**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ХАРЧОВИХ ТЕХНОЛОГІЙ**

*Кафедра інформаційних технологій та кібербезпеки*

**ЗАПИСКА ПОЯСНЕННЯ**

**До Розрахунково-Графічного завдання**

по дисципліні

«**Мультимедійне програмування**»

«**Редактор відео»**

Виконав Баштовий О.В

Група 353м

Дата здачі звіту “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2013

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**Попков Д.М.**)

**Одеса - 2013**

**1.ВЕДЕННЯ**

Знімати "домашнє кіно" хочуть й люблять усі, та заодно слова "видеоредактор і відеомонтаж" багатьох викликають зневіру та викликають навіть певний трепет, як перед чимось складним; і важким. Проте пригадаймо зовсім недалеке минуле, коли аматорське кіно знімалося на плівку. Видеомонтаж у роки перебував лише з розрізування, вирізання і склеювання шматків плівки. І слід зазначити, виходили дуже непогані короткометражки. Єдине, що доводилося мріяти, - це про технічної корекції фільму, оскільки яскравість, контраст, колірної баланс і кадрування можна зробити лише під час зйомки, задаючи витримку, діафрагму, освітленість та інших.

Під час цифрового відео багато що змінилося, і комп'ютерних ефектів не обходиться жодного фільму. Проте задля домашнього аматорського кіно предосить мінімального відеомонтажу з допомогою простих безплатних видеоредакторов. Складні відеоефекти для сімейної хроніки непотрібні. Бажання продемонструвати своє майстерність "сама собі режисера" може лише погіршити враження від фільму, бо витримує перевірки професійними роботами, які бачимо екранами кінозалів. З іншого боку, спецефекти відеомонтажу знижують побутової колорит зйомки, якого сімейне кіно стає схожим на комп'ютерну гру.

**2.Завдання**

Створити додатки для редагування відеопотоку (фільму). Програма повинна мати можливість обрізки відеоряду, мати статистичне вікно відображення різної інформації про відео, а також мати можливість регулювати яскравість відеоряду з наступним його збереженням.

**3.Предметна область**

Кількість кадрів на секунду — це число нерухомих зображень, що послідовно змінюються одне за одним впродовж однієї секунди відеоматеріалу, створюючи ефект руху об'єктів на екрані. Чим більша частота кадрів на секунду, тим плавнішим і природнішим буде здаватися рух. Системи телебачення [PAL](http://uk.wikipedia.org/wiki/PAL) й [SECAM](http://uk.wikipedia.org/wiki/SECAM) використовують 25 кадрів на секунду ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) 25 fps або 25 [Герц](http://uk.wikipedia.org/wiki/%D0%93%D0%B5%D1%80%D1%86)), а система [NTSC](http://uk.wikipedia.org/wiki/NTSC) використовує 29,97 кадри на секунду. Комп'ютерні оцифровані відеоматеріали гарної якості, як правило, використовують частоту 30 кадрів на секунду. Верхня гранична частота мерехтіння, що сприймається [людським мозком](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B7%D0%BE%D0%BA_%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B8), в середньому становить 39-42 Герца й індивідуальна для кожної людини. Деякі сучасні професійні камери можуть знімати з частотою до 120 кадрів на секунду. Застосовуються також спеціальні камери для надшвидкої зйомки з частотою до 1000 кадрів на секунду і вище, така швидкість необхідна для детального вивчення траєкторії польоту [кулі](http://uk.wikipedia.org/wiki/%D0%9A%D1%83%D0%BB%D1%8F_(%D0%B7%D0%B1%D1%80%D0%BE%D1%8F)) або структури вибуху.

Розгортка відеоматеріалу може бути прогресивною ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) progressive) або черезрядковою ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) interlaced, інтерлейс). При прогресивній розгортці всі горизонтальні лінії зображення (рядки) відображаються одночасно. При черезрядковій розгортці показуються почергово парні й непарні рядки (називані також полями кадру). Черезрядкову розгортку було винайдено для показу зображення на [кінескопах](http://uk.wikipedia.org/wiki/%D0%9A%D1%96%D0%BD%D0%B5%D1%81%D0%BA%D0%BE%D0%BF) і використовується зараз для передачі відео по «вузьких» каналах, що не дозволяє передавати зображення у всій якості. Системи [PAL](http://uk.wikipedia.org/wiki/PAL), [SECAM](http://uk.wikipedia.org/wiki/SECAM) й [NTSC](http://uk.wikipedia.org/wiki/NTSC) - це всі системи із черезрядковою розгорткою. Нові цифрові стандарти телебачення, наприклад,[HDTV](http://uk.wikipedia.org/wiki/HDTV) передбачають прогресивне розгорнення. Хоча з'явилися технології, що дозволяють імітувати прогресивне розгорнення при показі матеріалу з інтерлейсом. Черезрядкову розгортку звичайно позначають символом «i» після вказівки вертикальної роздільної здатності, наприклад 720×576і×50 для відео у форматі PAL.

Для придушення неприємних ефектів, що виникають при перегляді черезрядкового відео на порядковому екрані, застосовуються спеціальні математичні методи, іменовані [деінтерлейсінгом](http://uk.wikipedia.org/w/index.php?title=%D0%94%D0%B5%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BB%D0%B5%D0%B9%D1%81%D1%96%D0%BD%D0%B3&action=edit&redlink=1" \o "Деінтерлейсінг (ще не написана)).

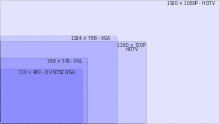
[](http://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Standard_video_res.svg)

Рис.1 Роздільні здатності у порівнянні

За аналогією з роздільною здатністю комп'ютерних [моніторів](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80), будь-який відеосигнал також має роздільну здатність ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) resolution, розділення), горизонтальне і вертикальне, що вимірюється у [пікселях](http://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%BA%D1%81%D0%B5%D0%BB%D1%8C). Звичайний аналоговий телевізійне розділення становить 720×576 пікселів для стандартів [PAL](http://uk.wikipedia.org/wiki/PAL) і [SECAM](http://uk.wikipedia.org/wiki/SECAM), при частоті кадрів 50 Герц (одне поле, 2×25); і 648×486 пікселів для [NTSC](http://uk.wikipedia.org/wiki/NTSC), при частоті 60 Герц (одне поле, 2×29,97). В позначенні 648×480 першим числом позначається кількість крапок у горизонтальній лінії (горизонтальне розділення), а другим числом кількість самих ліній (вертикальне розділення). Новий стандарт високоякісного ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) high-definition) цифрового телебачення [HDTV](http://uk.wikipedia.org/wiki/HDTV) передбачає роздільну здатність до 1920×1080 (тобто 1920 пікселів на лінію, 1080 ліній) при частоті коливань (мелькання, мерехтіння) 60 Герц з прогресивною розгорткою.

Роздільна здатність у випадку трьохмірного відео вимірюється у [вокселях](http://uk.wikipedia.org/wiki/%D0%92%D0%BE%D0%BA%D1%81%D0%B5%D0%BB%D1%8C" \o "Воксель) — елементах зображення, що являють собою крапки (кубики) у тривимірному просторі. Наприклад, для простого трьохмірного відео зараз використається в основному роздільність 512×512×512, демонстраційні приклади такого відео доступні навіть на [PDA](http://uk.wikipedia.org/wiki/PDA).

[](http://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Aspect_ratios.png)

Рис.2 Співвідношення сторін кадру у порівнянні

Співвідношення ширини й висоти кадру ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) aspect ratio) — найважливіший параметр у будь-якому відеоматеріалі. Ще з [1910](http://uk.wikipedia.org/wiki/1910) року кінофільми мали співвідношення сторін екрана 4:3 (4 одиниці завширшки до 3 одиницям у висоту; іноді ще записується як 1,33:1 або просто 1,33). Уважалося що глядачеві зручніше дивитися фільм на екрані такої форми. Коли з'явилося [телебачення](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D0%B1%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), воно успадкувало це співвідношення і майже всі аналогові телесистеми (і, отже, [телевізори](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D0%B2%D1%96%D0%B7%D0%BE%D1%80)) мали співвідношення сторін екрана 4:3. [Комп'ютерні](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80" \o "Комп'ютер) [монітори](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80) також успадкували телевізійний стандарт сторін. Хоча ще в 1950-х роках уява про 4:3 суттєво змінилася. Справа в тому, що поле зору людини має співвідношення аж ніяк не 4:3. Оскільки у людини 2 ока, розташовані на одній горизонтальній лінії, то поле зору людини наближається до співвідношення 2:1. Щоб наблизити форму кадру до природного поля зору людини (і, отже, підсилити сприйняття фільму), був уведений стандарт 16:9 (1,78), що майже відповідає так званому «Золотому перетину». [Цифрове телебачення](http://uk.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B5_%D1%82%D0%B5%D0%BB%D0%B5%D0%B1%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F" \o "Цифрове телебачення) в основному теж орієнтується на співвідношення 16:9. До кінця [XX століття](http://uk.wikipedia.org/wiki/XX_%D1%81%D1%82%D0%BE%D0%BB%D1%96%D1%82%D1%82%D1%8F), після ряду додаткових досліджень у цій області, почали з'являтися навіть і радикальніші співвідношення сторін кадру: 1,85, 2,20 й аж до 2,35 (майже 21:9), що покликані глибше занурити глядача в атмосферу відеоматеріалу.

Комп'ютерна обробка відео — процес редагування [файлів](http://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) [відео](http://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%B5%D0%BE) на комп'ютері, за допомогою спеціальних програм — [відеоредакторів](http://uk.wikipedia.org/w/index.php?title=%D0%92%D1%96%D0%B4%D0%B5%D0%BE%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80&action=edit&redlink=1" \o "Відеоредактор (ще не написана)). Весь процес комп'ютерної обробки відео включає в себе три послідовних і взаємопов'язаних дії:

* захоплення відео,
* монтаж
* фінальне стискання

Для того щоб кінцеве зображення вийшло максимально можливої ​​якості, необхідно робити захоплення відео, при якому здійснюється оцифровка кожного фрагмента даного відео, що дасть можливість покадрово редагувати весь [відеоролик](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D0%BF) і надати готовій роботі додаткові елементи.

Відеомонтаж може здійснюватися двома способами — використовуючи лінійний або нелінійний відеомонтаж:

Лінійний монтаж відбувається частіше в реальному часі. Відео з декількох джерел (програвачів, камер т. д.) надходить через комутатор на приймач (ефірний транслятор, записуючий пристрій). У цьому випадку перемиканням джерел сигналу займається режисер лінійного монтажу. Про лінійному монтажі також говорять у випадку процесу урізання сцен у відеоматеріалі без порушення їх послідовності.

При нелінійному монтажі відео розділяється на фрагменти (попередньо відео може бути перетворено в цифрову форму), після чого фрагменти записуються в потрібній послідовності, в потрібному [форматі](http://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82) на обраний відеоносій. При цьому фрагменти можуть бути урізані, тобто не весь вихідний матеріал потрапляє в цільову послідовність; під час скорочення бувають дуже масштабними.

На останній стадії комп'ютерної обробки відео відбувається стиснення, з необхідною щільністю, відредагованого відео.

Стиснення відео — означає зменшення кількості даних [цифрового відео](http://uk.wikipedia.org/w/index.php?title=%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B5_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE&action=edit&redlink=1). Основане на двох принципах:

надлишок просторової інформації присутньої в кожному кадрі.

надлишок часової інформації: переважна кількість кадрів подібні до попереднього та наступного.

Типові [алгоритми](http://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) стиснення відео починають зі стиснення першого кадру методами [стиснення зображень](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C). Далі виявляється та кодується інформація про відмінності наступного кадру від попереднього. Кадри, що істотно відрізняються від попереднього кодуються окремо.

Відео інформація являє собою тривимірний [масив](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%B8%D0%B2_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)) кольорових [пікселів](http://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%BA%D1%81%D0%B5%D0%BB%D1%8C). При цьому два виміри - це горизонтальне та вертикальне розділення кадрів, а третій - часовий. Кожен кадр таким чином є масивом пікселів із зображенням на даний момент часу.

Відео інформація звичайно містить як просторову, так і часову [надмірність](http://uk.wikipedia.org/w/index.php?title=%D0%9D%D0%B0%D0%B4%D0%BC%D1%96%D1%80%D0%BD%D1%96%D1%81%D1%82%D1%8C&action=edit&redlink=1). Її можна уникнути, якщо проаналізувати і перекодувати різницю як в середині кадру (просторову), так і між кадрами (часова). Просторове стиснення здійснюється, зважаючи на уваги той факт, що людське око не може розрізняти настільки малі відмінності у кольорі, наскільки може відрізняти різницю у яскравості - таким чином подібні за кольором фрагменти можуть бути усереднені, як це має місце при стисненні зображень формату [jpeg](http://uk.wikipedia.org/wiki/Jpeg" \o "Jpeg). Часове стиснення передбачає кодування різниці між сусідніми кадрами, його ефективність пояснюється тим, що велика кількість пікселів буде збігатися в серіях послідовних кадрів.

Переважно стиснення відео здійснюється [з втратами](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7_%D0%B2%D1%82%D1%80%D0%B0%D1%82%D0%B0%D0%BC%D0%B8) - вважається, що значна частина даних не є необхідною для досягнення хорошої якості сприйняття. Занадто велике стиснення, однак, може призвести до помітної втрати якості. В загальному випадку стиснення відео є компромісом між економією дискового простору, якістю відео, і вартістю апаратного забезпечення, необхідного для декомпресії відео в поточному часі.

Існують методи [стиснення без втрат](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B5%D0%B7_%D0%B2%D1%82%D1%80%D0%B0%D1%82), що передбачають можливість відновлення оригінального відео в точності до байту. Проте такі методи використовуються дуже рідко, оскільки стиснення з втрати має значно вищі коефіцієнти при прийнятній якості стиснення. Зокрема, DVD диски та супутникове мовлення також використовують стиснене відео.

Більшість сучасних алгоритмів стиснення використовують [дискретне косінусне перетворення](http://uk.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B5_%D0%BA%D0%BE%D1%81%D1%96%D0%BD%D1%83%D1%81%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F&action=edit&redlink=1) (DCT) або його модифікації для усунення просторової надмірності. Інші методи, такі як [фрактальне стиснення](http://uk.wikipedia.org/w/index.php?title=%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D1%81%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F&action=edit&redlink=1" \o "Фрактальне стиснення (ще не написана)) та [дискретне вейвлет-перетворення](http://uk.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B5_%D0%B2%D0%B5%D0%B9%D0%B2%D0%BB%D0%B5%D1%82-%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F&action=edit&redlink=1), також були об'єктами досліджень, але зараз зазвичай використовуються тільки для компресії нерухомих зображень.

Використання більшості методів стиснення (таких, як дискретні косінусное перетворення та вейвлет-перетворення) спричиняє також використання процесу квантування. Квантування може бути як скалярним, так і векторних, тим не менше, більшість схем стиснення на практиці використовують скалярне квантування завдяки його простоті.

Сучасне цифрове телемовлення стало доступним саме завдяки відео-компресії. Телевізійні станції можуть транслювати не лише відео високої чіткості ([HDTV](http://uk.wikipedia.org/wiki/HDTV)), але і кілька телеканалів у одному фізичному телеканалі (6 МГц).

Хоча більша частина відеозмісту сьогодні транслюється з використанням стандарту стиснення [MPEG-2](http://uk.wikipedia.org/wiki/MPEG-2), використовуються і новіші та ефективніші стандарти стиснення відео - наприклад [H.264](http://uk.wikipedia.org/wiki/H.264) і [VC-1](http://uk.wikipedia.org/w/index.php?title=VC-1&action=edit&redlink=1).

Якість — це [характеристика](http://uk.wikipedia.org/wiki/%D0%A5%D0%B0%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0) обробленого [відео](http://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%B5%D0%BE), як правило, в порівнянні з [оригіналом](http://uk.wikipedia.org/wiki/%D0%9E%D1%80%D0%B8%D0%B3%D1%96%D0%BD%D0%B0%D0%BB). З моменту запису першої відео послідовності, було розроблено велику кількість систем обробки відео. Різні системи можуть надавати різний вплив на відео послідовність, таким чином вимірювання якості відео — це дуже важлива задача.

За часів [аналогових відеосистем](http://uk.wikipedia.org/w/index.php?title=%D0%90%D0%BD%D0%B0%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2%D0%B5_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE&action=edit&redlink=1) було можливо вимірювати якість систем обробки відео шляхом вимірювання частотного відгуку системи на тестовий сигнал.

Зараз, коли [цифрове відео](http://uk.wikipedia.org/w/index.php?title=%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B5_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE&action=edit&redlink=1) витіснило аналогове, виникла необхідність у зміні методів оцінки якості. Продуктивність систем обробки цифрового відео може дуже сильно змінюватися і залежити від динамічних характеристик [вхідного відео сигналу](http://uk.wikipedia.org/wiki/%D0%92%D1%85%D1%96%D0%B4%D0%BD%D0%B8%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) (таких, як рух). Тому якість цифрового відео має вимірюватися на відео послідовностях, які можуть отримувати користувачі.

Об'єктивні техніки вимірювань — це [математичні моделі](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C), які вдало моделюють результати суб'єктивної оцінки якості, вони засновані на критеріях і метриках, що можуть бути виміряні об'єктивно. Об'єктивні методи класифікуються згідно з корисністю вихідного відео сигналу, для якого забезпечується висока якість. Тому вони класифікуються за трьома категоріями: повні референсні методи, скорочені референсні методи і нереференсні методи.

Найтрадиційнішим методом вимірювання якості системи обробки цифрового відео (таких як відеокодеки DivX, XviD)) є вимірювання відношення сигналу до шуму та пікове відношення сигналу до шуму між вихідним сигналом і сигналом на виході системи. PSNR — це одна з метрик об'єктивного якості відео. Вона може бути автоматично обчислена комп'ютерною програмою. Але гарний [PSNR](http://uk.wikipedia.org/w/index.php?title=PSNR&action=edit&redlink=1) не завжди гарантує гарну якість, через те що [зорова система](http://uk.wikipedia.org/wiki/%D0%97%D1%96%D1%80) людини має нелінійну поведінку. Не так давно було розроблено декілька більш складних і точних метрик, наприклад VQM і SSIM.

Всі розглянуті раніше об'єктивні методи вимагають повторення тестів, що проводяться з результатом [кодування](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), для визначення параметрів [кодування](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), які задовольняють певному рівню очікувань користувача, що робить їх швидкість дуже маленькою, такі методи є дуже складними і непрактичними для реалізації в комерційних додатках.

Тому, більшість досліджень направлено на дослідження нових методів об'єктивної оцінки якості, які дозволять прогнозувати сприймаючий рівень якості закодованого відео перед кодуванням.

Головною метою безлічі об'єктивних метрик оцінки якості є автоматична оцінка передбачуваного сприйняття користувачами обробленого системою відео. Але кращим способом визначення думки користувачів це просто запитати їх! Проте іноді, суб'єктивне вимірювання якості відео є важким завданням, оскільки вимагає досвідчених [експертів](http://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82) для його оцінки. Більшість «вимірювань суб'єктивного якості відео» описані в рекомендаціях ITU-T BT.500. В їх основі лежить Mean Opinion Score використовувана для [аудіо](http://uk.wikipedia.org/wiki/%D0%90%D1%83%D0%B4%D1%96%D0%BE): відео послідовності показуються групі глядачів і потім їх думку усереднюється для того, щоб отримати підсумкову оцінку якості кожної відеопослідовності.

**4.Розбір програми**

При розробці програмного продукту було використано мову програмування C# та IDE Visual Studio 2013 Express Windows.

Було взято її за основу так, як вона має усі необхідні компоненти и драйвера для роботи з відео.

Також був використаний компонент ffmpeg для роботи й редагування відео.

Для запуску програми необхідно запустити VideoWinForms.exe

Після чого відкриється форма програми.

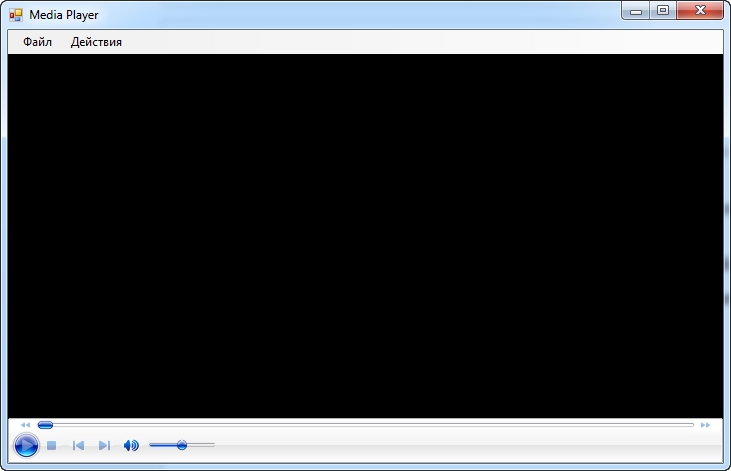


Рис.3 Головна форма програми

Для роботи й редагування відео необхідно відкрити відео доріжку. В інакшому випадку нам не будуть доступні елементи для редагування.

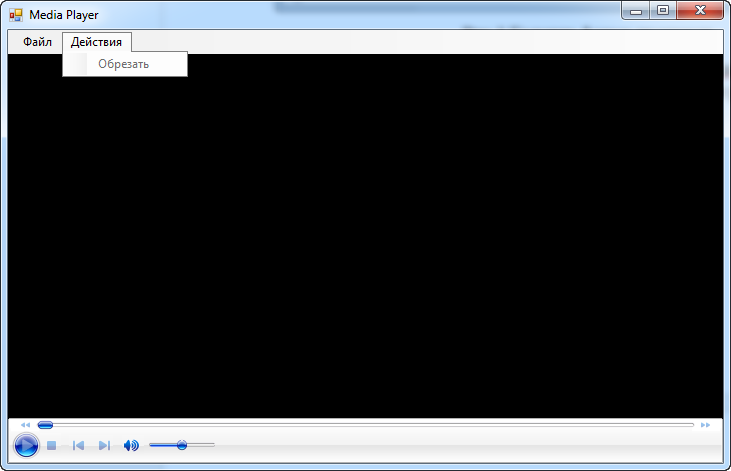


Рис.4 Відкритя редагування

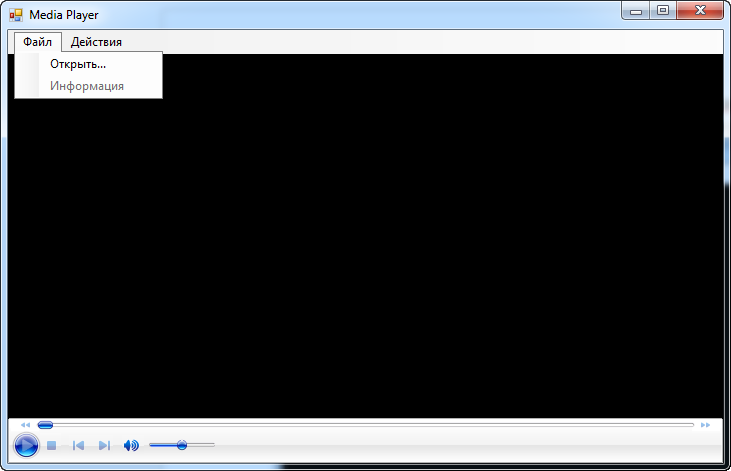


Рис.5 Перегляд Інформації про відео.

Після відкриття відео усі ці елементи нам будуть доступні й відео буде доступне для редагування.



Рис.6 Робота відео

Якщо обрано меню Действия – Обрезать буде запропоновано вибрати куди зберегти нове відео.

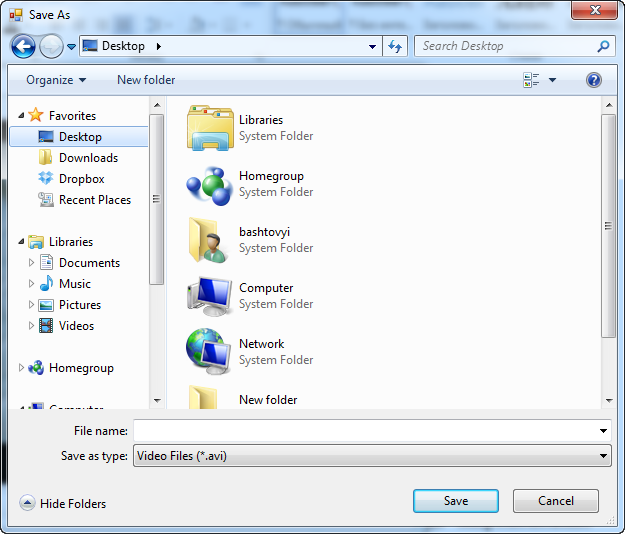


Рис.7 Діалог збереження

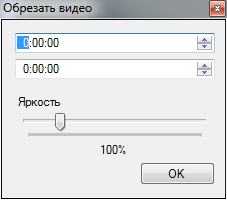
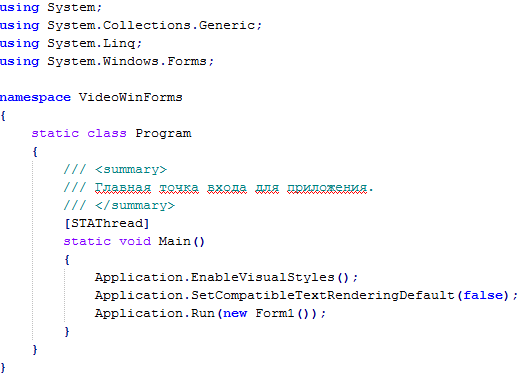


Рис.8 Меню редагування

Розглянемо функції роботи програми.

Головна форма.



Процедура обрізки відео. Використана бібліотека ffmpeg.

internal static string Trim(string inputPath, string outputPath, TimeSpan timeStart, TimeSpan timeEnd, string brightness)

{

var args = String.Format(@"-i {0} -ss {2} -to {3} -vf lutyuv=y=val\*{4} -c:v libx264 -c:a copy -y {1}", inputPath, outputPath, timeStart.ToString(), timeEnd.ToString(), brightness); //msmpeg4

return RunFFMPEG("ffmpeg.exe",args);

}

-i — вхідний файл

-ss – Вихідний файл

- vf – Яркість відео

-c:v Кодек кодування

Copy –y примусове копыювання

Функція перегляду інформації про відеопотік

private void Information\_Load(object sender, EventArgs e)

{

var mediaInfo = Media.GetInfo(\_filePath);

teAudio.Text = Media.AudioInfo(mediaInfo);

teBitrate.Text = Media.BitrateInfo(mediaInfo);

teDuration.Text = Media.DurationInfo(mediaInfo);

teFileName.Text = \_filePath;

teVideo.Text = Media.VideoInfo(mediaInfo);

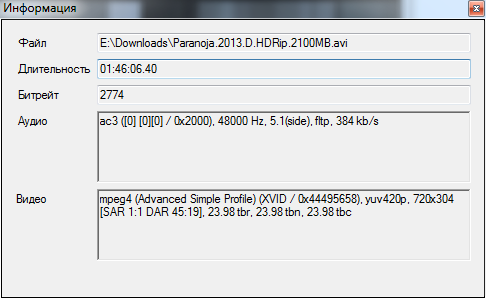
}

Рис.9 Коротка інформація про відео

**5.ВИСНОВОК**

У процесі роботи було вивчено роботу з редагуванням відео, його обрізки й зміною яскравості.

Були вдосконалені навички у програмуванні на C# та VisualStudio.

Під час виконання велика увага приділялась тестуванню програми з різніми відео файлами.

Література:

1. *В. Е. Джакония* Телевидение. — М.,: «Горячая линия — Телеком», 2002. — С. 311—316. — 640 с. — [ISBN 5-93517-070-1](Special:BookSources/5935170701)
2. *Е. М. Голдовский* Кинопроекция в вопросах и ответах. — М.,: «Искусство», 1971. — 220 с.
3. [↑](http://uk.wikipedia.org/wiki/HDTV#cite_ref-1) [«The HDTV Progressive Frame Rate Clarification Initiative»](http://www.webcitation.org/6HfRM6hD8)..
4. Використано матеріали зі статті в [англійській Вікіпедії](http://en.wikipedia.org/wiki/High-definition_television).
5. *MSU Video Group.* [Sixth MPEG-4 AVC/H.264 Video Codecs Comparison - Short Version](http://compression.ru/video/codec_comparison/h264_2010) (англ.). compression.ru.
6. [Суета вокруг кодека. Правда и домыслы о «чудесном» H.264](http://www.secnews.ru/articles/13219.htm) (рус.). Security News (27 марта 2009.
7. [MPEG-4 AVC / H.264 area](http://compression-links.info/MPEG-4_AVC_H264) (англ.). Compression Links. — Список ресурсов и кодеков H.264/AVC[Телевидение высокой чёткости в стандарте H.264 HD](http://truehd.ru/)
8. [All Papers MPEG4/AVC H.264](http://getonebyone.googlepages.com/media_h264)
9. [Использование кодека H.264 в вещании](http://www.videoprotest.com/stati/obmen-opytom-2/item/219-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%B4%D0%B5%D0%BA-h264-%D0%B2-%D0%B2%D0%B5%D1%89%D0%B0%D0%BD%D0%B8%D0%B8)
10. [DirectX Video Acceleration Specification for H.264/MPEG-4 AVC Multiview Video Coding (MVC), Including the Stereo High Profile](http://download.microsoft.com/download/2/D/0/2D02E72E-7890-430F-BA91-4A363F72F8C8/DXVA_H264_MVC.pdf)

Додаток

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace VideoWinForms

{

public partial class Information : Form

{

string \_filePath;

public Information(string filePath)

{

InitializeComponent();

\_filePath = filePath;

}

private void Information\_Load(object sender, EventArgs e)

{

var mediaInfo = Media.GetInfo(\_filePath);

teAudio.Text = Media.AudioInfo(mediaInfo);

teBitrate.Text = Media.BitrateInfo(mediaInfo);

teDuration.Text = Media.DurationInfo(mediaInfo);

teFileName.Text = \_filePath;

teVideo.Text = Media.VideoInfo(mediaInfo);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using System.Text.RegularExpressions;

namespace VideoWinForms

{

static class Media

{

internal static string Trim(string inputPath, string outputPath, TimeSpan timeStart, TimeSpan timeEnd, string brightness)

{

var args = String.Format(@"-i {0} -ss {2} -to {3} -vf lutyuv=y=val\*{4} -c:v libx264 -c:a copy -y {1}", inputPath, outputPath, timeStart.ToString(), timeEnd.ToString(), brightness); //msmpeg4

return RunFFMPEG("ffmpeg.exe",args);

}

internal static string GetInfo(string path)

{

var args = String.Format("-i {0}", path);

return RunFFMPEG("ffmpeg.exe",args);

}

internal static string GetStreamInfo(string path)

{

var args = String.Format("-show\_streams -i {0}", path);

return RunFFMPEG("ffprobe.exe", args);

}

private static string RunFFMPEG(string fileName, string arguments)

{

ProcessStartInfo startInfo = new ProcessStartInfo();

startInfo.FileName = fileName;

startInfo.Arguments = arguments;

startInfo.RedirectStandardOutput = true;

startInfo.RedirectStandardError = true;

startInfo.UseShellExecute = false;

startInfo.CreateNoWindow = true;

Process processTemp = new Process();

processTemp.StartInfo = startInfo;

processTemp.EnableRaisingEvents = true;

try

{

processTemp.Start();

var log = processTemp.StandardError.ReadToEnd();

return log;

}

catch (Exception e)

{

throw;

}

}

internal static TimeSpan GetDuration(string path)

{

var duration = DurationInfo(GetInfo(path)).Split(new char[] { ':', '.' });

return new TimeSpan(int.Parse(duration[0]), int.Parse(duration[1]), int.Parse(duration[2]));

}

internal static string DurationInfo(string info)

{

Regex re = new Regex("[D|d]uration:.((\\d|:|\\.)\*)", RegexOptions.Compiled);

Match m = re.Match(info);

return m.Groups[1].Value;

}

internal static string BitrateInfo(string info)

{

Regex re = new Regex("[B|b]itrate:.((\\d|:)\*)", RegexOptions.Compiled);

Match m = re.Match(info);

return m.Groups[1].Value;

}

internal static string AudioInfo(string info)

{

Regex re = new Regex("[A|a]udio:.\*", RegexOptions.Compiled);

Match m = re.Match(info);

return m.Value.Replace("Audio: ", "");

}

internal static string VideoInfo(string info)

{

Regex re = new Regex("[V|v]ideo:.\*", RegexOptions.Compiled);

Match m = re.Match(info);

return m.Value.Replace("Video: ", "");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace VideoWinForms

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace VideoWinForms

{

public partial class TrimForm : Form

{

string \_inputPath;

string \_outputPath;

public TrimForm(string inputPath, string outputPath, TimeSpan maxDuration)

{

InitializeComponent();

\_inputPath = inputPath;

\_outputPath = outputPath;

timeStart.MinDate = timeEnd.MinDate = new DateTime(2001, 1, 1, 0, 0, 0);

timeStart.MaxDate = timeEnd.MaxDate = new DateTime(2001, 1, 1, maxDuration.Hours, maxDuration.Minutes, maxDuration.Seconds);

}

private void btOk\_Click(object sender, EventArgs e)

{

var brightness = ((double)trackBar1.Value / 100).ToString(System.Globalization.CultureInfo.InvariantCulture);

var log = Media.Trim(\_inputPath, \_outputPath, timeStart.Value.TimeOfDay, timeEnd.Value.TimeOfDay,

brightness);

if (MessageBox.Show("Показать информацию?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)

MessageBox.Show(log);

this.Close();

}

private void trackBar1\_ValueChanged(object sender, EventArgs e)

{

percentageLabel.Text = trackBar1.Value + "%";

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Diagnostics;

namespace VideoWinForms

{

public partial class Form1 : Form, IMessageFilter

{

public Form1()

{

InitializeComponent();

}

private void OpenFile()

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Multiselect = false;

if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

\_inputPath = dialog.FileName;

}

axWindowsMediaPlayer1.URL = \_inputPath;

}

private string \_inputPath;

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFile();

Application.AddMessageFilter(this);

infoToolStripMenuItem.Enabled = true;

trimToolStripMenuItem.Enabled = true;

}

private void trimToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog saveDialog = new SaveFileDialog();

saveDialog.Filter = String.Format("Video Files (\*{0})|\*{0}", System.IO.Path.GetExtension(\_inputPath));

if (saveDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

if (\_inputPath != saveDialog.FileName)

{

using (var trimForm = new TrimForm(\_inputPath, saveDialog.FileName, Media.GetDuration(\_inputPath)))

{

trimForm.ShowDialog();

}

}

else MessageBox.Show("Невозможно сохранить файл в исходный файл");

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (axWindowsMediaPlayer1.playState == WMPLib.WMPPlayState.wmppsPlaying &&

e.Alt && e.KeyCode == Keys.Enter)

{

axWindowsMediaPlayer1.fullScreen = true;

}

}

public bool PreFilterMessage(ref Message m)

{

if (m.Msg == 0x0100)

{

Keys keyCode = (Keys)(int)m.WParam & Keys.KeyCode;

if (keyCode == Keys.Escape)

{

axWindowsMediaPlayer1.fullScreen = false;

}

return true;

}

return false;

}

private void infoToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Показать информацию?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)

MessageBox.Show(Media.GetStreamInfo(\_inputPath));

using (var info = new Information(\_inputPath))

{

info.ShowDialog();

}

}

}

}