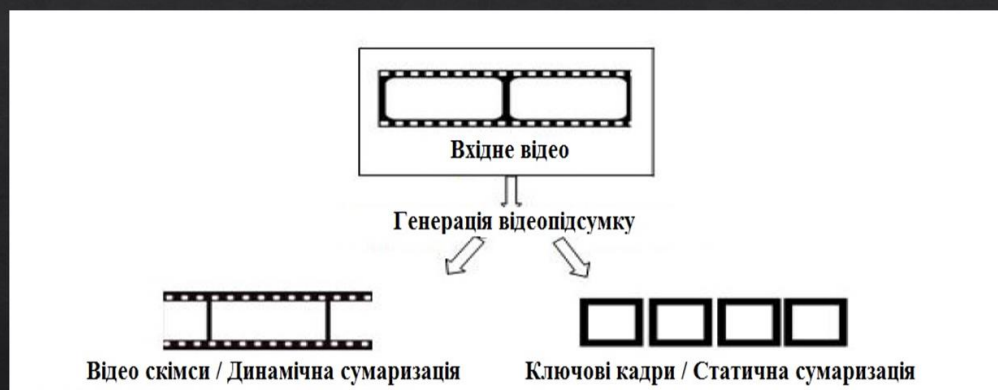


Рисунок В.4- Слайд №4

ПІДХОДИ ДО СУМАРИЗАЦІЇ ВІДЕО

- ◆ Сумаризація на основі низько- і високорівневих особливостей;
(Low Level Feature based Video Summarization and High Level Feature based Video Summarization)
- ◆ Одно- і багатокомпонентна сумаризація;
(Single Feature based Video Summarization and Multiple Feature based Video Summarization)
- ◆ Кластерна та некластерна сумаризація.
(Clustering based Video Summarization i Non-Clustering based Video Summarization)

Рисунок В.5- Слайд №5

СУМАРИЗАЦІЯ ШЛЯХОМ КЛАСТЕРИЗАЦІЇ

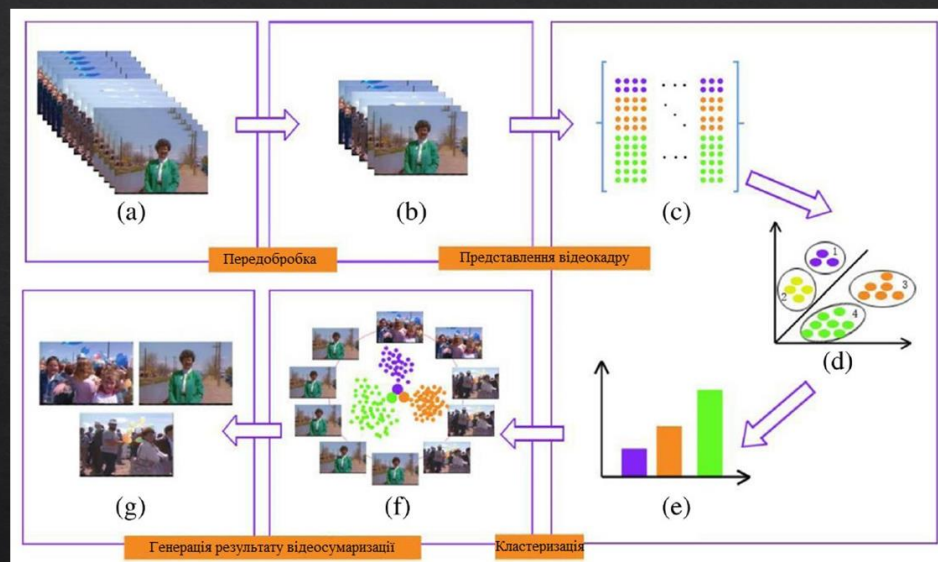


Рисунок В.6- Слайд №6

БУДОВА ВІДЕО

- Шот представляє собою послідовність схожих кадрів (будівельний блок відеопотоку)

- Ключовий кадр являє собою особливий візуальний зміст шоту (будівельний блок сумаризації)

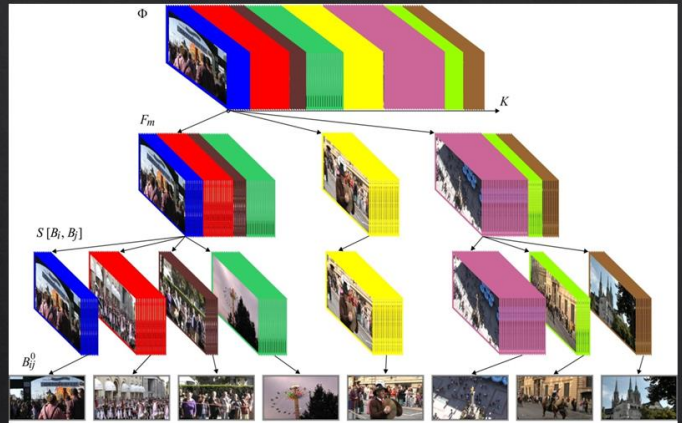
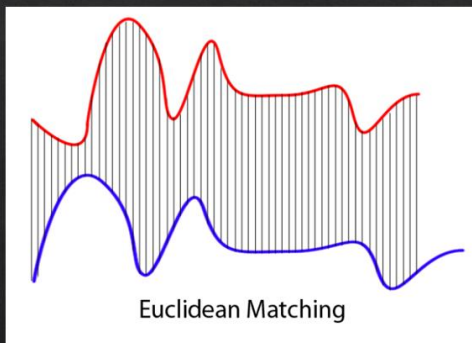
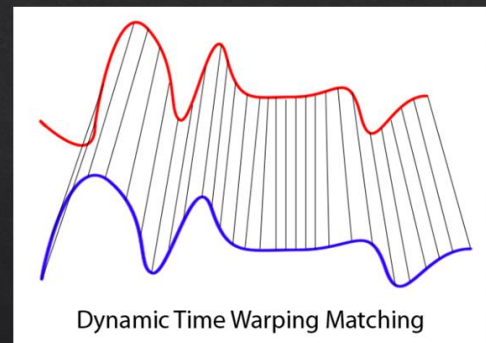


Рисунок В.7- Слайд №7

ДИНАМІЧНА ТРАНСФОРМАЦІЯ ЧАСОВОЇ ШКАЛИ



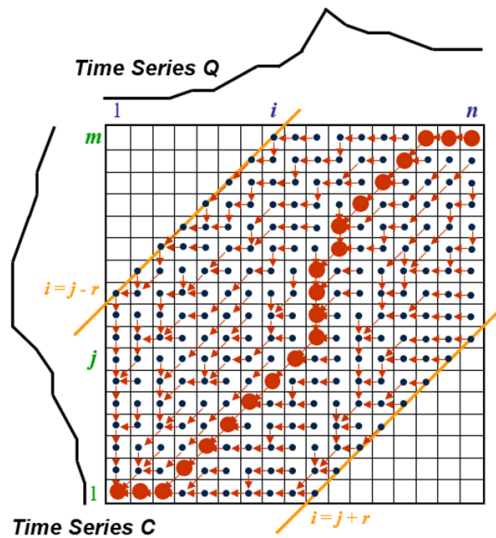
Евклідова відстань:
Послідовності вирівнюються один до одного



Викривлена вісь часу:
Можливі нелінійні вирівнювання послідовностей

Рисунок В.8- Слайд №8

ДИНАМІЧНА ТРАНСФОРМАЦІЯ ЧАСОВОЇ ШКАЛИ



Починаємо розрахунок з $g(1,1) = d(1,1)$
 Розраховуємо перший рядок $g(i, 1) = g(i-1, 1) + d(i, 1)$
 Розраховуємо першу колонку $g(1, j) = g(1, j-1) + d(1, j)$

Переходимо до другого рядку $g(i, 2) = \min(g(i, 1), g(i-1, 1), g(i-1, 2)) + d(i, 2)$. Книга зберігає для кожної комірки індекс сусідньої комірки, що сприяє мінімальному рахунку (червоні стрілки).

Продовжувати зліва направо і знизу вгору з рештою сітки $g(i, j) = \min(g(i, j-1), g(i-1, j-1), g(i-1, j)) + d(i, j)$.

Відстежуйте найкращий шлях через сітку, починаючи з $g(n, m)$ і рухаючись у напрямку $g(1,1)$ дотримуючись червоних стрілок.

Рисунок В.9- Слайд №9

КЛАСТЕРИЗАЦІЯ НА ОСНОВІ ПРЕДСТАВНИКІВ

$X_1, X_2, \dots, X_q, \dots, X_Q$ являє собою набір багатовимірних часових рядів, $\text{card } X_q = N_q$

$X_q, q = 1, 2, \dots, Q \Rightarrow WX_{q1}, WX_{q2}, \dots, WX_{qp}, \dots, WX_{qp}, p = 1, 2, \dots, P, \text{card } WX_{qp} = N_{qp}$

Де Q^P тимчасові сегменти

Ініціалізація $WX_{qp}^1(0), WX_{qp}^2(0), \dots, WX_{qp}^m(0)$ (m - задане число кластерів)

$\text{DTW}(WX_{qp}^1(0), WX_{qp}(0)) > \text{DTW}(WX_{rs}, WX_{qp}(0))$

$WX_{qp}^1(0)$ виключено з розгляду і обрано другого представника

$\text{DTW}(WX_{qp}^1(0), WX_{qp}(0)) > \text{DTW}(WX_{qp}^2(0), WX_{qp}(0)) > \dots > \text{DTW}(WX_{rs}, WX_{qp}(0))$

Вибір повторюється m разів

$\text{DTW}(WX_{qp}^1(0), WX_{qp}(0)) > \dots > \text{DTW}(WX_{qp}^m(0), WX_{qp}(0)) > \text{DTW}(WX_{rs}, WX_{qp}(0))$

Рисунок В.10- Слайд №10

КЛАСТЕРИЗАЦІЯ НА ОСНОВІ ПРЕДСТАВНИКІВ

Крок I)

Інші $QP - m$ шаблонів розподіляються над кластерами $Cl_j, j = \overline{1, m}$:

$$\forall j \neq l \in \{1, 2, \dots, m\}, j = \overline{1, m} \quad WX_{qp} \in Cl_j \text{ if } DTW(WX_{qp}, WX_{qp}^j(0)) < DTW(WX_{qp}, WX_{qp}^l(0))$$

Крок II)

У кожній з утворених груп визначається новий представник з мінімальною загальною відстанню до всіх точок початкового кластера

$$\sum DTW(WX_{qp} \in Cl_j, WX_{qp}^j(1)) < \sum DTW(WX_{qp} \in Cl_j, WX_{rs} \in Cl_j)$$

Після виявлення m уточнених $WX_{qp}^j(1), j = \overline{1, m}$ представників, повернення до етапу II) виконується

$$WX_{qp} \in Cl_j \text{ if } DTW(WX_{qp}, WX_{qp}^j(1)) < DTW(WX_{qp}, WX_{qp}^l(1)) \quad \forall j \neq l \in \{1, 2, \dots, m\}$$

Етап II) повторюється доти, доки не буде виконана умова стабілізації $WX_{qp}^j(\alpha+1) = WX_{qp}^j(\alpha)$ ($\alpha = 0, 1, 2, \dots$ це номер ітерації)

Рисунок В.11- Слайд №11



Рисунок В.12- Слайд №12

РЕЗУЛЬТАТ СУМАРИЗАЦІЇ



Рисунок В.13- Слайд №13