

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ**

**ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ
ХАРЧОВИХ ТЕХНОЛОГІЙ**



Артеменко С. В.

ОСНОВИ МУЛЬТИМЕДІА

Навчальний посібник

Одеса 2015

Артеменко С. В. Основи мультимедіа: Навчальний посібник. — Одеська національна академія харчових технологій, 2015. — __ с.

Навчальний посібник розроблено згідно з робочою навчальною програмою дисципліни для студентів, які навчаються за напрямом 050102 «Комп'ютерна інженерія» денної форми навчання. Він висвітлює теоретичний матеріал з дисципліни, містить контрольні питання для більш ґрунтовного його засвоєння студентами. Розглянуті загальні поняття мультимедіа, основні формати зберігання різних мультимедійних даних, деякі алгоритми, які знаходять широке застосування в реалізації технологій мультимедіа. Окремого розгляду заслуговує кожна зі складових мультимедіа: текст, графіка, анімація, звук, відео. В кінці посібника наведені джерела інформації для більш детального вивчення запропонованих тем, а також додаткових тем, що не були винесені на лекції.

Посібник може виявитися корисним для всіх, що цікавиться питаннями мультимедіа.

Рецензент: Шестопапов С. В., к. т. н., ст. викладач кафедри інформаційних систем та мереж ОНАХТ

Розглянуто та рекомендовано до видання на засіданні кафедри інформаційних систем та мереж.

Протокол № _ від __ _____ 2015 р.

Розглянуто та рекомендовано до видання на засіданні науково-методичної комісії з напрямку підготовки 050102 «Комп'ютерна інженерія»

Протокол № _ від __ _____ 2015 р.

©ОНАХТ, 2015

Зміст

Передмова	6
Тема 1. Вступ до мультимедіа	7
Поняття мультимедіа і мультимедійної системи	7
Базова архітектура мультимедійної системи	8
Види мультимедійних даних	9
Яку систему вважати мультимедійною?	10
Лінійне і нелінійне мультимедіа	10
Класифікація програмного забезпечення мультимедіа	11
Види прикладних мультимедійних систем	11
Контрольні питання	12
Тема 2. Засоби розробки мультимедійних систем	13
Класифікація засобів розробки мультимедійних систем	13
Концепція мультимедійної платформи	13
Огляд розповсюджених мультимедійних платформ	15
Контрольні питання	18
Тема 3. Основи кодування графіки	19
Загальні відомості про комп'ютерну графіку	19
Растрова двовимірна графіка	20
Векторна двовимірна графіка	20
Тривимірна графіка	21
Воксельні моделі	21
Узагальнений формат растрової графіки	22
Поняття глибини кольору. Кодування пікселя	23
Короткий огляд популярних форматів зображень	25
Контрольні питання	26
Тема 4. Стиск мультимедійних даних	27
Поняття стиску даних	27
Принципи стиску даних	27
Види стиску даних	27

Коефіцієнт стиску	28
Допустимість втрат при стиску	29
Системні вимоги алгоритмів	29
Алгоритми стиску даних невідомого формату	30
Кодування довжин серій (RLE).....	31
Словникове кодування — алгоритми сімейства LZ	32
Контрольні питання.....	33
Тема 5. Комп'ютерна анімація	34
Загальні відомості.....	34
Визначена анімація.....	34
Процедурна анімація	35
Програмована анімація	36
Конструктори анімацій	36
Зберігання анімації	36
Застосування анімації.....	36
Контрольні питання.....	37
Тема 6. Кодування аудіоданих	38
Основи цифрового представлення звуку	38
Антіалайзинг і дитеринг	39
Якість звуку.....	40
Швидкість потоку (бітрейт).....	40
Огляд форматів, що використовуються для зберігання цифрового звуку ..	41
Опис структури формату RIFF WAVE.....	42
Контрольні питання.....	48
Тема 7. Кодування відеоданих	49
Основні поняття комп'ютерного відео.....	49
Поняття розгорнення відеосигналу	50
Поняття медіаконтейнера	51
Огляд розповсюджених медіаконтейнерів.....	51
Поняття кодека.....	55
Принципи стиску відео. Технологія компенсації руху.....	56

Контрольні питання.....	57
Тема 8. Стандарт опису мультимедійних даних MPEG-7.....	58
Призначення стандарту MPEG-7	58
Структурні складові MPEG-7	59
Контрольні питання.....	60
Список прийнятих скорочень.....	61
Література.....	62
Основна література.....	62
Додаткова література	62

Передмова

Мультимедіа сьогодні стає невід'ємною складовою людського життя. Завдяки мультимедіа світ інформаційних технологій перетворюється на наших очах неймовірним чином: він стає легко сприйманим, яскравим, захоплюючим, а іноді і приголомшуючим.

Системи мультимедіа після своєї появи зробили воістину революційні зміни практично у всіх галузях діяльності людини: в освіті, комп'ютерному навчанні і тренінгах, у багатьох сферах професійної діяльності, науки, мистецтва, у сфері розваг (комп'ютерні ігри, відео, аудіо) тощо.

Поява систем мультимедіа обумовлена як вимогами практики, так і розвитком теорії в галузі інформаційних технологій. Однак, різкий ривок, що відбувся в цьому напрямку за останні роки, забезпечений насамперед розвитком технічних і системних засобів. Це і прогрес у технічному і технологічному розвитку обчислювальної техніки: постійно зростають об'єми оперативної та зовнішньої пам'яті, швидкодія, графічні можливості, і досягнення в області «некомп'ютерної» відеотехніки, лазерних дисків, а також їхнє масове впровадження. Важливу роль у цьому прогресі відіграє також розробка методів швидкого та ефективного стиску даних.

Сучасні персональні комп'ютери в масі своєї надають апаратну основу для більшості класичних застосувань мультимедіа. Сучасні комп'ютерні застосування професійного рівня прагнуть якнайшвидше й впевненіше завоювати популярність серед користувачів, привернути їхню увагу, зайняти відповідну частину ринку. І елементи мультимедіа дуже цьому сприяють. Тому вивчення питань, пов'язаних з мультимедійними технологіями, системами, засобами розробки надзвичайно важливі й актуальні для будь-яких фахівців комп'ютерних галузей.

Тема 1.

Вступ до мультимедіа

Поняття мультимедіа і мультимедійної системи

Термін "мультимедіа" утворений з латинських слів *multum* і *medium* і може бути дослівно перекладений як «безліч способів впливу». Поняття мультимедіа багатогранно: даним терміном позначається ціла низка різних його аспектів.

Перший аспект мультимедіа відповідає певній концепції роботи споживача з інформацією. Мультимедіа в цьому значенні являє собою одночасне використання різних форм її представлення і обробки в деякому єдиному об'єкті-контейнері. Найбільш типовим представником мультимедійного програмного забезпечення є комп'ютерна гра. Мультимедіа в зазначеному вище сенсі являє собою сам факт одночасної роботи гравця з графічною і звуковою інформацією, з обов'язковою наявністю інтерактивності, що забезпечує зворотний зв'язок гравця з ігровим світом, його вплив на ігрову ситуацію.

Другий аспект мультимедіа — технологічний. Із цього погляду поняття мультимедіа може трактуватися як безліч технологій, форматів і методів представлення, обробки і зберігання різноманітної інформації, що робить вплив на різні органи почуттів людини. У цьому змісті мультимедіа стосовно до комп'ютерної гри буде означати весь комплекс технологій кодування і передачі відповідних мультимедійних даних, які забезпечують ігровий процес (зберігання інформації про ігрові рівні, заставки, звуковий супровід гри, її декодування і вивід, ввід керуючої інформації від гравця тощо).

Третє — досить ємне і конкретне — розуміння терміну «мультимедіа» стосується комплексу апаратних і програмних засобів, що дають можливість користувачеві (споживачеві інформації) працювати в діалоговому режимі з різнорідними даними, організованими у вигляді єдиного інформаційного середовища (мультимедійними даними). Це визначення відповідає розумінню мультимедіа як *мультимедійної системи* (ММС).

Термін мультимедіа іноді використовується також для позначення власне мультимедійної інформації (мультимедіа-даних, ММД), а також її носіїв, що дозволяють зберігати значні обсяги даних і забезпечувати досить швидкий доступ до них. В останньому випадку термін мультимедіа може означати, що відповідне обладнання (наприклад, комп'ютер) може використовувати такі носії і надавати інформацію користувачеві всіма доступними способами з необхідною продуктивністю.

З урахуванням вищезазначеного, можна сформулювати два варіанти інтерпретації поняття мультимедіа-системи.

Перший варіант припускає розгляд ММС як сукупності програмного і апаратного забезпечення мультимедіа без включення в нього самих ММД. Він описується умовною формулою

$$MM = MMC + MMD = (MMPZ + MMAO) + MMD, \quad (1)$$

де MMPZ — програмне забезпечення мультимедіа (що не включає мультимедійних даних);

MMAO — апаратне забезпечення мультимедіа;

MMD — мультимедійні дані.

Другий варіант припускає розглядання ММД як складової частини ММС і описується умовною формулою

$$MM = MMC = MMD + MMPZ + MMAO. \quad (2)$$

ММС у цьому випадку виступає як більш цілісне, комплексне утворення.

Необхідність урахування інтерактивності може зробити природнім включення в ММС людини-користувача (К) як елемента людино-машинної системи:

$$MMC = MMD + MMPZ + MMAO + K. \quad (3)$$

Однак у цьому випадку лише частина ММС, що не включає користувача, являє собою самостійний об'єкт проектування.

Базова архітектура мультимедійної системи

У даному посібнику розглядаються ММС, в основі яких лежить персональний комп'ютер. Відповідно до первинного принципу структурування комп'ютерних обчислювальних систем, ММС може бути представлена сукупністю апаратного забезпечення (АЗ) і програмного забезпечення (ПЗ), які мають безпосереднє відношення до мультимедіа.

Як апаратне, так і програмне забезпечення мультимедіа умовно може розглядатися як сукупність універсального і спеціалізованого забезпечення. Універсальне забезпечення реалізує первинну, базову функціональність комп'ютера, без якої немислимо активне існування будь-яких комп'ютерних систем, у тому числі ММС. До таких функцій належать, зокрема, накопичення даних, забезпечення доступу до них, завантаження і виконання програм, виведення відеоданих, введення текстових даних, управління графічним курсором та ін. Спеціалізоване забезпечення реалізує суцільно мультимедійні функції, які можуть бути відсутні у немультимедійних системах. До таких

функцій належать, наприклад, відтворення багатоканального звуку, апаратне декодування відео високої чіткості, швидка обробка складної тривимірної графіки, керування віртуальними об'єктами, наприклад, шляхом зміни положення голови та ін.).

У зв'язку з нечіткістю границі між мультимедійними і немультимедійними даними і функціями, розрізнення універсального і спеціалізованого забезпечення ММС є певною мірою умовним і може бути суб'єктивним.

Види мультимедійних даних

До мультимедійних даних (ММД) традиційно відносять такі види даних [Wiki]:

- текст (у кодованому вигляді);
- графіка — статичні зображення штучного і природнього походження;
- анімація і відео — динамічні (рухомі) зображення штучного і природнього походження;
- звук (синтетичної і хвильової природи);
- інтерактивність — дані, що характеризують зворотний зв'язок з користувачем.



Рисунок 1.1. Основні види мультимедійних даних

Перераховані види ММД стосуються таких почуттів людини, як зір, слух і, можливо, дотик. Принципово можливо впливати на будь-які органи почуттів людину, включаючи також запах і смак. Ефектною і вже досить розповсюдженою реалізацією різноманітного мультимедіа, що включає всі види відчуттів, крім смаку, є кінотеатри 4DX та 5D.

З погляду розроблювача до ММД відносяться не тільки дані, які безпосередньо відповідають впливам на органи почуттів людину, але також і службові, допоміжні, проміжні дані (заголовки, результати розрахунків моделей та ін.), які можуть бути необхідні для створення і структурованого оформлення

безпосередніх ММД першого роду. Часто для позначення безпосередніх ММД першого роду використовується термін «контент».

Яку систему вважати мультимедійною?

Саме поняття мультимедіа з'явилося в слов'янськомовному комп'ютерному середовищі наприкінці 80-х років XX століття. Тоді мультимедійні функції були чимось екзотичним для більшості персональних комп'ютерів і їх користувачів, і поняття мультимедіа було досить конкретним і відособленим (персональний комп'ютер зі звуковою платою і приводом CD-ROM). Зараз, коли мультимедіа глибоко проникло в усі сфери комп'ютерного і некомп'ютерного життя, самі поняття мультимедіа і мультимедіа-системи придбали деяку розмитість. Усі сучасні персональні комп'ютери (як настільні, так і портативні) з позицій початку 90-х років минулого століття є мультимедійними, оскільки забезпечують можливість роботи з усіма видами ММД. Багато споконвічно немультимедійних систем отримують мультимедійні можливості, що робить їх більш привабливими в очах користувачів. Таким чином, сучасне апаратне і програмне забезпечення може вважатися мультимедійним у більшому або меншому ступені. Можна запропонувати різні способи оцінки ступеню мультимедійності, наприклад, адитивну функцію вигляду:

$$K_{\text{мм}} = C_{\text{т}} \cdot B_{\text{т}} + C_{\text{г}} \cdot B_{\text{г}} + C_{\text{а}} \cdot B_{\text{а}} + C_{\text{в}} \cdot B_{\text{в}} + C_{\text{з}} \cdot B_{\text{з}} + C_{\text{і}} \cdot B_{\text{і}}, \quad (4)$$

де C_i — ступінь присутності в ММС ММД i -го виду (обумовлена експертним шляхом або за допомогою деякого формального методу);

B_i — вагові коефіцієнти ступеня мультимедійності ММД i -го виду.

У найпростішому випадку вагові коефіцієнти можна прийняти рівними між собою. Однак логіка підказує, що в більшості випадків одні види ММД відіграють більш значну роль у ММС, ніж інші. Наприклад, відео і аудіо відіграють більш важливу роль у мультимедіа, ніж статична графіка, а остання — більш важливу роль, ніж текст. Проте, конкретні значення вагових коефіцієнтів будуть залежати від безлічі окремих умов, серед яких — думка розроблювачів і користувачів даної ММС, завдання, які розв'язує дана ММС, та ін.

Лінійне і нелінійне мультимедіа

Наявність у ММС останньої категорії ММД — інтерактивності — дозволяє розділити ММС на дві категорії: лінійні і нелінійні (гіпермедіа). *Лінійні* ММС забезпечують потокове виведення (відтворення) відповідних ММД, так що користувач відіграє роль пасивного приймача інформації. Типовим прикладом

лінійної ММС є звичайне кіно: глядач не може впливати на хід дій фільму (а в кінотеатрі — навіть регулювати параметри відтворення, наприклад, гучність звуку або яскравість зображення). У *нелінійних* ММС присутня інтерактивність, що дозволяє користувачеві, взаємодіючи із ММС, брати участь у відповідному мультимедійному процесі, впливати на хід виводу спочатку закладених ММД і генерування нових.

Класифікація програмного забезпечення мультимедіа

Подібність мультимедіа як програмно-апаратних комплексів з універсальними обчислювальними системами проявляється також у наявності схожої ієрархії використововуваного програмного забезпечення (ПЗ). Мультимедійне ПЗ (ММПЗ) можна, насамперед, підрозділити на системне і прикладне.

Завданням системного ММПЗ є створення інфраструктури функціонування прикладного ММПЗ у вигляді реалізації основних технологій мультимедіа (формати контейнерів і потоків даних, алгоритми кодування і декодування ММД, описові структури, інтерфейси доступу до мультимедійних обладнань та ін.). Класичним прикладом системного ММПЗ є мультимедійні кодеки.

Прикладне ММПЗ створюється для розв'язання конкретних, прикладних мультимедійних завдань кінцевого користувача. Види прикладного ММПЗ розглянуті далі.

В окрему категорію можна виділити інструментальне ММПЗ, завданням якого є підтримка процесів розробки прикладних ММС. Розповсюдженими представниками інструментального ММПЗ є так звані авторські системи (наприклад, Adobe Director), які можуть бути орієнтовані на створення конкретних ММС (спеціалізовані) або ММС низки різних категорій (універсальні).

Види прикладних мультимедійних систем

Мультимедіа знаходить своє застосування у всіляких областях людської діяльності, включаючи рекламу, мистецтво, освіту, індустрію розваг, техніку, медицину, математику, бізнес, наукові дослідження тощо. Таким чином, ММС можна умовно класифікувати за призначенням.

Досить розповсюдженою категорією ММС є інформаційні, основним завданням яких є зберігання і подання необхідної інформації в найбільш наочному і зрозумілому вигляді. Такі ММС знаходять найширше застосування в освіті, виробництві, керуванні персоналом. До таких ММС відносяться

комп'ютерні навчальні курси, мультимедійні довідники і енциклопедії, презентації.

У сфері навчання і розваг широко застосовуються комп'ютерні симулятори, завданням яких є імітація певного оточення, зовнішніх умов, обстановки з метою навчити певних навичок.

Свого роду навчання спостерігається в комп'ютерних іграх — мабуть, найпоширеніших і яскравих представниках прикладних ММС.

У математичних і наукових дослідженнях мультимедіа в основному використовується для моделювання. Наприклад: учений може глянути на молекулярну модель якої-небудь речовини і маніпулювати нею для того, щоб одержати іншу речовину. Опис подібних досліджень можна знайти в спеціалізованих журналах, таких як Journal of Multimedia.

У зв'язку з повсюдним поширенням Інтернету і веб-застосувань з'явилася безліч прикладних інтернет-ММС: інтернет-магазини і каталоги, інтернет-університети і центри тестування знань, програми для аудіо- і відео-конференцій, відео-хостинги та багато інших ресурсів.

Контрольні питання

1. Яка етимологія терміна «мультимедіа»?
2. Що по суті являє собою мультимедійна система?
3. Які складові мультимедійної системи?
4. Які завдання можуть вирішувати мультимедійні системи?
5. Які існують типи мультимедійних даних?
6. Які існують види прикладних мультимедійних систем?
7. У чому полягає основна відмінність нелінійних мультимедійних систем від лінійних?
8. Як можна оцінити ступінь мультимедійності прикладної програми?

Тема 2.

Засоби розробки мультимедійних систем

Класифікація засобів розробки мультимедійних систем

Як відзначалося вище, у найбільш загальному випадку під ММС як об'єктом розробки розуміється сукупність апаратного і програмного забезпечення, а також відповідних мультимедійних даних. У якості початкових умов приймемо для подальшого викладу, що є необхідне апаратне забезпечення мультимедіа являє собою деякий набір стандартних пристроїв, з'єднаних стандартними апаратними інтерфейсами, і що забезпечується необхідна підтримка з боку ПЗ найнижчого рівня: ОС і драйверів пристроїв. У цьому випадку об'єктом розробки стає ПЗ більш високого рівня, що реалізує необхідний програмний функціонал ММС.

Також, як і при розробці програмних систем взагалі, для розробки ММС можуть застосовуватися різні засоби, що реалізують необхідну мультимедіа-функціональність відповідного рівня доступу. До цих засобів, зокрема, можна віднести:

- функції і бібліотеки мультимедійних функцій;
- класи і бібліотеки класів, що описують мультимедіа-об'єкти;
- компоненти і бібліотеки компонентів мультимедійних застосувань;
- мультимедійні платформи (фреймворки);
- універсальні середовища програмування;
- дизайнерські засоби створення мультимедіа-моделей;
- середовища проектування мультимедіа-підсистем певного виду;
- застосування обробки ММД певного виду;
- авторські системи створення ММС різного призначення.

З перерахованих перші три види засобів реалізують так звану інфраструктуру мультимедіа — функції і можливості низького рівня (такі як кодування-декодування потоків, мультиплексування-демультиплексування потоків, фільтрація, розрахунок моделей тощо), які використовуються засобами більш високого рівня.

Концепція мультимедійної платформи

В основі сучасних систем і технологій розробки програмних систем лежать найважливіші концепції повторного використання і грамотного структурування програмного коду. На сучасному етапі розробки ці концепції знаходять

найбільш яскраве втілення в так званих програмних платформах (фреймворках). Програмна платформа в інформаційних системах являє собою програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту [Wiki] шляхом структурування інформаційної системи та її підтримки у вигляді великої кількості різноманітних бібліотек, що реалізують специфічну для даного типу інформаційних систем функціональну інфраструктуру.

Типовими представниками програмних платформ є бібліотеки програмних класів і компонентів, що реалізують інтерфейс прикладного програмування (API), забезпечені візуальними середовищами розробки. До таких програмних платформ відносяться, наприклад, комплекси розробки MFC або .Net-framework + Visual Studio, VCL + Delphi, JDK + Netbeans IDE та ін. Також досить розповсюджений каркасний підхід у веб-програмуванні (Symphony, Yii та ін.) і веб-проектуванні (системи керування сайтами (CMS), такі як Drupal, Joomla та ін.). Ці системи є досить універсальними засобами розробки різноманітних інформаційних систем і, будучи оснащені відповідними мультимедійними бібліотеками, також можуть виступати в ролі засобів розробки ММС.

Проте, в області розробки мультимедіа також утворювалося чимало спеціалізованих програмних каркасів різної спрямованості, що пропонують більший або менший ступінь підтримки розробки і функціонування ММС — так званих мультимедійних платформ (мультимедійних фреймворків). Як правило, мультимедійна платформа складається із системи модулів, що підключаються (кодеків, фільтрів, мультиплексорів і демультиплексорів, модулів виводу на екран, роботи з файлами і таке інше), які можна об'єднати в ланцюжок для конвеєрної обробки аудіо- і відеопотоків. Повноцінні мультимедійні платформи мають зручний інтерфейс прикладного програмування (API), забезпечують можливість роботи з різними мультимедійними форматами файлів і мережних потоків, дозволяють встановлювати і розробляти додаткові модулі, що підключаються (плагіни).

У системі класифікації ММПЗ мультимедійні платформи є представниками системного і інструментального ПО. Вони можуть бути крос-платформними або розрахованими на одну конкретну платформу; пропрієтарними або вільними. Далі наводиться огляд найбільш значимих представників даної категорії ПО. Спочатку даний огляд мультимедійних платформ для операційних систем Windows і MacOS, потім пропрієтарних крос-платформних і, нарешті, вільних крос-платформних.

Огляд розповсюджених мультимедійних платформ

Quicktime

Quicktime являє собою технологію, розроблену в 1991 році компанією Apple для відтворення цифрового відео, звуку, тексту, анімації, музики і панорамних зображень у різних форматах на платформі Macintosh. Після своєї появи в 1991 році дана технологія постійно оновлюється компанією Apple і до цього дня залишається актуальною і сучасною. QuickTime також є мультимедійним фреймворком, який можуть використовувати інші застосування для виконання своїх мультимедійних завдань — наприклад, iTunes, iMovie і Final Cut Pro виробництва самої Apple, а також різне програмне забезпечення сторонніх виробників (зокрема, Adobe Photoshop CS4 і Ulead Video Studio).

До складу продуктів Quicktime входять, зокрема:

- безкоштовний мультимедійний плеєр Quicktime Player, призначений для перегляду безлічі різних типів файлів, включаючи файли відео, аудіо, зображень, графіки і фільмів віртуальної реальності (VR);
- розширена версія плеєра — Quicktime Pro — може створювати і редагувати відео, звук, зберігати і експортувати в різні формати, включаючи H.264;
- програмне забезпечення Quicktime Broadcaster — дозволяє здійснювати пряму трансляцію аудіо- і відеоданих в Інтернеті. [Wiki]

Video for Windows

Програмне забезпечення Video for Windows (VfW) було представлено компанією Microsoft у листопаді 1992 року як технологія, яка дозволяє системі, насамперед Windows, відтворювати цифрове відео. Платформа була розроблена в якості відповіді на технологію Apple Quicktime, яка реалізувала підтримку цифрового відео в платформі Macintosh. Повний платний пакет програмного забезпечення Video for Windows включав ряд програм для редагування і кодування відео, використовуваних у комплексі із платами нелінійного монтажу відео. Був доступний також варіант платформи, призначений тільки для перегляду відео. Спочатку випущена в якості безкоштовного доповнення до Windows 3.1 і Windows 3.11, платформа Video for Windows пізніше стала частиною операційної системи Windows 95.

В якості однієї з технологій Video for Windows був представлений формат файлу Audio Video Interleave (AVI), розроблений для зберігання цифрового відео. Також був реалізований прикладний програмний інтерфейс, який дозволив розроблювачам програмного забезпечення для Windows додавати в

застосування можливість відтворювати цифрове відео і керувати ним. Нарешті, Video for Windows включала комплект програмного забезпечення для ігор і керування цифровим відео [enwiki]. Істотний недолік технології VfW полягав у складності роботи з MPEG-відео.

ActiveMovie i DirectShow

Перед виходом Windows 95 компанія Microsoft стартувала новий проект ActiveMovie для підтримки технологій MPEG, 32-бітної архітектури, а також потокової передачі відео- і аудіоданих. Особливістю нового проекту виявилось використання розробленої Microsoft компонентної об'єктної моделі — Component Object Model (COM). Об'єктами в Activemovie є фільтри, які можуть поєднуватися в загальний граф-фільтр. Фільтри можуть створюватися сторонніми розроблювачами для власних потреб (наприклад, для підтримки функціональності якого-небудь пристрою).

В 1997 році Microsoft інтегрувала ActiveMovie в DirectX і змінила його назву на DirectShow (DShow). У такому варіанті даний мультимедійний фреймворк використовується і сьогодні. Як інтерфейс програмування застосувань потокового мультимедіа, він підтримує різні формати файлів (в основному, виробництва Microsoft) — WAV, AVI, Windows Media; дозволяє Windows-застосуванням управляти широким спектром пристроїв введення-виведення аудіо і відео, включаючи DV- і веб-камери, DVD-пристрої, ТБ-тюнери та ін. [Wiki] Будучи заснований на COM, DirectShow надає загальний інтерфейс роботи з мультимедіа, підтримуваний більшістю мов програмування. Крім того, DirectShow може розширюватися і дозволяє підтримувати пристрої, формати і компоненти обробки сторонніх виробників. Комплект засобів розробки і документації раніше розповсюджувався як частина DirectX SDK; у цей час ці засоби розповсюджуються разом з Platform SDK, яка у свою чергу входить до складу Windows SDK. Microsoft планує поступову заміну DirectShow на Media Foundation (див. далі).

Windows Media

В огляді мультимедійних платформ особливої уваги заслуговує мультимедійний набір Windows Media, розроблений компанією Microsoft з метою створення і поширення аудіо- і відеофайлів для операційної системи Windows. Даний набір надає ряд застосувань, засобів розробки, а також інтерфейс програмування застосувань [Wiki]. До складу набору входять застосування Windows Media Player, Windows Media Encoder, Windows Media Center і Windows Media Services, а також комплект засобів розробки Windows Media

SDK. Підтримуються формати Advanced Systems Format (ASF), Windows Media Audio (WMA), Windows Media Video (WMV), Windows HD Photo (WDP). [Wiki]

Media Foundation

Media Foundation (MF) являє собою мультимедійний фреймворк і інтерфейс програмування застосувань нового покоління, створений корпорацією Microsoft для роботи із цифровим мультимедіа під операційними системами сімейства Windows, починаючи з Vista. Згідно із планами Microsoft, він замінить DirectShow, Windows Media SDK, Directx Media Objects (DMOS) і більш старі мультимедійні API, такі, як Audio Compression Manager (ACM) і Video for Windows (VfW). При цьому протягом деякого часу передбачається паралельне співіснування DirectShow і Media Foundation. В Windows XP і більш старих операційних системах використання MF не планується.

У першому релізі Media Foundation, випущеному з Windows Vista, основна увага приділена якості відтворення аудіо і відео, підтримці відео високої чіткості (ТВЧ), а також більш уніфікованому підходу в здійсненні контролю доступу до цифрових даних і взаємодії засобів захисту авторських прав (DRM) [Wiki].

Flash

Adobe Flash (раніше Macromedia Flash) — це мультимедійна платформа компанії Adobe для створення веб-застосувань або мультимедійних презентацій. Широко використовується для створення рекламних банерів, анімації, ігор, а також відтворення на веб-сторінках відео- і аудіозаписів.

Платформа містить у собі ряд засобів розробки, насамперед інструмент дизайнера Adobe Flash Professional і інструмент програміста Adobe Flash Builder (раніше Adobe Flex Builder), а також програму для відтворення flash-контенту — Adobe Flash Player. Flash-Контент також вміють відтворювати багато плеєрів сторонніх виробників.

Adobe Flash дозволяє працювати з векторною, растровою і обмежено із тривимірною графікою, а також підтримує двоспрямовану потокову трансляцію аудіо і відео. Для програмування в Flash використовується мова програмування ActionScript, заснована на ECMAScript. Для КПК та інших мобільних пристроїв випущена спеціальна «полегшена» версія платформи Flash Lite, функціональність якої обмежена, розраховуючи на можливості мобільних пристроїв і їхніх операційних систем. [Wiki]

Silverlight

Microsoft Silverlight — це програмна платформа, що включає в себе модуль, який підключається для браузера і дозволяє запускати застосування з анімацією, векторною графікою, аудіо- і відеороликами, що характерно для так званих «насичених» Інтернет- застосувань — RIA (Rich Internet Application). У версію Silverlight 2.0 була додана підтримка мов .NET та інтеграція із відповідним середовищем розробки.

Платформа Silverlight реалізована для всіх сучасних версій Windows і Mac OS X, а також для більшості відомих браузерів (Internet Explorer, Opera, Mozilla Firefox, Safari, Google Chrome). Silverlight включено в Windows Phone 7, планується також підтримка мобільних ОС Windows Mobile 6, Symbian (Series 60) і, можливо, інших платформ.

Контрольні питання

1. Які існують засоби підтримки розробки мультимедіа-систем?
2. За якими критеріями можна класифікувати засоби розробки мультимедійних систем?
3. Що являє собою мультимедійна платформа (фреймворк)?
4. Які відомі розповсюджені мультимедійні платформи?

Тема 3.

Основи кодування графіки

Загальні відомості про комп'ютерну графіку

Комп'ютерна графіка (також машинна графіка) — це галузь діяльності, у якій комп'ютери використовуються як інструмент для синтезу (створення) зображень, а також для обробки візуальної інформації, отриманої з реального світу. Також комп'ютерною графікою називають результат такої діяльності. [Wiki]

Перші комп'ютерні програми, що активно використовують графіку, з'явилися ще на початку 60-х років XX століття. В 1961 році під керівництвом програміста С. Рассела була розроблена перша комп'ютерна гра із графікою; в 1963 році А. Сазерлендом створений перший векторний графічний редактор, реалізований на комп'ютері; в 1964 році компанія General Motors представила систему автоматизованого проектування DAC-1, розроблену разом з IBM.

Поступово комп'ютерна графіка міцно увійшла в повсякденне життя людини. До основних сфер застосування технологій комп'ютерної графіки належать:

- графічний інтерфейс користувача;
- спецефекти, візуальні ефекти (VFX), цифрова кінематографія;
- цифрова фотографія і живопис;
- візуалізація наукових і ділових даних;
- комп'ютерні ігри, системи віртуальної реальності (наприклад, тренажери керування літаком);
- системи автоматизованого проектування;
- комп'ютерна томографія.

За числом вимірів простору, у якому розміщено зображення, графіку розділяють на двовимірну і тривимірну.

За способами завдання (представлення, зберігання) зображень графіку розділяють векторну і растрову, хоча іноді відокремлюють ще й фрактальний тип представлення зображень.

Двовимірна графіка (2D — від англ. *two dimensions*, два виміри) оперує із графічними об'єктами у двовимірному просторі, а *тривимірна (3D)*, відповідно, — у тривимірному просторі.

Растрова двовимірна графіка

Растрове двовимірне зображення представляється в пам'яті комп'ютера двовимірним масивом (матрицею) окремих його елементів — пікселів. Кожному пікселю зіставляється певне значення — яскравості, кольору, прозорості — або комбінація цих значень.

У растровому вигляді може бути представлено будь-яке зображення, однак цей спосіб зберігання має свої недоліки, які більш виразні при роботі із графікою «діаграмного» типу (схемами, кресленнями та ін.): найчастіше більший обсяг пам'яті, необхідний для роботи із зображеннями, втрати, що зростають при багатоетапному редагуванні. Без особливих втрат растрові зображення можна лише зменшувати, хоча деякі деталі зображення при цьому зникають, не даючи можливість точно відновити оригінальне зображення.

Незважаючи на недоліки, растрова графіка активно використовується, особливо при роботі з фотографіями або іншими художніми зображеннями, які створені людиною і згодом оцифровані за допомогою сканера.

Векторна двовимірна графіка

Векторна двовимірна графіка представляє зображення у вигляді набору геометричних примітивів. зазвичай в якості примітивів вибираються точки, прямі, окружності, прямокутники, а також, в загальному випадку, сплайни деякого порядку. Об'єктам привласнюються деякі атрибути, наприклад, товщина ліній, колір заповнення замкнених фігур. Малюнок зберігається як набір координат, векторів та інших чисел, що характеризують набір примітивів. При відтворенні об'єктів, що перекриваються, має значення відносний порядок їх розташування.

Зображення у векторному вигляді може без втрат масштабуватися, повертатися, деформуватися. Кожне перетворення векторного зображення виконується шляхом стирання старого зображення (або його фрагмента) і побудови замість нього нового. Математичний опис векторного малюнка при цьому залишається тим самим, змінюються тільки значення деяких змінних, наприклад, коефіцієнтів.

Разом з тим, як ми вже відзначали, не всяке зображення можна представити як набір із примітивів. Векторний спосіб представлення добре підходить для схем і креслень, масштабованих шрифтів, ділової графіки, широко використовується для створення «мальованих» мультфільмів. До недоліків при роботі винятково з векторними зображеннями відноситься також необхідність

перемальовувати (заново будувати) зображення навіть при простому зсуві, коли можна було б просто скопіювати частину растрового масиву.

Тривимірна графіка

Як ми вже відзначали, тривимірна графіка оперує з об'єктами в тривимірному просторі. При цьому зазвичай говорять не про тривимірні зображення, а про тривимірні моделі об'єктів і їх видимих проекціях-зображеннях. Тривимірний об'єкт представляється в пам'яті комп'ютера як набір невеликих ділянок поверхонь — полігонів (векторна 3D-графіка), або як набір елементарних просторових часток — вокселів (растрова 3D-графіка). У якості полігонів найчастіше використовуються трикутники, а в якості вокселів — куби.

Візуальні перетворення *полігональних 3D-моделей* здійснюються за допомогою матриць трьох видів (здебільшого): повороту, зсуву і масштабування. Кожний полігон представляється у вигляді набору з координат його вершин. Координати кожної вершини являють собою трикомпонентний вектор (x, y, z) . Множення вектора на відповідну матрицю дає новий вектор. Результатом такого перетворення з усіма вершинами полігона є новий полігон, а результатом перетворення всіх полігонів — новий об'єкт, повернений, зрушений або масштабований щодо вихідного.

Воксельні моделі

Воксельні тривимірні моделі складаються з окремих елементів — вокселів. Слово «*воксель*» (*voxel*) утворене зі слів «об'ємний» (англ. *volumetric*) і «піксель» (англ. *pixel*). Воксели є аналогами пікселів для тривимірного простору. Воксельні моделі часто використовуються для візуалізації та аналізу медичної і наукової інформації.

Як і у випадку з пікселями, самі по собі воксели не містять інформації про свої координати у просторі. Їхні координати обчислюються з їхньої позиції в тривимірній матриці — структурі, що моделює об'ємний об'єкт або поле значень параметра в тривимірному просторі. Цим воксели відрізняються від об'єктів векторної графіки, для яких відомі координати їх опорних точок (вершин) та інші параметри. Воксельні моделі мають певну роздільність. Кожний воксель має певне значення, наприклад, колір.

Для зберігання воксельної моделі застосовують масив розмірами $X \times Y \times Z$. Нестиснені воксельні моделі (у порівнянні з векторними) займають набагато більше місця в пам'яті для обробки. Наприклад, одна нестиснена модель розміром $256 \times 256 \times 256$ вокселів буде займати пам'ять обсягом від 16 Мбайт, у

той час як для векторної моделі може знадобитися в десятки або навіть сотні раз менше пам'яті.

Узагальнений формат растрової графіки

Файли растрової графіки можуть мати різну структуру, представлену сукупністю блоків певного призначення. Однак низка блоків є необхідними для кодування будь-якого растрового зображення, і тому вони присутні в кожному з форматів. Набір таких блоків можна розглядати як узагальнений формат растрової графіки:

Заголовок	[Палітра кольорів]	Растровий масив
-----------	--------------------	-----------------

Рисунок 3.1. Узагальнений формат растрової графіки

Заголовок описує характеристики файлу растрової графіки та, власне, растрового зображення, такі як логічні ширина і висота зображення, глибина кольору, використовувана колірна система, наявність альфа-каналу та ін.

Растровий масив описує дані самого зображення. Кожний елемент растрового масиву відповідає одному пікселю зображення і містить інформацію про його колір. Якщо палітра кольорів відсутня, то ця інформація безпосередньо кодує колір відповідно до використовуваної колірної системи; при цьому говорять про повнокольорове кодування зображення. Повнокольорове кодування доцільно застосовувати для представлення зображень фотографічного типу, де рідкі або взагалі відсутні пікселі одного кольору. Ця ситуація типова у випадку, коли число можливих кольорів набагато перевищує число пікселів зображення.

Палітра, якщо вона присутня, визначає набір використовуваних у зображенні кольорів; при цьому говорять про палітрову графіку. Використання палітри особливо ефективно при кодуванні діаграмної графіки, коли число використовуваних кольорів набагато менше числа пікселів зображення і, таким чином, у зображенні можуть багаторазово зустрічатися пікселі одного кольору. При цьому в растровому масиві містяться відносно короткі індекси (номера) кольорів з палітри замість відповідних довгих колірних кодів.

Палітра може бути повною, коли вона містить усі можливі кольори, які можна отримати при використанні конкретної колірної системи з певними параметрами, і неповною, коли вона представляє тільки частину можливих кольорів. Повна палітра за визначенням є фіксованою і первинно заданою

незалежно від зображень, які кодуються. У випадку використання неповної палітри доцільно формувати її на основі аналізу конкретного зображення, заповнюючи кольорами, які використовуються в ньому.

Поняття глибини кольору. Кодування пікселя

Глибиною кольору називається кількість двійкових біт, що відводяться на кодоване представлення кольору одного пікселя. Чим більша глибина кольору, тем більше число кольорів можна закодувати. Залежність між числом кольорів K и глибиною кольору Γ описується виразами:

$$K = 2^{\Gamma},$$

$$\Gamma = (\log_2 K) \uparrow,$$

де символом $\uparrow$ позначена операція округлення до цілого в більшу сторону.

При повнокольоровому кодуванні зображення елемент растрового масиву повністю відводиться для кодування колірних компонентів пікселя. У найпростішому випадку повнокольорового кодування глибина кольору збігається з розміром цього елемента в бітах. У загальному випадку в елементі растрового масиву може також утримуватися додаткова інформація, наприклад, про рівень прозорості пікселя.

Можливі варіанти представлення колірних палітр:

- 1-бітний колір ($2^1 = 2$ кольори) — бінарний колір, найчастіше представляється чорним і білим кольорами (або чорний і зелений);
- 2-бітний колір ($2^2 = 4$ кольори) — CGA, градації сірого кольору NeXTstation;
- 3-бітний колір ($2^3 = 8$ кольорів) — застарілі персональні комп'ютери з TV-виходом;
- 4-бітний колір ($2^4 = 16$ кольорів) — відомий як EGA і, меншою мірою, як стандарт VGA з високою роздільністю;
- 5-бітний колір ($2^5 = 32$ кольори) Original Amiga chipset;
- 6-бітний колір ($2^6 = 64$ кольори) Original Amiga chipset;
- 8-бітний колір ($2^8 = 256$ кольорів) — застарілі робочі станції Unix, VGA низької роздільності, Super VGA, AGA;
- 12-бітний колір ($2^{12} = 4096$ кольорів) — деякі системи Silicon Graphics, колір систем NeXTstation і Amiga-систем Нам-режиму.

Зі збільшенням кількості біт у представленні кольору кількість відображуваних кольорів стає непрактично великою для кольірних палітр (при 20-бітній глибині кольору, як правило, потрібно більше пам'яті для збереження повної кольірної палітри, ніж пам'яті для збереження самих пікселів зображення).

8-бітна глибина кольору дозволяє сформувати сильно обмежену, однак «реальну» кольірну схему, у якій 3 біта (8 можливих значень) для червоної (R) і зеленої (G) складових, і два біта, що залишилися, — на піксель для кодування синьої (B) складової (4 можливих значення), дозволяють представити $256 (8 \times 8 \times 4)$ різних кольорів. Нормальне людське око менш чутливе до синьої складової, ніж до червоної і зеленої, тому на синю складову приділяється на один біт менше. Така схема використовувалася в MSX 2-серії комп'ютерів в 1990-х роках.

12-бітний «реальний» колір кодується 4 бітами (16 можливих значень) для кожної R, G і B-складових, що дозволяє представити $4096 (16 \times 16 \times 16)$ різних кольорів. Така глибина кольору іноді використовується в простих пристроях з кольоровими дисплеями (наприклад, у мобільних телефонах).

Колір HighColor або HiColor розроблений для представлення відтінків «реального життя», тобто найбільш зручно сприймається людським оком. Такий колір кодується 15 або 16 бітами. 15-бітний колір використовує 5 біт для представлення червоної складової, 5 для зеленої і 5 для синьої, тобто $2^5 = 32$ можливих значення кожного кольору, які дають $32768 (32 \times 32 \times 32)$ об'єднаних кольорів. 16-бітний колір використовує 5 біт для представлення червоної складової, 5 для синьої і 6 біт для представлення зеленої (оскільки людське око більш чутливе при сприйнятті зеленої складової), відповідно 64 можливих значення. У такий спосіб виходять $65536 (32 \times 64 \times 32)$ кольорів. 16-бітний колір згадується як «тисячі кольорів» («thousands of colors») у системах Macintosh.

Більшість сучасних LCD-дисплеїв відображають 18-бітний колір ($64 \times 64 \times 64 = 262144$ комбінації), але завдяки технології dithering різниця з TrueColor-дисплеями на око незначна.

24-бітна глибина кольору і відповідні зображення одержали назву TrueColor. Такий набір кольорів наближений до кольорів «реального світу», надаючи $16,7$ мільйонів різних кольорів. Такий колір найбільш приємний для сприйняття людським оком різних фотографій, для обробки зображень. 24-бітний колір TrueColor використовує по 8 біт для представлення червоної, синьої і зеленої складових, $2^8 = 256$ різних варіантів представлення кольору для кожного

каналу, або всього 16 777 216 кольорів ($256 \times 256 \times 256$). 24-бітний колір згадується як «мільйони кольорів» («millions of colors») у системах Macintosh.

«32-бітний колір» — це приклад неправильного вживання терміна при описі глибини кольору. Оманом є те, що 32-бітний колір дозволяє представити $2^{32} = 4\,294\,967\,296$ різних відтінків. У реальності 32-бітний колір є 24-бітним (TrueColor) з додатковим 8-бітним каналом, який або заповнений нулями (не впливає на колір), або являє собою альфа-канал, який задає прозорість зображення для кожного пікселя. Причиною, по якій використовують «порожній» канал, є прагнення оптимізувати роботу з відеопам'яттю, яка в більшості сучасних комп'ютерів має 32-бітну адресацію і 32-бітну шину даних.

Наприкінці 1990-х деякі high-end графічні системи, наприклад SGI, почали використовувати більш 8 біт на канал, наприклад 12 або 16 біт. Зрозуміло, така кількість відтінків не є затребуваною при відображенні кольорів, однак програми професійного редагування зображень стали зберігати по 16 біт на канал, надаючи «захист» від накопичення помилок округлення і погрешностей при обчисленні в умовах обмеженої розрядної сітки чисел.

Короткий огляд популярних форматів зображень

BMP (*Windows Device Independent Bitmap*) — найпростіший растровий формат, що представляє в більшості випадків незжате зображення, хоча і підтримує алгоритм стиску RLE. Розроблений компанією Microsoft для зберігання графіки. Є «рідним» форматом у системах Windows.

GIF (*Graphics Interchange Format*) — розроблений в 80-х роках XX ст. спеціально для передачі кольорових зображень по комп'ютерних мережах. Здатний зберігати стислі дані без втрати якості у форматі не більше 256 кольорів. Завдяки своїм перевагам — малий розмір файлу, підтримка прозорого фону і анімації — продовжує використовуватися донині.

PNG (*Portable Network Graphics*) — формат зберігання графічної інформації, що використовує стиск без втрат. Був розроблено з метою замінити формат GIF — має кращий стиск даних, підтримує кольорові зображення аж до 48-бітних, а також забезпечує множину рівнів прозорості пікселів завдяки підтримці альфа-каналу.

JPEG (*Joint Photographic Experts Group*) дозволяє стискати дані повнобарвних зображень (до 24 біт) з досить високою швидкістю і ефективністю. На відміну від GIF і PNG використовує алгоритм стиску із втратами.

RAW (англ. *raw* — сирий) — формат даних, що містить неопрацьовані графічні дані, одержувані з датчиків пристрою для сканування. У таких файлах

утримується повна інформація про збережений сигнал, що дозволяє уникнути первісних втрат інформації.

PSD (*Photoshop Document*) — растровий формат зберігання графічної інформації, що використовує стиск без втрат, використовується програмою Adobe Photoshop. Зберігає зображення з багатьма шарами, додатковими кольорними каналами, масками та іншою інформацією.

Контрольні питання

1. Які існують види комп'ютерної графіки?
2. У чому полягає принцип кодування растрової графіки?
3. У чому полягає принцип кодування векторної графіки?
4. Що являють собою воксельні моделі?
5. Яка структура узагальненого формату растрової графіки?
6. Що таке глибина кольору?
7. Як здійснюється кодування пікселя при різних глибинах кольору?
8. Яке призначення колірної палітри при кодуванні растрової графіки?
9. У чому полягає суть каналного представлення графічної мультимедійної інформації?
10. Які існують розповсюджені графічні формати?
11. Що таке альфа-канал? Які розповсюджені графічні формати його підтримують?

Тема 4.

Стиск мультимедійних даних

Поняття стиску даних

Стиск даних (англ. *data compression*) — алгоритмічне перетворення даних, що здійснюється з метою зменшення їхнього об'єму. Застосовується для більш раціонального використання пристроїв зберігання і передачі даних. Синоніми — упакування даних, компресія, стискаюче кодування джерела. Зворотна процедура називається відновленням даних (розпакуванням, декомпресією).

Стиск заснований на усуненні надмірності, що міститься у вихідних даних. Найпростішим прикладом надмірності є повторення в тексті фрагментів (наприклад, слів природньої або машинної мови). Подібна надмірність зазвичай усувається заміною повторюваної послідовності посиланням на вже закодований фрагмент із зазначенням його довжини. Інший вид надмірності пов'язаний з тим, що деякі значення в даних, які стискаються, зустрічаються частіше за інші. Скорочення об'єму даних досягається за рахунок заміни, даних, що часто зустрічаються, короткими кодовими словами, а рідкісних — довгими (ентропійне кодування). Стиск даних, які не мають властивості надмірності (наприклад, випадковий сигнал або білий шум, зашифровані повідомлення), принципово неможливий без втрат.

Принципи стиску даних

В основі будь-якого способу стиску лежить модель надмірності вихідних даних. Іншими словами, для стиску даних використовуються деякі апріорні відомості про те, якого роду дані стискаються. Не володіючи такими відомостями про джерело, неможливо зробити ніяких припущень про перетворення, яке дозволило б зменшити об'єм повідомлення. Модель надмірності може бути статичною, незмінною для всього повідомлення, що стискається, або будуватися чи параметризуватися на етапі стиску (і відновлення). Методи, що дозволяють на основі вхідних даних змінювати модель надмірності інформації, називаються адаптивними. Неадаптивними є зазвичай вузькоспеціалізовані алгоритми, застосовувані для роботи з даними, що мають добре відомі і незмінні характеристики. Більшість досить універсальних алгоритмів є тією чи іншою мірою адаптивними.

Види стиску даних

Усі методи стиску даних діляться на два основні класи:

- стиск без втрат (*lossless*) — при якому можливо повне відновлення вихідних даних;
- стиск із втратами (*lossy*) — дозволяє відновити дані з викривленнями, зазвичай несуттєвими з погляду сприймання користувачем і подальшого використання відновлених даних.

Стиск без втрат зазвичай використовується для передачі і зберігання текстових даних, комп'ютерних програм, рідше — для скорочення об'єму аудіо- і відеоданих, цифрових фотографій і т. п., у випадках, коли викривлення неприпустимі або небажані. Стиск із втратами відіграє більш важливу роль у мультимедіа, оскільки, як правило, має значно більшу ефективність, дозволяючи, таким чином, сильніше упаковувати мультимедійний контент, зберігаючи при цьому показники його якості на необхідному рівні. Стиск із втратами зазвичай застосовується для скорочення об'єму аудіо- і відеоданих і цифрових фотографій у тих випадках, коли таке скорочення є пріоритетним, а повна відповідність вихідних і відновлених даних не потрібна.

Коефіцієнт стиску

Коефіцієнт стиску — основна характеристика будь-якого алгоритму стиску. Вона зазвичай визначається як відношення об'єму вихідних незжатих даних до обсягу стислих:

$$k = \frac{S_o}{S_c},$$

де k — коефіцієнт стиску, S_o — об'єм вихідних даних, а S_c — об'єм стиснутих.

При такому визначенні коефіцієнт стиску показує число разів, у яке вдалося стиснути дані з використанням певного алгоритму. Чим вищий коефіцієнт стиску, тем алгоритм ефективніший. Слід зазначити: якщо $k = 1$, то алгоритм не робить стиску, тобто вихідне повідомлення виявляється за обсягом рівним вхідному; якщо $k < 1$, то алгоритм породжує повідомлення більшого розміру, ніж незжате, тобто, робить «шкідливу» роботу.

Ситуація з $k < 1$ цілком можлива при стиску. Але навіть коли алгоритм стиску збільшує розмір вихідних даних, легко домогтися того, щоб їхній об'єм гарантоване не міг збільшитися більш, ніж на 1 біт. Тобто зробити так, щоб навіть у самому гіршому випадку мала місце нерівність:

$$k \geq \frac{S_o}{S_o + 1},$$

Робиться це в такий спосіб: якщо об'єм стиснутих даних менший, ніж об'єм вихідних, потрібно повернути стиснуті дані, додавши до них «1», інакше повернути вихідні дані, додавши до них «0».

Коефіцієнт стиску може бути постійним або змінним. У другому випадку він може бути визначений або для кожного конкретного повідомлення, або оцінений за деякими критеріями:

- середній (звичайно по деякому тестовому набору даних);
- максимальний (випадок найкращого стиску);
- мінімальний (випадок найгіршого стиску);

або яким-небудь іншим. Коефіцієнт стиску із втратами при цьому сильно залежить від припустимої погрішності стиску або якості, яка зазвичай виступає як параметр алгоритму. У загальному випадку постійний коефіцієнт стиску здатні забезпечити тільки методи стиску даних із втратами.

Допустимість втрат при стиску

Основним критерієм відмінності між алгоритмами стиску є описана вище наявність або відсутність втрат. У загальному випадку алгоритми стиску без втрат універсальні в тому розумінні, що їхнє застосування безумовно можливе для даних будь-якого типу, у той час як можливість застосування стиску із втратами повинна бути обґрунтована. Для деяких типів даних викривлення не припустимі в принципі. У їхньому числі:

- символічні дані, зміна яких неминуче приводить до зміни їхньої семантики: програми та їхні вихідні тексти, двійкові масиви і т. п.;
- життєво важливі дані, зміни в яких можуть призвести до критичних помилок: наприклад, ті, що отримуються з медичної вимірювальної апаратури або контрольних приладів літальних, космічних апаратів і т. п.;
- проміжні дані, що багаторазово зазнають стиску і відновлення при багатоетапній обробці (графічні, звукові, відеодані) (наявність втрат, несуттєвих у випадку однократної обробки, при багаторазовій обробці приводить до нагромадження втрат і перетворенню їх в істотні).

Системні вимоги алгоритмів

Різні алгоритми стиску можуть вимагати різної кількості ресурсів обчислювальної системи, на яких вони реалізовані:

- оперативної пам'яті (під проміжні дані);
- постійної пам'яті (під код програми і константи);

- процесорного часу.

У цілому, ці вимоги залежать від складності і «інтелектуальності» алгоритму. Загальна тенденція така: чим більш ефективним і універсальним є алгоритм, тем більші вимоги до обчислювальних ресурсів він пред'являє. Проте, у специфічних випадках прості і компактні алгоритми можуть працювати краще за складні й універсальні. Системні вимоги визначають їхні споживчі якості: чим менш вимогливий алгоритм, тим на більш простій, а отже, компактній, надійній і дешевій системі він може бути реалізований.

Оскільки алгоритми стиску і відновлення працюють у парі, має значення співвідношення системних вимог до них. Нерідко можна, ускладнивши один алгоритм, значно спростити інший. Таким чином, можливі три варіанти:

1) Алгоритм стиску вимагає більших обчислювальних ресурсів, ніж алгоритм відновлення — це найпоширеніше співвідношення, характерне для випадків, коли однократно стиснуті дані (звук, відео) будуть використовуватися багаторазово.

2) Алгоритми стиску і відновлення вимагають приблизно рівних обчислювальних ресурсів — найбільш прийнятний варіант для ліній зв'язку, коли стиск і відновлення відбувається однократно на двох її кінцях (наприклад, у цифровій телефонії).

3) Алгоритм стиску суттєво менш вимогливий, ніж алгоритм відновлення — така ситуація характерна для випадків, коли процедура стиску реалізується простим, часто портативним обладнанням, для якого обсяг доступних ресурсів досить критичний, наприклад, космічний апарат або велика розподілена мережа датчиків. Це можуть бути також дані, розпакування яких потрібно в дуже малому відсотку випадків, наприклад запис камер відеоспостереження.

Алгоритми стиску даних невідомого формату

Є два основні підходи до стиску даних невідомого формату:

1) На кожному кроці алгоритму стиску або черговий символ поміщається у вихідний буфер стискаючого кодера як є (зі спеціальним прапором, що позначає, що він не був стиснутий), або група з декількох символів замінюється посиланням на співпадаючу з нею групу із уже закодованих символів. Оскільки відновлення стиснутих у такий спосіб даних виконується дуже швидко, такий підхід часто використовується для створення програм, які самі розпаковуються.

2) Для кожної послідовності символів, яка стискається, однократно або в кожний момент часу збирається статистика її зустрічальності у кодованих

даних. На основі цієї статистики обчислюється ймовірність появи чергового символу (або послідовності символів). Після цього застосовується той чи інший різновид ентропійного кодування, наприклад, арифметичне кодування, або кодування Хаффмана, для представлення тих послідовностей, які часто зустрічаються, короткими кодовими словами, а тих, що рідко зустрічаються — більш довгими.

Кодування довжин серій (RLE)

Кодування довжин серій (англ. *Run-length encoding, RLE*) — простий алгоритм стиску даних, який оперує серіями даних, тобто послідовностями, у яких той самий символ зустрічається кілька разів підряд. При кодуванні рядок однакових символів, що становлять серію, замінюється рядком, який містить сам повторюваний символ і кількість його повторів.

Розглянемо, наприклад, зображення, що містить простий чорний текст на суцільному білому фоні. Тут буде багато серій білих пікселів у порожніх місцях, і багато коротких серій чорних пікселів у тексті. Як приклад наведений якийсь довільний рядок зображення в чорно-білому варіанті. Тут В представляє чорний піксель, а W позначає білий:

WWWWWWWWWWWBWWWWWWWWWWBBBWWWWWWWW
WWWWWWWWWWWWWWWWWWBWWWWWWWWWW

Якщо ми застосуємо просте кодування довжин серій до цього рядка, то отримаємо наступне:

12W1B12W3B24W1B14W

Останній запис інтерпретується як «дванадцять W», «одна V», «дванадцять W», «три V» і т. д. Таким чином, код представляє вихідні 67 символів у вигляді всього лише 18.

Однак у випадку, якщо рядок складається з великої кількості неповторюваних символів, її об'єм може зрости:

ABCABCABCABCDEFFFFFFF

1A1B1C1A1B1C1A1B1C1A1B1C2D1E8F

Проблема вирішується досить просто. Алфавіт, у якому записані довжини серій, розділяється на дві (зазвичай рівні) частини. Алфавіт цілих чисел можна, наприклад, розділити на дві частини: позитивні і негативні. Позитивні використовують для запису кількості повторюваних однакових символів, а негативні — для запису кількості неоднакових.

-12ABCABCABC2D1E8F

Оскільки чисельні типи даних на комп'ютері завжди мають деяку межу, виникає ще одна проблема. Припустимо, ми використовуємо для запису довжин серій тип Signed Char. Тоді ми не можемо записати серію довше 127 символів однією парою "довжина-символ". Якщо підряд записано 256 символів A, їх розділяють на мінімальну кількість груп:

127A127A2A

Кодування, яке використовується для зберігання зображень, оперує із двійковими даними, а не із символами ASCII, як у розглянутому прикладі, однак принцип залишається той же.

Очевидно, що таке кодування є ефективним для даних, що містять велику кількість серій, наприклад, для простих графічних зображень, таких як іконки і графічні малюнки. Однак це кодування погано підходить для зображень із плавним переходом тонів, таких як фотографії.

Розповсюджені формати для упаковки даних за допомогою RLE містять у собі Packbits, PCX і ILBM.

Методом кодування довжин серій можуть бути стиснуті довільні файли із двійковими даними, оскільки специфікації на формати файлів часто містять у собі повторювані байти в області вирівнювання даних. Проте, сучасні системи стиску (наприклад, DEFLATE) частіше використовують алгоритми на основі LZ77, які є узагальненням методу кодування довжин серій і оперують із послідовностями символів виду «BWWBWWBWWBWW».

Звукові дані, які мають довгі послідовні серії байт (такі як низькоякісні звукові семпли) можуть бути стиснуті за допомогою RLE після того, як до них буде застосоване дельта-кодування.

Словникове кодування — алгоритми сімейства LZ

LZ (LZ*) — сімейство алгоритмів словникового стиску даних, що отримало назву по ініціалах двох дослідників — Абрахама Лемпеля (*Abraham Lempel*) і Якоба Зіва (*Jacob Ziv*), що розробили в 1970-х роках алгоритми LZ77 і LZ78. На базі LZ77 і LZ78, змінюючи і комбінуючи їх з іншими методами, дослідниками була створена низка інших алгоритмів сімейства LZ — LZO, LZS, LZW, LZC, LZMA.

Обидва первісних алгоритми — LZ77 і LZ78 — відносяться до словникових методів, на відміну від інших методів зменшення надмірності, таких як RLE і арифметичний стиск. LZ77 є алгоритмом з «ковзним вікном», що еквівалентно неявному використанню словникового підходу, вперше запропонованого в LZ78. Можна сказати, що алгоритми сімейства LZ являють собою більш

складне узагальнення простого і інтуїтивного способу стиску даних, використовуюваного в RLE.

Принцип ковзного вікна полягає в тому, що алгоритм кодування враховує інформацію, що вже раніше зустрічалася, тобто інформацію, яка вже відома для кодувальника і декодувальника (друге і наступні входження деякого рядка символів у повідомленні замінюються посиланнями на їхнє перше входження). Завдяки цьому принципу алгоритми LZ іноді називаються методами стиску з використанням ковзного вікна. Ковзне вікно можна представити у вигляді буфера, який організований так, щоб запам'ятовувати «сказану» раніше інформацію і надавати до неї доступ. Таким чином, сам процес стискаючого кодування згідно LZ77 нагадує написання програми, команди якої дозволяють звертатися до елементів «ковзного вікна», і замість значень послідовності, що стискається, вставляти посилання на ці значення в «ковзному вікні». Розмір ковзного вікна може динамічно змінюватися і становити 2, 4 або 32 Кбайти. Слід також зазначити, що розмір вікна кодувальника може бути меншим або дорівнювати розміру вікна декодувальника, але не навпаки.

На відміну від LZ77, що працює із уже отриманими даними, LZ78 орієнтується на дані, які тільки будуть отримані (LZ78 не використовує «ковзне» вікно, він зберігає словник із вже переглянутих фраз). Алгоритм зчитує символи повідомлення доти, поки підрядок, що накопичується, входить цілком в одну із фраз словника. Як тільки цей рядок перестане відповідати хоча б одній фразі словника, алгоритм генерує код, що складається з індексу рядка в словнику, який до останнього введенного символу містив вхідний рядок, і символу, що порушив збіг. Потім у словник додається введений підрядок. Якщо словник уже заповнений, то з нього перш за все видаляють фразу, яка менш за всіх використовувана в порівняннях.

Контрольні питання

1. Яке призначення операції стиску даних?
2. Яка роль стиску при обробці мультимедійних даних?
3. Які принципи стиску даних?
4. Які існують види стиску даних?
5. Що таке коефіцієнт стиску і як він обчислюється?
6. Коли припустимі втрати при стиску мультимедійних даних?
7. Які системні вимоги алгоритмів стиску?
8. Які існують алгоритми стиску даних невідомого формату?
9. Який принцип алгоритму кодування довжин серій (RLE)?
10. Які принципи словникового кодування (алгоритмів сімейства LZ)?

Тема 5.

Комп'ютерна анімація

Загальні відомості

Комп'ютерна анімація — вид анімації, який створюється за допомогою комп'ютера — являє собою послідовний показ зображень (кадрів, слайдів), у результаті чого створюється ілюзія руху. На сьогодні комп'ютерна анімація набула широкого застосування як в галузі розваг, так і у виробничій, науковій та діловій сферах.

Як похідне від комп'ютерної графіки, анімація підрозділяється на аналогічні види: векторна і растрова анімація; двовимірна і тривимірна (3D) анімація.

Кадри, з яких створюється анімація, можуть бути заздалегідь підготовлені (визначена анімація, лінійна MMC) або породжуватися під час самої анімації (процедурна анімація і, як правило, нелінійна MMC).

Визначена анімація

Існує два основні способи створення визначеної анімації: анімація по ключових кадрах і запис руху.

У першому випадку створення анімації припускає виконання двох етапів: створення ключових кадрів і створення проміжних кадрів. Створення і розміщення ключових кадрів проводиться аніматором у ручному або напівавтоматичному режимі, а проміжні кадри генерує спеціальна програма-фазовщик. Цей спосіб найбільш близький до традиційної мальованої анімації, тільки роль фазовщика бере на себе комп'ютер, а не людина.

При використанні запису руху дані анімації записуються спеціальним устаткуванням з об'єктів, що реально рухаються, і переносяться на їхню імітацію в комп'ютері. Розповсюджений приклад такої техніки — *Motion capture* (захоплення рухів). Актори в спеціальних костюмах з датчиками роблять рухи, які записуються камерами і аналізуються спеціальним програмним забезпеченням. Підсумкові дані про переміщення суглобів і кінцівок акторів застосовують до тривимірних кістяків віртуальних персонажів, чим домагаються високого рівня достовірності їхнього руху. Такий самий метод використовують для переносу міміки живого актора на його тривимірний аналог у комп'ютері.

Процедурна анімація

Процедурна анімація повністю або частково розраховується комп'ютером у режимі реального часу згідно із установленими правилами, законами і обмеженням. Сюди можна включити такі її види:

- Симуляція фізичної взаємодії твердих тел.
- Імітація руху систем часток, рідин і газів.
- Імітація взаємодії м'яких тіл (тканини, волосся).
- Розрахунки руху ієрархічної структури зв'язків (кістяка персонажа) під зовнішнім впливом (*Ragdoll*).
- Імітація автономного (самостійного) руху персонажа. Прикладом такої системи є програма *Euphoria*.

На відміну від визначеної анімації, коли аніматор вручну визначає кожний кадр і всі параметри створюваної анімації, при процедурній анімації результат може бути в деякій мірі непередбачений і при кожному запуску може генерувати різноманітну анімацію.

Процедурна анімація використовується для створення і моделювання системи часток (дим, вогонь, вода), тканини і одягу, динаміки твердих тіл, динаміки волосся і хутра, а також для анімації гуманоїдних і негуманоїдних персонажів.

Одним з видів процедурної анімації є фізика *Ragdoll* (фізика «ганчіркової ляльки»). При використанні фізики *Ragdoll* персонаж складається з послідовності зв'язаних твердих тіл, які запрограмовані з використанням системи діючих на них сил.

Ще одним явним прикладом процедурної анімації є технологія «*Dynamic Motion Synthesis*» (синтез динамічних рухів), яка розроблена компанією *NaturalMotion* і використовується в продуктах *Euphoria* і *Endorphin*. Синтез рухів заснований на моделюванні біомеханіки і моторної нервової системи в режимі реального часу, при цьому використовуються елементи з біології і теорії керування роботами. Синтез рухів дозволяє створювати моделі повністю інтерактивних тривимірних персонажів, які не використовують заздалегідь перелічену анімацію. *Euphoria* використовує технологію *DMS* для генерації анімації в режимі реального часу в комп'ютерних іграх або інших інтерактивних застосуваннях.

Програмована анімація

Комп'ютерна анімація знаходить широке застосування в середовищі Web. При реалізації анімації для цього середовища можливі два варіанти:

- попередня підготовка анімованого зображення, розміщення його на боці сервера і копіювання на клієнт у готовому вигляді;
- написання програми генерації анімації, яка виконуватиметься на боці клієнта, що і створює необхідну анімацію в міру доступу до інформації (без урахування або з урахуванням інтерактивності).

Перевага другого варіанта (програмованої анімації) перед першим полягає в зменшенні розміру вихідного файлу анімації, що завантажується на комп'ютер клієнта, а недолік — у підвищеному навантаженні на процесор клієнта.

Широке застосування в мережі одержали мови, за допомогою яких програмуються рухи анімованих об'єктів. Найбільше поширення з них набули:

- Java-Script — браузерна мова;
- Action-Script — мова роботи з застосуваннями Flash.

Конструктори анімацій

Для створення анімованих зображень існує безліч програм, більш або менш спеціалізованих. Серед них можна згадати Adobe Photoshop, GIMP, Adobe Flash Professional, CoffeeCup, Blender та інші.

Також можливе створення анімації за допомогою цифрового фотоапарата. Для цього використовуються спеціалізовані програми, що забезпечують керування цифровим фотоапаратом через комп'ютер і роботу з отриманими кадрами.

Зберігання анімації

Комп'ютерна анімація може зберігатися в універсальних графічних файлах (наприклад, у форматі GIF або APNG) у вигляді набору незалежних зображень, або в спеціалізованих файлах відповідних пакетів анімації (3ds Max, Blender, Maya і т. п.) у вигляді текстур і окремих елементів, або у форматах, призначених для перегляду (FLIC) і застосування в іграх (Bink). Також, анімація може зберігатися у форматах, призначених для зберігання відео (наприклад, MPEG-4).

Застосування анімації

Комп'ютерна анімація може застосовуватися в комп'ютерних іграх, мультимедійних застосуваннях (наприклад, енциклопедіях), а також для

«пожвавлення» окремих елементів оформлення, наприклад, веб-сторінок і реклами (анімовані банери).

Тривимірна комп'ютерна анімація набуває усе більш широкого застосування в кіноіндустрії.

Контрольні питання

1. У чому полягають характерні риси анімації як виду мультимедійних даних?
2. Які існують різновиди комп'ютерної анімації?
3. У чому полягає сутність визначеної анімації?
4. У чому полягає сутність процедурної анімації?
5. Які види процедурної анімації набули самостійного розвитку?
6. У чому полягає сутність програмувальної анімації?
7. Які застосовуються засоби створення анімації?
8. Які мови програмування застосовуються для створення анімації в середовищі Web?
9. У яких форматах може зберігатися комп'ютерна анімація?
10. Які сфери застосування анімації?

Тема 6.

Кодування аудіоданих

Основи цифрового представлення звуку

Цифровий звук (англ. *Digital audio*) — термін, що вживається на окреслення цифрових технологій роботи зі звуком. У вузькому сенсі слова, цифровий звук являє собою цифровий сигнал, в якому закодовано звук. В більш широкому сенсі поняття цифровий звук охоплює також цифрові технології обробки, зберігання та відтворення звуку.

Цифровий звук принципово відрізняється від аналогового. В аналогових звукових системах (наприклад при запису на грамофонні платівки, магнітофонні касети тощо), акустичні коливання повітря перетворюються на аналогічні за формою електричні за допомогою мікрофона та зберігаються, а відтворення звуку відбувається зворотнім шляхом — через підсилювач та конвертацію в акустичні коливання через гучномовець.

Натомість цифровий звук починається з перетворення аналогового сигналу на електричний імпульсний («on/off») сигнал, що відрізняється від електро-механічного формою. Цей сигнал перекодовується для запобігання помилок при зберіганні чи передачі сигналу.

Як правило, цифровий сигнал отримують з аналогового шляхом його конвертації у цифровий. Цей процес називається оцифруванням (оцифровуванням) і здійснюється за допомогою аналогово-цифрового перетворювача (АЦП). В музичній практиці використовують однак і сигнали, що мають винятково цифрове походження, наприклад, згенеровані на цифрових синтезаторах, в цьому випадку аналогово-цифрове перетворення не здійснюється.

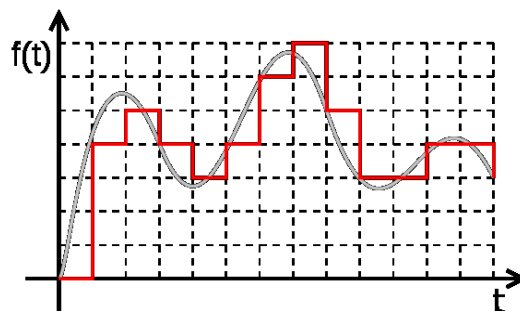


Рисунок 6.1 – Звукова хвиля (крива лінія), представлена в цифровому вигляді (ступінчаста лінія)

АЦП здійснює заміри амплітуди звукового сигналу з певною частотою дискретизації (*sample rate*) та з певною роздільністю (*bit resolution*).

Частота дискретизації f визначає частоту, з якою здійснюються заміри амплітуди сигналу, вимірюється у герцах або кілогерцах (кГц). Згідно з теоремою Котельникова, частота дискретизація повинна бути щонайменше вдвічі більшою за найвищу частоту корисного сигналу. Оскільки людина сприймає звуки частотою до 20 кГц, то для якісного аудіо мінімальне значення частоти дискретизації становить 40 кГц.

Амплітудна роздільність (дискретність квантування) δ — визначає точність, з якою здійснюються заміри амплітуди сигналу. Амплітудна роздільність вимірюється у кількості бітів, що відводяться для запису значення амплітуди. Оскільки 1 біт відповідає одному розряду у двійковій системі, ця величина також називається розрядністю і грає у кодуванні ту ж роль, що і глибина кольору у графіці. Кількість можливих значень амплітуди x та відповідна до неї кількість розрядів δ знаходяться у залежності, що описується співвідношенням $x = 2^\delta$. Таким чином, наприклад, 16-бітова розрядність забезпечує запис $2^{16} = 65\,536$ рівнів амплітуди.

Підвищення обох параметрів уможливлюють більш якісне оцифрування звуку, проте збільшують і обсяги даних. Тому на практиці вживаються різні стандарти дискретизації та розрядності. Наприклад, стандартний Audio CD має частоту дискретизації 44,1 кГц (44 100 семпли в секунду) та 16-бітову розрядність для кожного каналу (стерео). Натомість DVD-Audio може використовувати частоту дискретизації до 192 кГц, а розрядність — до 24 біт.

Антіалайзинг і дитеринг

Якщо записуваний сигнал включає частоти, вищі за максимально припустиму граничну частоту (англ. *Nyquist frequency*), при його оцифруванні виникає ефект накладання спектрів частот (англ. *aliasing*). Для запобігання цього ефекту необхідний фільтр захисту накладання спектрів, що обмежує спектр сигналу граничною частотою.

Іншим небажаним ефектом при оцифруванні звуку є кореляція шуму квантування з квантованим сигналом. При цьому шум квантування сприймається як доволі неприємне спотворення на частоті 3-5 кГц. Для зменшення цього ефекту використовується дитеринг — ефект додавання до сигналу псевдовипадкового сигналу. Хоча загальний рівень шумів при дитерингу збільшується, суб'єктивно сприйманий неприємний ефект зменшується. Сприйману гучність шуму квантування можна зменшити, застосовуючи

дітеринг спільно з нойз-шейпінгом (*noise shaping*). В якості шуму дітеринга найчастіше використовується білий шум з трикутним розподілом амплітуд (TPDF) між -1 і $+1$ LSB (крок квантування). Дітеринг застосовується, зокрема, для запису дисків CD, коли цифровий звук високої роздільності (24 біта) знижується до розрядності CD (16 біт) в процесі мастерингу.

Якість звуку

Завданням як аналогової, так і цифрової систем є відтворення звуку з максимальною якістю. Проте існують ряд перешкод для досягнення бажаного результату. Серед них:

- аналоговий рівень власних шумів, який залежить від ємності та індуктивності, що обмежують смугу пропускання системи, а також опору, що обмежує амплітуду;
- цифровий шум квантування, який залежить від частоти дискретизації, що обмежує смуги пропускання, а також розрядності, що обмежує динамічний діапазон.

Швидкість потоку (бітрейт)

Цифровий звук має поточкову природу. Одним з характеристик цифрового потоку є бітрейт (англ. *bitrate*) — швидкість проходження бітів інформації в одиницю часу. Бітрейт прийнято використовувати при вимірюванні ефективної швидкості передачі інформації з каналу, тобто швидкості передачі «корисної інформації» (крім такої по каналу може передаватися службова інформація, наприклад, стартові й стопові символи при асинхронній передачі або контрольні символи при надлишковому кодуванні).

Бітрейт виражається бітами на секунду (біт/с, bps), а також похідними величинами з префіксами кіло-, мега- і т. д.

Бітрейт може також виступати характеристикою каналу або пристрою — максимальна кількість біт, яку можна передати в одиницю часу.

Бітрейт також є характеристикою стиснених форматів потокового відео і аудіо (наприклад MPEG і MP3), яка виражає ступінь стиснення потоку, і, тим самим, визначає розмір каналу, для якого стиснено потік даних. Найчастіше бітрейт звуку й відео вимірюють у кілобітах у секунду (англ. *kilobit per second* — kbps).

Існує три різновиди бітрейту, які відповідають трьом режимам стиску поточкових даних:

- з постійним бітрейтом (англ. Constant BitRate — CBR)

- зі змінним бітрейтом (англ. Variable BitRate — VBR)
- з усередненим бітрейтом (англ. Average BitRate — ABR)

Розвинений кодек може вибрати потрібний режим стиску, виходячи з параметрів, які задають рівень бажаної якості. При стиску звуку потрібний бітрейт визначається на основі психоакустичної моделі. ABR — різновид VBR, у якій кодек при стисненні дотримується заданого середнього значення з тим, щоб отримати на виході файл заданого розміру.

Огляд форматів, що використовуються для зберігання цифрового звуку

Існує три основні групи аудіоформатів.

- нестиснені формати — такі як WAV, AIFF, AU, PCM;
- формати із стисненням без втрат — FLAC, Monkey's Audio (розширення APE), Shorten, Tom's lossless Audio Kompressor (TAK), TTA, ATRAC Advanced Lossless, Apple Lossless, MPEG-4 SLS, MPEG-4 ALS, MPEG-4 DST, WMA Lossless;
- формати із стисненням з втратами, наприклад MP3, Ogg Vorbis, Musepack, AAC, ATRAC, lossy WMA.

Слід мати на увазі, що до аудіоформатів не відносяться такі музичні формати як MIDI або файли нотних редакторів, які являють собою лише послідовність команд для музичного інструменту, однак не містять інформації власне про звук.

Формат аудіофайлу тісно пов'язаний з алгоритмом роботи аудіокодека. Кодек здійснює кодування чи розкодування звукових даних, а самі дані зберігаються у файлі відповідного звукового формату. Більшість аудіоформатів підтримують лише один тип кодування (формат потоку) звукових даних, проте деякі мультимедійні контейнери (наприклад, MKV або AVI) можуть підтримувати різні типи аудіо- і відео-потоків.

WAV

WAV (від англ. *wave* — хвиля) — формат файла-контейнера, який широко використовується для зберігання запису оцифрованого аудіопотоку. Перша версія формату представлена компанією Microsoft в 1991 р. Формат WAV ґрунтується на структурі універсального контейнерного формату RIFF (див. далі). У системі Windows цей формат найчастіше використовується в якості оболонки для нестисненого звуку (PCM), коли для кожного відліку амплітуди сигналу виділяється задане число біт [Wiki]. Менш поширені випадки, коли в

контейнері WAV розміщується звук, стиснений деяким кодеком (наприклад, MP3).

Внаслідок того, що формат WAV є надбудовою над універсальним багатоплатформним контейнером RIFF, існує кілька різновидів формату WAV для різних платформ і з деякими відмінностями в структурі.

MP3

MP3 (більш точно, англ. MPEG-1/2/2.5 Layer 3; але не MPEG-3) — третій шар формату кодування звукової доріжки MPEG, ліцензований формат файлу для зберігання аудіоінформації [Wiki]. MP3 є одним з найпоширеніших і найпопулярніших форматів цифрового кодування звукової інформації з втратами. Звук у даному форматі може відтворюватися практично у всіх популярних операційних системах, на більшості портативних аудіоплеєрів, а також підтримується всіма сучасними моделями музичних центрів і DVD-програвачів.

Важливою відмінністю формату MP3 від WAV є те, що MP3 не є медіаконтейнером, оскільки даний формат специфікує як спосіб опису параметрів аудіоданих, так і формат представлення самого аудіопотока (але не алгоритм його кодування) у межах одного файлу.

Опис структури формату RIFF WAVE

Загальні відомості про формат WAV

Формат WAV споконвічно використовувався в системі Windows для збереження цифрових аудіоданих. Це найвідоміший і широко підтримуваний формат завдяки популярності платформи Windows і великій кількості написаних для неї програм. Майже будь-яка сучасна програма, що працює зі звуком, може прочитати або записати формат WAV. Далі докладно описуються основні структури даних формату WAV.

Оскільки формат WAV прийшов від операційної системи Windows, у якій традиційно використовувалися процесори Intel, усі значення даних формату зберігаються як Little-Endian, тобто самий молодший значущий байт іде першим.

WAV-файли можуть містити рядки тексту, наприклад мітки секцій, інформаційні коментарі тощо. Рядки зберігаються таким чином, що перший байт вказує кількість байт тексту ASCII у рядку. Приклад формату рядка Wave — 7 ‘e’ ‘x’ ‘a’ ‘m’ ‘p’ ‘l’ ‘e’.

Загальна структура файлу

використовує стандартну структуру RIFF, яка групує вміст файлу з окремих секцій (*chunks*) — формат вибірок аудіоданих, самі аудіодані і т. д. Кожна секція має свій заголовок секції і дані секції. Заголовок секції вказує тип секції і кількість байтів, що містяться в секції. Такий принцип організації дозволяє програмам аналізувати тільки необхідні секції, пропускаючи інші секції, які невідомі або не вимагають обробки. Деякі певні секції можуть мати у своєму складі підсекції (*sub-chunks*). Наприклад, як можна побачити на діаграмі, що описує основний формат WAV-файлу, секції "fmt " і "data" є підсекціями секції "RIFF".

```
Chunk ID "RIFF"
Chunk Data Size

RIFF Type ID "WAVE"

Chunk ID "fmt "
Chunk Data Size

Sample Format Info

Chunk ID "data"
Chunk Data Size

Digital Audio Samples

Заголовок секції (Chunk Header)
Дані секції (Chunk Data Bytes)
```

Одна з особливостей, пов'язана із секціями файлу RIFF, полягає в тому, що адреси початку секцій повинні бути вирівняні на розмір слова (2 байта). Це означає, що загальний розмір секції повинен бути кратний 2. Якщо секція містить непарне число байт даних (невирівняне до 2 байт), то додається додатковий нульовий байт даних у кінець даних секції. Цей додатковий байт не враховується в розмірі секції заголовка, таким чином, програма завжди повинна враховувати вирівнювання для розрахунків зсуву початку наступної секції.

Заголовок WAV-файлу, секція типу RIFF

Перші 8 байт WAV-файлу містять стандартний заголовок секції RIFF, який має ID секції "RIFF" і розмір секції, що дорівнює розміру файлу мінус 8 байт, що використовуються для RIFF-заголовка. Перші 4 байта даних у секції "RIFF" визначають тип ресурсу, який можна знайти в секції. WAV-файли завжди

використовують тип ресурсу "WAVE". Після типу ресурсу (ID "WAVE") ідуть усі секції звукового файлу, які визначають аудіосигнал.

Таблиця 3.1.

Варіанти тем для зображень №№2, 3 і 4

Таблиця 6.1

Значення полів секції RIFF

Зсув	Розмір	Опис	Значення
0x00	4	Chunk ID	"RIFF" (0x52494646)
0x04	4	Chunk Data Size	(file size) — 8
0x08	4	RIFF Type	"WAVE" (0x57415645)
0x10	Wave chunks (секції Wav-Файлу)		

Секції WAV-файлу

Існує досить багато типів секцій, заданих для файлів WAV, але більшість WAV-файлів містять тільки дві з них — секцію формату ("fmt ") і секцію даних ("data"). Це саме ті секції, які необхідні для опису формату вибірок аудіоданих і для зберігання самих аудіоданих. Хоча офіційна специфікація не задає жорсткий порядок проходження секцій, найкращою практикою буде розміщення секції формату перед секцією даних. Багато програм очікують саме такий порядок секцій, і він найбільш розумний для передачі аудіоданих через повільні послідовні канали зв'язку, наприклад, в Інтернеті.

Усі секції формату RIFF і відповідно секції Wave зберігаються у форматі, показаному в таблиці 6.2:

Таблиця 6.2

Формат секцій RIFF і Wave

Зсув	Розмір	Опис
0x00	4	Chunk ID
0x04	4	Chunk Data Size
0x08	Chunk Data Bytes	

Далі описані деякі найбільш важливі типи секцій Wave, формат і значення даних, що містяться в них.

Секція формату — "fmt "

Секція формату містить інформацію про те, як збережені аудіодані і як вони повинні відтворюватися. Інформація містить у собі тип використовуваної компресії, кількість каналів, швидкість видачі вибірок (sample rate), кількість біт у вибірці (bits per sample) та інші атрибути.

Таблиця 6.3

Значення полів секції формату (Wave Format Chunk)

Зсув	Розмір	Опис	Значення
0x00	4	Chunk ID	"fmt " (0x666D7420)
0x04	4	Chunk Data Size	16 + extra format bytes
0x08	2	Compression code	1 - 65,535
0x0a	2	Number of channels	1 - 65,535
0x0c	4	Sample rate	1 - 0xffffffff
0x10	4	Average bytes per second	1 - 0xffffffff
0x14	2	Block align	1 - 65,535
0x16	2	Significant bits per sample	2 - 65,535
0x18	2	Extra format bytes	0 - 65,535
0x1a	Додаткові дані формату (Extra format bytes)		

Ідентифікатор секції (Chunk ID) завжди містить значення "fmt " (0x666D7420) і об'єм даних дорівнює розміру стандартного формату WAV (16 байт) плюс розмір усіх додаткових байтів формату, необхідних для підтримки специфічних форматів звуку, якщо він не містить нестиснених даних PCM.

Код стиску (Compression code) — перше слово даних формату вказує на тип стиску, використовуваного для аудіоданих. У таблиці 6.4 наводиться список кодів стиску, які використовуються у цей час.

Кількість каналів (Number of Channels) вказує, скільки окремих аудіосигналів закодовано в секції даних звуку (wave data chunk). Значення 1 означає монофонічний сигнал, 2 означає стерео, і т. д.

Швидкість вибірок (Sample Rate) — це число вибірок аудіосигнала, що припадають на одну секунду. На цю величину не впливає кількість каналів.

Коди стиску (Compression Codes)

Код	Опис
0 (0x0000)	Unknown
1 (0x0001)	PCM/uncompressed
2 (0x0002)	Microsoft ADPCM
6 (0x0006)	ITU G.711 a-law
7 (0x0007)	ITU G.711 μ -law
17 (0x0011)	IMA ADPCM
20 (0x0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0x0040)	ITU G.721 ADPCM
80 (0x0050)	MPEG
65536 (0xffff)	Experimental

Середня кількість байтів на секунду (Average Bytes Per Second) — величина, що показує, скільки байтів за секунду даних повинне бути пропущене через цифро-аналоговий перетворювач (D/A converter, DAC) під час відтворення файлу. Ця інформація корисна, щоб визначити, чи можуть дані надходити від джерела з потрібною швидкістю, щоб не відставати від відтворення. Ця величина обчислюється за формулою:

$$\text{AvgBytesPerSec} = \text{SampleRate} \cdot \text{BlockAlign}$$

Вирівнювання блоку (Block Align) — кількість байтів на одну вибірку. Ця величина може бути обчислена за формулою:

$$\text{BlockAlign} = \text{SignificantBitsPerSample} \cdot \text{NumChannels} / 8$$

Кількість використовуваних бітів на вибірку (Significant Bits Per Sample) — кількість бітів що формують кожен вибірку сигналу; зазвичай ця величина дорівнює 8, 16, 24 або 32. Якщо число бітів не вирівняне по байту (не ділиться без остачі на 8), кількість використовуваних байтів на вибірку округляється вгору до найменшої кількості байтів. Невикористовувані біти встановлюються в 0 і ігноруються. Такі формати (із числом бітів на вибірку, не кратним 8) зустрічаються рідко.

Додаткові дані формату (Extra Format Bytes) — дана величина вказує, скільки далі йде додаткових даних, що описують формат. Вона зазвичай відсутня, якщо код стиску рівний 1 (uncompressed PCM file), але може бути присутнім і мати будь-яку іншу величину для інших типів стиску, що залежить

від кількості необхідних для декодування даних. Якщо величина не вирівняна на слово (не ділиться без остачі на 2), повинен бути доданий додатковий байт у кінець даних, але величина повинна залишатися невирівняною.

Секція даних — "data"

Секція даних Wave (Wave Data Chunk) містить дані цифрових вибірок аудіосигналу, які можна декодувати з використанням формату і методу компресії, зазначених у секції формату Wave (Wave Format Chunk). Якщо код компресії рівний 1 (нестиснений PCM, Pulse Code Modulation), то дані представлені у вигляді сирих, неперетворених (raw) величин вибірок.

Wav-Файли зазвичай містять тільки одну секцію даних, але секцій може бути декілька, якщо вони містяться в секції списку Wave (Wave List Chunk "wavl").

Таблиця 6.5

Формат секції даних "data"

Зсув	Довжина	Тип	Опис	Значення
0x00	4	char[4]	chunk ID	"data" (0x64617461)
0x04	4	dword	chunk size	залежить від кількості вибірок і компресії
0x08	дані вибірок (sample data)			

Аудіовибірки багатоканального цифрового аудіо зберігаються як дані, що чергуються (interlaced): вибірки каналів, що відносяться до деякого моменту часу, зберігаються послідовно один за одним, перед тим як відбудеться перехід до наступного часу вибірки. Це зроблене з метою можливості послідовного програвання файлу навіть тоді, коли ще не весь файл прочитаний цілком. Це зручно, коли програватиметься великий файл із диска (який не може бути розміщений цілком у пам'яті) або файл передається в послідовному потоці даних через мережне з'єднання. Значення в діаграмі нижче були б збережені в WAV-файлі в тому порядку, як вони перераховані в стовпці значень (від початку до кінця).

Вибірки стерео Wave, що чергуються

Час	Канал	Значення
0	1 (лівий)	0x0053
	2 (правий)	0x0024
1	1 (лівий)	0x0057
	2 (правий)	0x0029
2	1 (лівий)	0x0063
	2 (правий)	0x003C

8-бітні вибірки визначені як значення без знаку (unsigned). Двійкове значення 0 (0x00) відповідає значенню сигналу максимальної негативної амплітуди, значення 255 (0xff) — максимальної позитивної амплітуди, а значення 128 (0x80) — відсутності сигналу. Усі інші бітові розміри вказуються як величини зі знаком (signed). Наприклад, вибірка довжиною 16 біт може приймати значення в діапазоні від -32768 (0x8000) до $+32767$ (0x7FFF), значення 0 відповідає відсутності сигналу.

Як уже було зазначено раніше, усі секції RIFF (включаючи секції WAVE "data") повинні бути вирівняні по розміру на слово (2 байта). Якщо дані вибірок містяться в непарній кількості байтів, у кінець даних вибірок повинен бути доданий нульовий байт. У заголовку секції "data" розмір не повинен враховувати цей байт.

Інші додаткові секції WAV-файлу використовуються досить рідко, тому для реалізації найпростішого програмного аналізу WAV-файлу обробку цих секцій можна вилучити.

Контрольні питання

1. У чому полягає сутність імпульсно-кодової модуляції?
2. У чому полягають сутність і призначення операцій антїалайзингу і дитерингу?
3. Які параметри характеризують якість цифрового звуку?
4. Що таке бітрейт? Які існують різновиду бітрейта?
5. Що таке аудіокодек, які його функції?
6. Які формати файлів використовуються для зберігання цифрового звуку?
7. Яка загальна структура формату RIFF WAVE?

Тема 7.

Кодування відеоданих

Основні поняття комп'ютерного відео

Нагадаємо, що відео як вид ММД має багато спільного з анімацією: в обох цих видах ММД має місце «зображення, що рухається», яке формується частою зміною окремих кадрів. Однак, на відміну від анімації, яка, у більшості випадків, має штучну природу (заснована на двовимірних або тривимірних моделях), класичне відео виходить шляхом зйомки на камеру реальних об'єктів навколишнього світу («природне» відео).

Відео, таким чином, має більш виражену, ніж анімація, потокову природу, що ріднить відео зі звуком. Ще одною важливою відмінною рисою відео є те, що в більшості випадків воно не є самостійним видом ММД. Як правило, відеопотік записується, зберігається і відтворюється паралельно з одним або декількома аудіопотоками, а також, можливо, текстовими доріжками субтитрів. Ця особливість є причиною того, що більшість відповідних форматів файлів — медіаконтейнерів (див. далі) — призначені для зберігання саме таких множинних потоків, включаючи відео.

Відео існує в різних областях застосування, у кожній з яких зложилися свої принципи, засоби і стандарти формування, зберігання, відтворення потоку даних. Традиційно розрізняють кінотеатральне, телевізійне, побутове та комп'ютерне відео.

В області комп'ютерного відео слід, насамперед, виділити два його різновиди, що відрізняються способом введення і виведення відеопотоку: аналогове і цифрове. Аналогове відео усе ще застосовується, в основному, у двох випадках: при оцифруванні за допомогою комп'ютера відео, що зберігається на аналоговому носії (найчастіше, магнітній стрічці відеокасети), а також при виведенні відеоконтенту на аналоговий телевізор. У першому випадку комп'ютер повинен бути оснащений платою нелінійного відеомонтажу або ТВ-тюнера, здатною виконувати загарбування аналогового відеосигналу в реальному часі. У другому випадку необхідна плата з аналоговим відеовиходом, композитним або компонентним (у період повсюдного поширення аналогових джерел і приймачів відеосигналу були поширені моделі відеокарт, оснащені такими виходами). Композитний відеовихід змішує сигнали кольоровості і яскравості в загальний аналоговий сигнал, а компонентний передає ці сигнали роздільно по самостійних аналогових лініях.

Поступово аналогове відео замінюється цифровим у всіх галузях розвитку і поширення відео. Цифрове відео утворюється з первинного сигналу, який одержується і оцифровується скануючим пристроєм — цифровою відеокамерою, після чого зберігається і виводиться на пристрій відеовиводу в цифровому вигляді. Оскільки відеодані, у порівнянні з іншими видами ММД, відрізняються найбільш значними об'ємами, для їхнього зберігання і пересилання по каналах зв'язку необхідно застосовувати ефективні алгоритми стиску. Якщо в області зберігання і відтворення звуку вже достатньо широко застосовуються алгоритми стиску без втрат, в області відео для зберігання і передачі кінцевого відеопродукту використовуються тільки алгоритми стиску із втратами, і ця ситуація, очевидно, зміниться ще не швидко (хоча алгоритми стиску відео постійно розвиваються в напрямку підвищення якості, тобто зменшення втрат при кодуванні).

Поняття розгорнення відеосигналу

Растрове розгорнення являє собою метод перетворення плоского рухомого зображення в сигнал, при якому відбувається його розкладання на окремі елементи послідовним рядковим зчитуванням. Така ж технологія використовується для перетворення сигналу назад у зображення. Головна область застосування растрового розгорнення — телебачення, тому воно часто називається телевізійним, хоча використовується також для відображення інформації комп'ютерів і радіолокаційних станцій. Телевізійне растрове розгорнення складається з рядкової і кадрової розгорнень і може бути як механічним, так і електронним.

У телевізійному віщанні прийняте прямокутне горизонтально-рядкове розгорнення (зліва направо і зверху донизу) і воно може бути двох видів: черезстрокове і прогресивне (рядкове).

При прогресивному розгорненні кадр формується скануванням елементів зображення в кожному рядку зліва направо і зчитуванням підряд кожного рядка зверху донизу. Після кожного рядка і кожного кадра передаються сигнали синхронізації.

При черезстроковому розгорненні кожен кадр розбивається на два напівкадри (або поля), складені з рядків, обраних через один. Черезстрокове розгорнення застосовується в тих або інших випадках для прискорення виведення зображень при обмеженій смузі пропускання (в аналоговій техніці) або ширині каналу (у цифровій техніці). У відеосигналі, при збереженні кількості рядків зображення, застосування черезстрокового розгорнення в 2

рази підвищує кадрову частоту в порівнянні із прогресивним. Це важливо для растрових пристроїв виведення на електронно-променевих трубах (таких як класичні телевізори), оскільки дозволяє суттєво знизити мерехтіння відеозображення.

Поняття медіаконтейнера

Медіаконтейнер (англ. *Media container*) — формат файлу або потоку, специфікації якого визначають спосіб збереження підготовлених ММД і не визначають спосіб їх кодування і представлення. Спосіб представлення самих ММД визначається форматом поточкових ММД, який, у свою чергу, може не визначати алгоритм кодування потоку.

Таким чином, медіаконтейнер визначає, скільки ММД фактично може бути збережене в його межах, разом з тим він не визначає жодної кодифікації самих ММД. Медіаконтейнер фактично є метаформатом, тому що він зберігає дані та інформацію про те, як ці дані будуть зберігатися безпосередньо у файлі або потоці. Внаслідок цього програма, яка здатна коректно ідентифікувати і відкрити файл медіаконтейнера, згодом може бути не здатна декодувати фактичні дані, записані усередині медіаконтейнера, оскільки або метаданих у медіаконтейнері недостатньо для цього, або програмне забезпечення нездатне декодувати дані, закодовані в медіаконтейнері [Wiki].

У теорії формат-контейнер здатний зберігати будь-який тип даних, однак на практиці для кожного типу даних існують окремі групи контейнерів. Ці групи «настроєні» для специфічних вимог і інформації, яка буде зберігатися в них. Більшість медіаконтейнерів призначені для збереження поточкових ММД типу фільмів, що включають у себе, у загальному випадку, відео, аудіо, зображення і текст. У випадку фільмів медіаконтейнер повинен не тільки належним чином зберігати відео- і аудіопотік, але і при відтворенні забезпечувати їхню синхронізацію. Також у медіаконтейнері може зберігатися кілька однотипних потоків, наприклад фільм (відео-потік) з декількома звуковими доріжками (аудіопотоками) і субтитрами (текстовими потоками).

Огляд розповсюджених медіаконтейнерів

AVI

Медіаконтейнер AVI (англ. *Audio Video Interleave* — «чергування аудіо і відео») був уперше використаний Microsoft в 1992 році в пакеті Video for Windows і призначений для зберігання мультимедійних відео- і аудіоданих. Основою структури контейнера AVI є формат — RIFF. Відео і

аудіо дані, що зберігаються у форматі AVI, можуть бути стиснуті з використанням різних комбінацій кодеків. Як і DVD (VOB), AVI файли підтримують многопотокowe аудіо в одному контейнері.

На початок 2012 року цей контейнер є найпоширенішим серед медіаконтейнерів, призначених для зберігання відео на комп'ютері (у файлових системах комп'ютерів). Однак, внаслідок історичних обмежень формату RIFF максимальний розмір файлу AVI не перевищує 2 Гбайт, що робить його досить незручним для роботи з відеоданими великого обсягу. У міру розвитку комп'ютерного відео і збільшення його обсягів зазначене обмеження стає згодом все більш істотним, що створює передумови для розвитку і поширення більш сучасних медіаконтейнерів.

VOB

VOB (*Dvd-video Object* або *Versioned Object Base*) — формат файлів, використовуваний для зберігання DVD-video. Даний медіаконтейнер заснований на стандарті MPEG-2 і здатний містити в собі кілька потоків відео і аудіо, субтитри, а також меню фільму. Усього можливо не більш 9 різних аудіопотоків і 32 потоки субтитрів.

Бітрейт стисненого відео варіюється від 2000 до 9800 Кбіт/с, часто є змінним (VBR). Розмір відеокадра для стандарту PAL рівний 720×576 точок, а для стандарту NTSC — 720×480 точок. Ця роздільність забезпечує так звану «стандартну чіткість» кадра (SD, *Standard-definition*). Більш високу чіткість стандарт DVD не підтримує, що приводить до поступового зниження його поширеності.

Аудіодані в контейнері VOB можуть бути збережені у форматі PCM, DTS, MPEG або Dolby Digital (AC-3). У країнах, що використовують стандарт NTSC, усі фільми на DVD повинні містити звукову доріжку у форматі PCM або AC-3, а всі NTSC-програвачі повинні ці формати підтримувати. Таким чином, будь-який стандартний диск може бути відтворений на будь-якому стандартному устаткуванні.

У країнах, що використовують стандарт PAL (більшість країн Європи), спочатку прагли ввести в якості стандарту звуку для DVD формати PCM і MPEG-2, але під суспільним тиском і в розріз із побажаннями Philips DVD-forum включив Dolby AC-3 у список опціональних форматів звуку на дисках і обов'язкових форматів у програвачах.

Matroska (MKV)

Matroska — це цілий проект, націлений на створення однойменного відкритого, гнучкого, кросплатформного (включаючи апаратні платформи) формату мультимедійного контейнера, а також набору інструментів і бібліотек для роботи з даними в цьому форматі.

Проект Matroska є прямою відкритою альтернативою контейнерам AVI, ASF, MOV, RealMedia, MP4, MPG. Він є розвитком проекту MCF, але значно відрізняється від останнього тим, що заснований на розширюваній двійковій метамові EBML (Extensible Binary Meta Language) — двійковому аналогу мови XML. Використання EBML дозволяє розширювати формат без втрати сумісності зі старими програмами.

Крім, власне, контейнера MKV для відео в проекті Matroska є спеціальні контейнери для аудіо — MKA, і для субтитрів — MKS.

Можливості формату, що закладаються в Matroska:

- підтримка відеопотоку високої (HD) і надвисокої (Full HD) чіткості;
- трансляція через Інтернет (протоколи HTTP і RTP);
- швидке перемотування по файлу;
- стійкість до помилок;
- екранні меню (як на DVD-дисках);
- розбивка файлу на глави (Chapters);
- субтитри, що перемикаються;
- звукові доріжки, що перемикаються;
- доріжки відео, що перемикаються;
- модульна розширюваність.

Matroska є відкритим проектом (open standard). Це означає, що для персонального використання вона абсолютно безкоштовна, а технічна специфікація формату бітового потоку доступна будь-кому, навіть компаніям, що бажають вмонтувати підтримку формату у свої продукти (для цього буде потрібно спеціальна ліцензія). Вихідний код усіх бібліотек, створених групою розроблювачів проекту Matroska, поширюється на умовах LGPL; бібліотека для програвання, написана на мові C з використанням цілочисельної арифметики, поширюється на умовах ліцензії BSD.

Підтримується адаптація і реалізація бібліотек Matroska для OpenBeOS Mediakit і Gstreamer (кросплатформного мультимедійного середовища, схожого

на DirectShow для Windows) і набору фільтрів DirectShow для відтворення і створення файлів Matroska в Windows.

Flash Video (FLV)

Медіаконтейнер Flash Video отримав основне поширення як засіб передачі відео через Інтернет. Використовується він такими сервісами відеохостингу як Youtube, Google Video, Вконтакте, Rutube та іншими. Хоча опис формату контейнера був відкритий, кодеки захищені патентами і залишаються власницькими.

FLV-файл містить бітовий потік, який є варіантом відеостандарту H.263. Flash Player 8 і більш нові редакції підтримують потокове відео On2 Truemotion VP6. On2 VP6 забезпечує більш якісне зображення, особливо при використанні низького бітрейту. З іншого боку, цей формат більш складний, що може створити труднощі при перегляді на застарілих машинах.

Опціональний альфа-канал, що представляє собою попиксельну прозорість, підтримується за допомогою додаткового відеопотоку, який кодує тільки альфа-канал. Реалізація припускає, що YUV-дані основного On2 VP6 відеопотоку завжди конвертуються клієнтом в RGB. Ця можливість доступна тільки для відео On2 VP6.

Починаючи з Flash Player 9 Update 3 підтримується новий формат мультимедіа файла ISO Base MPEG-4 Part 12, з новим відеокодеком — H.264. Цей стандарт відеостиску при тому ж низькому бітрейті видає значно більш деталізоване і чітке зображення, особливо в динамічних сценах. Недоліками є, знов-таки, підвищення вимог до обчислювальних ресурсів і платні патенти.

Звук в FLV, як правило, закодований в MP3, однак іноді можуть використовуватися Nellymoser codec, нестиснене аудіо або ADPCM аудіоформат. У версії Flash Player 9 Update 3, відповідно до впровадження Adobe формату ISO Base (MPEG-4 Part 12), додана підтримка AAC аудіо (профілі AAC-LC, Main Profile, і HE-AAC). У версії Flash Player 10 Beta доданий відкритий кодек SPEEX.

MPEG-4 Part 14 (MP4)

Формат медіаконтейнера MPEG-4 Part 14 (або MP4), офіційно відомий як ISO/IEC 14496-14:2003, є частиною стандарту MPEG-4. Він використовується для упакування цифрових відео- і аудіопотоків, субтитрів, статичних зображень (наприклад, постерів) і метаданих, які визначені групою розроблювачів. Як і більшість сучасних медіаконтейнерів, MPEG-4 Part 14 передбачає можливість здійснювати потокове віщання через Інтернет, додатково до файлу передаються метадані, що містять необхідну для віщання інформацію.

Контейнер MP4 підтримує такі формати даних:

Відеопотоки: MPEG-4 SP/ASP, MPEG-4 AVC/H.264, MPEG-2, MPEG-1 (за винятком DivX3/MS Mpeg-4, WM9, RV9 і VP6).

Аудіопотоки: AAC, MP3, MP2, MP1.

Субтитри: MPEG-4 Timed Text.

Статичні зображення: JPEG, PNG.

У цей час найбільше застосування одержало зв'язування з MPEG-4 AVC/H.264 і AAC, тому що більшість програвачів має апаратні можливості декодування цих потоків.

Контейнер MP4 активно використовує компанія Apple, застосовуючи при цьому власні, не передбачені стандартом, розширення файлу: m4a (аудіо-файл із потоком у форматі AAC або ALAC), m4v (файл із аудіо- і відеопотоками), m4b (файл AAC, що підтримує закладки), m4p (захищений файл AAC), m4r (файл рінгтону).

TS (MPEG-TS)

TS — формат медіаконтейнера, який інкапсулює пакети елементарних потоків транспортного потоку MPEG-TS і, можливо, інших даних. Транспортний потік MPEG-TS (TS, TP, M2TS) є протоколом для передачі аудіо і відео даних, описаним у стандарті MPEG2, Частина 1 [1]. Ціль розробки цього формату — мультиплексування аудіо і відео даних і синхронізація їх виведення. Потік MPEG-TS широко використовується в цифровому телебаченні, надаючи можливості для вдосконалення (у тому числі, виправлення помилок) інших телевізійних транспортних засобів, таких як DVB і ATSC [Wiki].

Поняття кодека

Кодек (англ. *codec*, від *coder/decoder* — шифратор/дешифратор — кодування/декодування або *compressor/decompressor*) — пристрій або програма, здатна виконувати перетворення даних або сигналу.

Кодеки можуть як кодувати потік (сигнал), як правило, для передачі, зберігання або шифрування, так і декодувати — для перегляду або внесення змін — у форматі, більш підходящому для цих операцій. Кодеки часто використовуються при цифровій обробці відео і звуку.

Більшість кодеків для звукових і візуальних даних використовують стиск із втратами, щоб отримати прийнятний розмір готового (стисненого) файлу. Існують також кодеки, що стискають без втрат (англ. *lossless codecs*). Але для

більшості застосувань вигідніше використовувати кодеки із втратами, тому що малопомітне погіршення якості виправдовується значним зменшенням об'єму даних. Майже єдине виключення — ситуація, коли дані зазнаватимуть подальшої обробки: у цьому випадку втрати, що накопичуються при багаторазовому кодуванні-декодуванні, можуть суттєво вплинути на якість.

Принципи стиску відео. Технологія компенсації руху

Для кодека відеопотік, по суті, являє собою тривимірний масив кольорових пікселів. Два виміри відповідають вертикальній і горизонтальній координатам кадра, а третій вимір — це час. Кадр — це масив усіх пікселів, видимих камерою в цей момент часу, або просте зображення.

Одна з найбільш потужних технологій, що дозволяють підвищити ступінь стиску відео, є технологія компенсації руху. У будь-якій сучасній системі стиску відео наступні кадри в потоці використовують подібність областей у попередніх кадрах для збільшення ступеня стиску. Технологія компенсації руху дозволяє знаходити схожі ділянки, навіть якщо вони зрушені щодо попереднього кадру.

Зараз практично всі алгоритми стиску відео (наприклад, стандарти, прийняті ITU-T або ISO) використовують дискретне косинусне перетворення (DCT) або його модифікації для усунення просторової надмірності. Використання більшості методів стиску (таких, як дискретне косинусне перетворення) тягне також використання процесу квантування. Квантування може бути як скалярним, так і векторним, проте, більшість схем стиску на практиці використовують скалярне квантування внаслідок його простоти.

Сучасне цифрове телемовлення стало доступним саме завдяки відео-компресії. Телевізійні станції можуть транслювати не тільки відео високої чіткості (HDTV), але і кілька телеканалів в одному фізичному телеканалі (частотному діапазоні).

Хоча більша частина цифрового відеоконтенту сьогодні транслюється з використанням стандарту стиску відео MPEG-2, проте в телемовленні вже використовуються нові і більш ефективні стандарти стиску відео, наприклад MPEG-4 H.264 і VC-1.

Для кодування і зберігання інформації про черговий кадр можуть використовуватися різні способи, що відрізняються один від одного наявністю або відсутністю залежностей цього кадру від попередніх і наступних.

Зазвичай кадр розбивається на квадратні макроблоки, і тип посилання для кожного з макроблоків визначається індивідуально, однак з обмеженням, заданим типом усього кадру:

- I-кадри (також називаються ключовими (*keyframes*) або «опорними») можуть містити тільки незалежно стиснуті макроблоки;
- P-кадри («різницеві» кадри) можуть містити як незалежно стиснуті макроблоки, так і макроблоки з посиланням на інший кадр;
- B-кадри («двоспрямовані», «зворотні» кадри) можуть містити такі макроблоки: незалежні (*intra*), з посиланням на один кадр (*predicted*) або з посиланням на 2 кадри (*bi-predicted*).

У новому стандарті MPEG-4 AVC/H.264 також вводиться поняття SI- і SP-кадрів.

Контрольні питання

1. У чому проявляються подібність і відмінність між комп'ютерним відео і анімацією?
2. У чому полягає сутність поняття розгорнення відеосигналу?
3. Що таке медіаконтейнер? Яку роль він відіграє у представленні відеоданих?
4. Які медіаконтейнери дістали найбільш широкого поширення?
5. Що таке мультимедійний кодек?
6. Які принципи стиску відео?
7. У чому полягає суть технології компенсації руху?

Тема 8.

Стандарт опису мультимедійних даних MPEG-7

Призначення стандарту MPEG-7

MPEG-7 — стандарт ISO/IEC, розроблений Moving Picture Experts Group (MPEG) — автором усіх відомих MPEG-стандартів (MPEG-1, MPEG-2, MPEG-4). MPEG-7 виступає в ролі стандарту, що поєднує стандарти кодування MPEG (1, 2 і 4) і описує контент, отриманий з їхньою допомогою. Цим пояснюється номер 7 у назві стандарту — це сума номерів стандартів 1, 2 і 4.

MPEG-7 стандартизує опис різних типів мультимедіа-інформації. Такі описи асоційовані з мультимедіа-контентом, тим самим дозволяючи робити зручний асоціативний пошук. Формально MPEG-7 називається Інтерфейсом Опису Мультимедіа-Контенту (Multimedia Content Description Interface).

Крім стандартизації описів визначених типів контенту, MPEG-7 стандартизує механізм створення подібних описів. Споконвічно передбачається, що мультимедіа-контент може містити такі типи інформації: зображення (растрові статичні зображення, векторну графіку), 3D-моделі, аудіо, мовлення, відео, а також інформацію про комбінацію елементів цих типів у єдиному мультимедіа-поточі або презентації (у вигляді своєрідних сценаріїв).

Головною областю застосування і позиціонування MPEG-7 є мережа Інтернет як основне сховище розрізненої інформації з досить утрудненим пошуком і навігацією в ній.

Останнім часом виникла тенденція до об'єднання інформації, представленої в різних видах, у єдиний медіа-контент, більш зручний для сприйняття кінцевим споживачем, ніж той, що поставляється в розрізненому виді (наприклад, текст — окремо, музика — окремо). Крім того, у зв'язку з лавиноподібним ростом обсягу інформації, орієнтація в ньому вимагає все більших зусиль.

Особливо проблематичним представляється пошук у мультимедійному контенті різних типів. Універсальні пошукові машини досить успішно справляються з пошуком у текстових масивах, але здійснювати пошук у графічному, аудіо- і відео-контенті з їхньою допомогою досить важко — у загальному випадку пошукові системи обмежуються пошуком у текстових описах до мультимедійних файлів.

На сайті MPEG представлений такий приклад пошукових запитів:

1. Музика. Користувач грає кілька нот на мультимедійній клавіатурі і одержує список музичних фрагментів, що містять ці ноти (або дуже близькі за звучанням), або зображення, що якимось чином сполучаються із цими нотами (наприклад, в емоційному плані).

2. Графіка. Користувач малює кілька ліній на екрані і одержує набір картинок, що містять схожі лінії.

3. Зображення. Користувач визначає набір об'єктів, що включають світлові схеми або текстури, і одержує деякі приклади або шаблони, які він може використати для побудови свого зображення.

4. Анімація. На певній множині об'єктів користувач задає їхній рух та відносини між ними і одержує набір анімаційних роликів, у яких присутні задані часові і просторові відносини.

5. Сценарії. На деякому контенті користувач задає деякі дії і одержує список сценаріїв, у яких є схожі дії.

6. Голосова інформація. Маючи звукові уривки з голосом певного виконавця (співака), користувач одержує список записів, відеокліпів, у яких співає цей співак, або відеокліпів, у яких він присутній.

MPEG-7 не стандартизує самі методи аналізу мультимедіа-контенту (наприклад, розпізнавання мови), а лише специфікує принципи формування опису такого контенту. Створення самих засобів розпізнавання приділяється фірмам, які будуть створювати продукти, що базуються на технології MPEG-7.

Структурні складові MPEG-7

MPEG-7 складається з наступних частин:

- MPEG-7 Systems — бінарний формат кодування описів (дескрипторів) в MPEG-7;
- MPEG-7 Description Definition Language (DDL) — мова опису дескрипторів, дескрипторних засобів, а також нових дескрипторних схем;
- MPEG-7 Visual — дескрипторна схема, що відповідає винятково за візуальні дескриптори;
- MPEG-7 Audio — дескрипторна схема, що відповідає винятково за аудіо-дескриптори;
- MPEG-7 Multimedia Description Schemes (MDS) — схема опису мультимедіа, що працює із загальними властивостями і мультимедіа-дескрипторами;
- MPEG-7 Reference Software — програмна реалізація істотних частин MPEG-7;

- MPEG-7 Conformance Testing — правила і процедури тестування відповідності стандарту MPEG-7;
- MPEG-7 Extraction — технічні описи засобів роботи з дескрипторами.

Мова опису дескрипторів (DDL) базується на мові XML і, зокрема, XML Schema, створеного для опису структурних елементів. Однак, оскільки XML Schema споконвічно не передбачена для опису аудіо-візуальної інформації, MPEG-7 дещо його розширює.

Схема опису мультимедіа (MDS) містить у собі набір дескрипторних засобів, що працюють із мультимедіа-елементами в цілому, які можуть бути розділені на п'ять складових:

- опис контенту: представлення сприйманої інформації;
- керування контентом: інформація про властивості, створення і використання аудіо-візуального контенту;
- організація контенту: представлення аналізу і класифікації різних аудіо-візуальних складових контенту;
- навігація і доступ: узагальнена специфікація складових аудіовізуального контенту;
- взаємодія з користувачем: опис користувацьких налаштувань і історії використання мультимедіа-матеріалу.

Контрольні питання

1. Яке призначення стандарту MPEG-7?
2. Чому стандарт MPEG-7 одержав номер 7 у своєму позначенні?
3. Які структурні складові стандарту MPEG-7?
4. Які характеристики можуть виступати елементами опису мультимедійних даних різних типів?

Список прийнятих скорочень

АЦП — аналого-цифровий перетворювач

ІКМ — імпульсно-кодова модуляція

ММ — мультимедіа, мультимедійний

ММС — мультимедійна система

ОС — операційна система

ПЗ — програмне забезпечення

ПК — персональний комп'ютер

Література

Основна література

1. Жук Ю. А. Мультимедийные технологии [Электронный ресурс] : учеб. пособие : самост. учеб. електрон. изд. / Ю. А. Жук ; Сыкт. лесн. ин-т. — Электрон. дан. — Сыктывкар : СЛИ, 2012. — Режим доступа: <http://lib.sfi.komi.com>. — Загл. с экрана.
2. Засоби мультимедіа в нових інформаційних технологіях : Методичні рекомендації / уклад. Іващук В. В. — К. : НУХТ, 2012. — 13 с.
3. Крапивенко А. В. Технологии мультимедиа и восприятие ощущений / А. В. Крапивенко.— М.: Бином. Лаборатория знаний, 2009. — 271 с.

Додаткова література

4. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. — М.: Диалог-МИФИ, 2003. — 384 с.
5. Есенин С. А. DirectX и Delphi. Разработка графических и мультимедийных приложений. — СПб.: БХВ-Петербург, 2006. — 512 с.
6. Залогова Л. А. Компьютерная графика. Практикум. 2-е изд. — М.: Лаборатория Базовых Знаний, 2005. — 320 с.
7. Краснов М. В. DirectX. Графика в проектах Delphi. — СПб.: БХВ-Петербург, 2003. — 416 с.
8. Миано Дж. Сжатие изображений в действии. Форматы и алгоритмы. — М.: Триумф, 2003. — 336 с.
9. Осипов Д. Графика в проектах Delphi. — СПб: Символ Плюс, 2008. — 648 с.
10. Розробка мультимедіа-систем. Завдання до контрольних робіт і методичні вказівки до їх виконання. Для студентів IV курсу спеціальності 091501 «Комп'ютерні системи і мережі» заочної форми навчання / Уклад. О. Л. Нєнов. — Одеса: ОДАХ, 2011. — 40 с.
11. Залогова Л. А. Компьютерная графика. Элективный курс: Практикум / Л. А. Залогова — М. : БИНОМ. Лаборатория Базовых Знаний, 2005. — 245 с., 16 с. ил. — ISBN 5-94774-152-0.
12. Казанцев О. В. Методы и средства Мультимедиа. Звук [Электронный ресурс] / О. В. Казанцев, И. К. Никитин. — М. : МАИ, 2011. — 89 с. — Режим доступа: <http://www.twirpx.com/file/405661/>. — Загл. с экрана.
13. Нєнов О. Л. Розробка мультимедійних систем. Навчальний посібник / О. Л. Нєнов. — Одеса: Одеська державна академія холоду, 2012. — 76 с.

Артеменко С. В.

ОСНОВИ МУЛЬТИМЕДІА

Навчальний посібник

Підписано до друку __.__.2015 р. Формат 60×84 1/16.

Умовн. друк. арк. **3,94**. Наклад __ прим.

Надруковано видавничий центр ОНАХТ «Технолог».

65039, Одеса, вул. Канатна, 112