# РОЗДІЛ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

УДК 004.934 DOI 10.31319/2519-2884.32.2018.174

ЯЛОВА К.М., к.т.н., доцент
ЯШИНА К.В., к.т.н., доцент
ВАСИЛЬЄВА А.В., магістр

Дніпровський державний технічний університет, м. Кам'янське

## ДОСЛІДЖЕННЯ СИСТЕМ ТА МЕТОДІВ РОЗПІЗНАВАННЯ МОВНОГО СИГНАЛУ

**Вступ.** Мова – основний вид передачі думок, ідей та почуттів у людському середовищі. Можливість контролю оточення голосом підштовхнув науковців та експертів передових корпорацій до створення таких програмних додатків, як Cortana від Microsoft, Siri від Apple та інших. Вирішення задачі розпізнавання голосу дозволив перейти від командного та WIMP (Window, Image, Menu, Pointer) до SILK (Speech, Image, Language, Knowledge) інтерфейсу програмних систем. На відміну від взаємодії користувача з комп'ютерною системою за допомогою клавіатури, миші, джойстика та дисплею мовний інтерфейс має наступні переваги:

- для спілкування з комп'ютером людині немає необхідності мати спеціальні навички або уміння в галузі інформаційних технологій;
- мова знижує психологічну та фізичну відстань між людиною і комп'ютером та може бути пов'язана з ним через системи комунікацій, наприклад, телефон;
- мовний інтерфейс надає оперативність і мобільність спілкування, звільнення рук і розвантаження зорового каналу при отриманні інформації [1].

**Постановка задачі.** Метою даної роботи є дослідження та проведення аналітичного порівняння систем автоматичного розпізнавання мови (САРМ), методів та алгоритмів, що використовуються при розпізнаванні мовного сигналу.

**Результати роботи.** Під розпізнанням мови розуміють процес трансформації мовного сигналу в цифрову інформацію (наприклад, текстові дані) [2]. Для цього існує безліч САРМ – систем, що перетворюють вхідний мовний сигнал в розпізнане повідомлення [3]. При цьому повідомлення може бути представлено як у формі тексту цього повідомлення, так і одразу перетворено в зручну для його подальшої обробки форму з метою формування відповідної реакції програмної системи. САРМ класифікуються за такими ознаками, як:

1) розмір словника (обмежений набір слів або великий словник) – чим більший розмір словника, з яким працює система розпізнавання мови, тим більше помилок при розпізнаванні слів. Словник, що складається тільки з цифр, може бути розпізнаний практично безпомилково, тоді як вірогідність помилок при розпізнаванні словника в сто тисяч слів може досягати 45%. Потрібно також враховувати унікальність слів в словнику. Якщо слова дуже схожі, то похибка розпізнавання збільшується;

2) залежність від диктора (дикторозалежні або дикторонезалежні) – дикторозалежна система призначена для роботи тільки з людиною, яка навчала цю систему, в той час як дикторонезалежна система призначена для роботи з будь-яким диктором. На поточному етапі розвитку САРМ вірогідність виникнення помилок в дикторонезалежній системі в 3-5 разів більша, ніж у дикторозалежних;

3) тип мови (злитна, роздільна). Роздільна мова – це мова, в якій слова відокремлюються одне від одного проміжком тиші. Злитна мова – це природно вимовлений

текст. Розпізнавання злитного мовлення складніше, тому що у вимовлених слів немає чітких меж. САРМ, що працюють з ізольованими словами, досягли високо рівня точності розпізнавання – 95-99%, в той час як задача розпізнавання злитної мови в достатній мірі не вирішена [4]. У системах для розпізнавання злитного мовлення від IBM та Microsoft частота помилок становить 5.5-5.9%;

4) призначення (системи диктування, командні системи) – визначає необхідний рівень абстракції, на якому буде відбуватися розпізнавання мови. Системи голосового набору мобільного телефону, де здійснюється розпізнавання за шаблоном, називаються командними. На відміну від них, система диктування вимагає розпізнавання на базі виділення лексичних елементів. При інтерпретації виголошеної фрази вона буде покладатися не тільки на те, що було виголошено в поточний момент, але і на те, як це співвідноситься з тим, що було вимовлено до цього. Також в таку систему повинен бути вбудований набір граматичних правил. Чим суворіші ці правила, тим простіше реалізовувати систему розпізнавання, але набір слів, які вона зможе розпізнати, буде меншим;

5) алгоритм, що використовується. Після того, як мовний сигнал розбивається на певні частини, відбувається імовірнісна оцінка належності цих частин до того чи іншого елемента словника, що здійснюється за допомогою одного з алгоритмів розпізнавання;

6) по типу структурної одиниці (фрази, слова, фонеми, діфони, алофони) – САРМ, які використовують цілі слова або фрази, називаються САРМ за шаблоном. Вони як правило дикторозалежні, і їх реалізація є простішою, ніж створення САРМ, які розпізнають мовлення на базі виділення лексичних елементів. У таких системах структурними одиницями мови є лексичні елементи;

7) по принципу виділення структурних одиниць. Найпоширеніший підхід виділення структурних одиниць заснований на перетворенні Фур'є, яке переводить вихідний сигнал з амплітудно-часового простору в частотний. Однак аналіз Фур'є має цілу низку недоліків, в результаті яких відбувається втрата інформації стосовно часових характеристик оброблюваних сигналів. У зв'язку з цим для завдання виділення структурних одиниць мови виправдано використання вейвлет-аналізу. Вейвлет – це математична функція, яка дозволяє аналізувати частотні компоненти даних. В загальному випадку, аналіз сигналів проводиться в площині вейвлет-коефіцієнтів – масштаб-час-рівень (Scale-Time-Amplitude). Отримані вейвлет-спектри відрізняються від спектрів Фур'є тим, що дають чітку прив'язку властивостей сигналу до часу. Крім вейвлет і Фур'є-аналізу в САРМ використовується кепстральний аналіз, але створення таких систем є трудомістким і вимагає високої кваліфікації розробника.


Рисунок 1 – Структурна схема автоматичного розпізнавання мови

Спрощену структурну схему роботи САРМ наведено на рис.1.

Метою аналізу мовного сигналу є виділення в складі отриманого сигналу компонентів, які є основними для розпізнавання отриманого повідомлення. До таких компонентів належать параметри, що описують мову, аналогічні тим, які формуються в процесі синтезу мови. Набір зазначених параметрів залежить від обраного методу розпізнавання.

Модель розпізнавання мови і прийняття рішення – це блок, в рамках якого здійснюється формування розпізнаного повідомлення на основі аналізу послідовності параметрів, отриманих після аналізу мовного сигналу. Наприклад, якщо вико-

ристовується формантна модель опису мови, то на основі отриманих в першому блоці частот формант будується послідовність розпізнаних фонем, що складають вхідне повідомлення. При цьому здійснюється прийняття рішення про те, чи розпізнано вхідне повідомлення правильно. При прийнятті рішення можливі наступні варіанти: повідомлення розпізнано правильно (підтвердженням цього є текст, що відповідає нормам природної мови) або повідомлення не розпізнається, або розпізнано неправильно (таке рішення приймається в разі наявності в розпізнаному повідомленні явних помилок, які важко виправити автоматично, або взагалі повна нісенітниця).

У процесі розпізнавання мови найскладніше полягає в здійсненні процедури порівняння вхідного та еталонного елемента, задача ускладнюється ще й тим, що вхідні сигнали характеризуються протяжністю в часі. На теперішній час існують багато методів та алгоритмів розпізнавання мови, найпоширенішими з яких є: лінійні моделі, нейронно-мережеві методи, приховані марковські моделі, метод динамічного трансформування часу тощо.

***Лінійні моделі.*** Першими з'явилися лінійні моделі розпізнавання мови. В них передбачається, що для порівняння мовного сигналу з еталоном досить простого масштабування в часі. Метод лінійної екстраполяції передбачає представлення мовного сигналу $s(n)$ в якості комбінації попередніх відліків сигналу, а математично модель сигналу представляється у вигляді:

$$s(n) = -\sum_{i=1}^{N_{LP}} \alpha_{LP}(i) s(n-i) + e(n),$$

де $N_{LP}$ – кількість коефіцієнтів моделі або порядок передбачення; $\alpha_{LP}$ – коефіцієнти лінійного передбачення; $e(n)$ – функція помилки моделі (різниця між передбаченим значенням і реально зміненим значенням). Помилка передбачення визначається у вигляді різниці між вихідним та передбаченим відліками сигналу. Основна задача методу лінійного передбачення зводиться до визначення набору коефіцієнтів передбачення, які б забезпечили мінімізацію помилки передбачення.

Оскільки в мові мають місце нелінійні спотворення часу, тобто лінійна модель передбачала порівняння реалізації з еталоном за лінійним законом, тоді як зміни в реалізації піддаються нелінійних спотворень.

***Нейронні мережі.*** Одним з найбільш ефективних методів розпізнавання мови є метод з використанням нейронних мереж. Нейрон представляє собою комірку мережі і може знаходитись у збудженому або загальмованому стані та має зв'язки з іншими нейронами мережі – синапси (однонаправлені вхідні зв'язки), аксони – вихідні зв'язки нейрона, по яким сигнали (збудження або гальмування) надходять до синапсів наступних нейронів.

Кожний односпрямований зв'язок характеризується вагою $w_i$. Позитивні та негативні значення $w_i$ відповідають збудженому або загальмованому стану синапсів. Сума всіх входів визначає поточний стан нейрона, що визначається за формулою:

$$S = \sum_{i=1}^{n} x_i w_i .$$

Результат підсумовування передається до функції активації, яка може бути представлена нарізно, але найпоширенішою є логістична функція, яку можна розрахувати за формулою:

$$F(S) = \frac{1}{1 + e^{-\alpha S}} .$$

При використанні нейронних мереж для розпізнавання мовних сигналів необхідно побудувати відповідну призначену для цієї задачі мережу та підібрати вагові коефіцієнти синапсів для мінімізації помилок.

***Динамічне трансформування часу (ДТЧ).*** (ДТЧ – Dynamic Time Warp). Вимова одного й того ж слова, зазвичай, має різну тривалість. Навіть якщо слово було вимовлено з однаковою тривалістю, тривалість вимови окремих частин слова може бути різною. Тому, щоб отримати оцінку розходження між двома мовними сигналами у вигляді вектора, має бути виконано вирівнювання за часом, яке реалізується за допомогою ДТЧ. ДТЧ – метод еластичного порівняння вектора спостереження зі збереженим шаблоном. Вектор спостережень та шаблон лежать на відповідних осях сітки. Для кожної комірки сітки розраховується різниця між відповідними фрагментами вектора спостережень та шаблона.

Метод ДТЧ працює з фрагментами мовного сигналу, тобто аналіз ознак складається з обробки вектора ознак в регулярних інтервалах. Вектор ознак може мати велику кількість фрагментів, тому є потреба у засобах розрахунку локальної оцінки відстані між точками сигналу $x$ та шаблону $y$ у n-вимірному просторі за формулою:

$$(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2},$$

де $x_i$, $y_i$ – елементи, що порівнюються, $i$ – номер фрагмента.

Таким чином, вхідний сигнал порівнюється зі всіма шаблонами. Результатом порівняння буде шаблон, для якого знайдено мінімальне розходження між вхідним сигналом і шаблоном, що є сумою локальних відстаней між фрагментами сигналу і шаблону.

Оскільки для визначення основи послідовності в динамічному програмуванні оптимальним є використання методу зворотного програмування, необхідно використовувати певний динамічний стек. Подібно будь-якому динамічному алгоритму програмування ДТЧ має поліноміальну складність. У випадку великих послідовностей виникають незручності із збереження великих числових матриць та виконання великої кількості розрахунків відхилень. Існує поліпшена версія алгоритму, FastDWT, яка вирішує дві вищевказані проблеми. Рішення полягає в розбитті матриці станів на 2, 4, 8, 16 і т.д. менших за розміром матриць за допомогою повторюваного процесу розбиття послідовності введення на дві частини. Таким чином, розрахунки відхилення здійснюються тільки на цих невеликих матрицях і шляхах деформації, розрахованих для невеликих матриць.

У роботі [5] авторів Запрягаєва С.А. та Коновалова А.Ю. представлено результати розробки програмного додатку для розпізнавання командних слів. Якість розпізнавання методом ДТЧ в даній роботі становить 94%. В роботі [6] авторів Є.С.Малькової і О.А.Шабаліної описано методи розпізнавання мови в задачі автоматизованого виявлення дефектів вимови. Автори статті змогли створити простий класифікатор із 80% коректністю класифікації при застосуванні методу ДТЧ.

***Приховані марковські моделі.*** В основі прихованої марковської моделі (ПММ) лежить кінцевий автомат, що складається з $N$ прихованих станів. Перехід між станами в кожний дискретний момент часу $t$ не є детермінованим, а відбувається відповідно до ймовірнісного закону і описується матрицею переходів. Знаходження моделі в стані $i$ відповідає певній стаціонарності сигналу на обмеженому інтервалі часу. При здійсненні чергового переходу в новий стан $i$ в момент часу $t$ здійснюється генерація вихідного параметричного вектора $x_t$ у відповідності до багатомірної функції розподілу ймовірностей $f_j(x)$. Результатом роботи ПММ є послідовність векторів спостереження довжиною $T$. Перевагою ПММ є можливість обробки послідовностей і сигналів різної

довжини, що ускладнено при роботі зі штучними нейронними мережами. Функція щільності ймовірностей $f_j(x)$ для стану $j$ описується:

$$f_j(x) = \sum_{i=1}^{M} w_i p_i(x),$$

де $M$ – кількість компонентів; $w_i$ – вага компонента; $p_i(x)$ – нормальний розподіл для D-вимірного випадку.

Функція $p_i(x)$ описується наступним виразом:

$$p_i(x) = \frac{1}{2\pi^{\frac{D}{2}} |\sigma_i|^{\frac{1}{2}}} exp\left\{ -\frac{1}{2(x-\mu_i)^{\tau} \sigma_i^{-1} (x-\mu_i)} \right\},$$

де $D$ – розмірність вектору; $\mu_i$ – вектор математичного очікування; $\sigma_i$ – матриця коваріації.

Робота з ПММ проводиться в 2 етапи:

1) визначення параметрів моделі (алгоритм Баума-Велча);
2) визначення ймовірності того, що послідовність векторів, за якими ведеться спостереження, була згенерована даною моделлю (алгоритм Вітербі).

Для якісного навчання ПММ необхідна велика кількість зразків сигналу. Також необхідно дотримуватись умов лінійної незалежності навчальних зразків, у разі невиконання чого має місце виродження матриці коваріації, що може призвести до повної непрацездатності моделі. Однією з найуспішніших САРМ, що використовує ПММ є дикторонезалежна система розпізнавання неперервної мови Sphinx.

**Висновки.** Швидкий розвиток інформаційних технологій та комп'ютерної техніки ставить перед науковцями та винахідниками задачу оптимізації машинно-людинного інтерфейсу та зменшення проблем комунікації між людиною та комп'ютером. Створення мовних інтерфейсів може знайти застосування в системах різного призначення: голосове управління для людей з обмеженими можливостями, автовідповідачі, оброблення в автоматичному режимі сотні тисяч дзвінків на добу (наприклад, в системі продажу авіаквитків) тощо.

Одним із можливих способів скорочення фізичної та психологічної відстані між користувачем та комп'ютерною технікою є застосування мовного інтерфейсу для керування комп'ютерними пристроями та системами на відстані, базуючись на мовні сигнали користувача. В даній роботі подано класифікацію САРМ за різними властивостями та признаками, перелічені представники САРМ та описані якісні показники розпізнавання мовного сигналу. На основі аналітичного огляду представлено основні методи обробки мовних сигналів, які використовуються в САРМ, а саме: лінійне передбачення мовного сигналу, нейронні мережі, приховані марковські моделі та метод динамічного трансформування часу. Поданий опис дозволяє оцінити можливості існуючих методів обробки мовних сигналів і визначити перспективність застосування їх математичних апаратів у задачах обробки мовних сигналів у САРМ.

ЛІТЕРАТУРА

1. Алимурадов А.К. Обзор и классификация методов обработки речевых сигналов в системах распознавания речи / А.К.Алимурадов, П.П.Чураков // Измерение. Мониторинг. Управление. Контроль. – 2015. – №2(12). – С.27-35.
2. Федосин С.А. Классификация систем распознавания речи [Електронний ресурс] / Федосин С.А., Еремин А.Ю. – Режим доступу: http://fetmag.mrsu.ru/2010-2/pdf/SpeechRecognition.pdf.

3. Аграновский А.В. Теоретические аспекты алгоритмов и классификации речевых сигналов / А.В.Аграновский, Д.А.Леднов. – М.: Радио и связь, 2004. – 164с.
4. Титов Ю.Н. Современные технологии распознавания речи / Ю.Н.Титов // Вестник ТГУ. – 2006. – Т.11. – Вип.4. –С.571-574.
5. Запрягаев С.А. Распознавание речевых сигналов / С.А.Запрягаев, А.Ю.Коновалов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2009. – №2. – С.37-46.
6. Малькова Е.С. Методы распознавания речи в задаче автоматизированного выявления дефектов произношения / Е.С.Малькова, О.А.Шабалина // Известия Волгоградского государственного технического университета. – 2015. – №2. – С.65-71.

*Надійшла до редколегії 29.01.2018.*

УДК 004.42 DOI 10.31319/2519-2884.32.2018.175

ДЕМЧЕНКО Ю.Ю., студент
БАБЕНКО М.В., к.т.н., доцент

Дніпровський державний технічний університет, м. Кам'янське

## ВИКОРИСТАННЯ КОЛІРНОЇ МОДЕЛІ RGB ТА МЕТОДУ LSB ПРИ СТЕГАНОГРАФІЧНОМУ ЗАХИСТІ ІНФОРМАЦІЇ У ФАЙЛАХ ФОРМАТУ OFFICE OPEN XML

**Вступ.** Розглядаючи способи захисту інформації, доречно звернути увагу на стеганографічні методи. Сам термін «стеганографія» означає приховане повідомлення, яке повністю виключає можливість дізнатися про його існування третій особі. В якості сучасного прикладу можна привести випадок роздрукування контрактів з малопомітними викривленнями обрисів певних символів тексту на ЕОМ. Таким чином вносились дані про умови складання контракту, які необхідно було зашифрувати.

Комп'ютерна стеганографія ґрунтується на двох основних принципах [1]. По-перше, файли з оцифрованими зображеннями, а також аудіо- та відеофайли можна певною мірою змінити без втрати їх функціональності. По-друге, можливості людини розрізняти незначні зміни звуку або кольору досить обмежені. Стеганографічні методи дають можливість замінити несуттєві частки даних потрібною інформацією. Це означає, що сімейне фото може містити інформацію комерційного характеру, а файл з улюбленою мелодією – секретне повідомлення.

Проте найчастіше стеганографія застосовується для створення цифрових водяних знаків, які на відміну від звичайних можна виявити, лише використовуючи необхідне програмне забезпечення. Цифрові водяні знаки записуються у вигляді псевдовипадкових послідовностей сигналів шуму, які згенеровані на базі секретних ключів. Такі знаки забезпечують автентичність або недоторканість документа, дають можливість ідентифікувати власника або автора, перевірити права користувача або дистриб'ютора навіть в тому випадку, коли файл був спотворений або оброблений.

Щодо впровадження засобів програмно-технічного захисту в ІС, виділяють два головних способи:

1 – вбудований захист – механізми захисту розподілені за іншими компонентами системи або реалізуються у вигляді окремих складових ІС;

2 – додатковий захист – засоби захисту являють собою доповнення до основних

апаратних і програмних засобів комп'ютерної системи.

Другий спосіб є більш гнучким, його механізми при необхідності можна додавати та вилучати, але під час його реалізації можуть з'явитися проблеми забезпечення сумісності методів захисту між собою та з програмно-технічним комплексом інформаційної системи. Вбудований захист вважається більш надійним та оптимальним, але в той же час є жорстким, адже в нього складно внести зміни. Таким доповненням характеристик методів захисту зумовлюється те, що в реальній системі їх комбінують.

Існуючі алгоритми вбудовування секретної інформації поділяють на декілька груп:

1 – ті, які працюють з самим цифровим сигналом. До цієї групи відноситься метод LSB;

2 – «впаювання» таємної інформації. У цьому випадку відбувається накладення зображення, звуку або тексту, які необхідно приховати, поверх оригіналу. Досить часто застосовується для вбудовування ЦВЗ (цифровий водяний знак);

3 – використання можливостей файлових форматів. Сюди відноситься вкладення інформації в метадані або в інші зарезервовані поля файлу, які не використовуються.

За методом вбудовування інформації стеганографічні алгоритми поділяють на лінійні, нелінійні та інші.

Оскільки ми будемо вбудовувати приховані дані в текстовий файл, розглянемо методи, за допомогою яких це можна реалізувати [2].

1. Зміна регістру літер. Наприклад, нам треба сховати букву «А» в тексті «машина». Для цього ми беремо двійкове представлення коду символу «А» – «01010». Нехай для позначення біта, який містить одиницю, використовується символ верхнього регістру, а для нуля – нижнього. Результатом такого приховання буде «мАшИна». Закінчення «а» не використовується, так як для приховання цього символу потрібно було лише 5 біт, в той час як довжина рядка складає 6 символів. Таким чином вийшло, що остання літера – «зайва». Використовуючи такий підхід, можна сховати в текст довжиною N повідомлення з N/5 символів, що досить незручно.

2. Зміна кількості проміжків. Будемо вважати, що один проміжок відповідає біту «0», а два – «1». Програма отримує будь-який текст в якості контейнера і вкладає в нього повідомлення, замінюючи його біти на відповідну кількість проміжків. Важливу роль тут також відіграє і спосіб кодування символів. Треба отримати код символів оптимальної довжини, і щоб при цьому подвійний проміжок зустрічався якомога менше разів.

3. Line-shiftcoding. Змінюється відстань між рядками електронного тексту.

4. Word-shiftcoding. Змінюється відстань між словами тексту. Суть методу полягає в тому, що береться текст з різними відстанями між словами. Виділяється максимальна та мінімальна відстані, які позначаються відповідно 1 та 0, а інші відстані збільшують або зменшують до розмірів виділених. Окремим випадком цього методу являється метод зміни кількості проміжків, розглянутий вище.

5. Featurecoding. Внесення специфічних змін у шрифти окремих літер, наприклад, варіації довжини нижньої частини літери р.

Розглянуті вище методи досить легко вбудовуються в будь-який текст незалежно від його змісту, призначення та мови. Але, на жаль, такі методи легко зламуються, і секретна інформація може стати доступною третій особі. Також великим недоліком є те, що цими методами не можна передавати велику кількість прихованої інформації. Тому ми будемо використовувати інший метод, який має назву LSB.

LSB (Least Significant Bit, найменший значущий біт). Суть полягає в заміні найменш значущих бітів контейнера (аудіо-, відеофайл, зображення або текстовий файл) на біти повідомлення, яке необхідно приховати [3]. Оскільки можливості людського

ока розрізняти відтінки одного й того самого кольору досить обмежені, така заміна буде непомітною для людини. Саме на базі методу LSB і буде реалізовано алгоритм приховання таємної інформації, якому присвячена дана робота.

**Постановка задачі.** Засобами мови програмування C# розробити програмне забезпечення, за допомогою якого можна буде приховати таємну інформацію таким чином, щоб про її існування не дізнався будь-хто інший. Також необхідно забезпечити можливість витягнення секретного повідомлення з контейнера, в якому воно вже приховане. В якості контейнера (або сховища) для таємних даних ми будемо використовувати файл формату Office Open XML на прикладі документа Microsoft Office Word з розширенням docx. Чому саме docx, а не, наприклад, doc або txt? На це є декілька важливих причин. По-перше, файл з розширенням docx, на відміну від doc, являє собою zip-архів з XML-документами, який можна розпакувати та отримати всю необхідну інформацію: текст, зображення, таблиці тощо. Завдяки цьому досить легко вкладати та діставати приховані в нього дані. По-друге, docx – найбільш популярний і масовий формат, і його часте використання не буде викликати ні в кого сумнівів на предмет вкладених у ньому даних, що безсумнівно є великим плюсом у цій справі. По-третє, docx-файл важить значно менше, ніж його аналог з розширенням doc. Особливо ця різниця помітна в файлах, які містять велику кількість зображень або графіків. Для зберігання docx набагато зручніший, адже він займає мало місця на жорсткому диску.

**Результати роботи.** Як було вже сказано раніше, людське око не в змозі відрізнити незначні відтінки одного й того ж кольору. Цим можна вдало скористатися при побудові алгоритму вкладення таємних даних у контейнер. Суть цього алгоритму полягає в наступному. У нас є повідомлення, яке треба приховати в документ з розширенням docx. При цьому сам документ повинен вже містити у собі текстову інформацію. Від обсягу цієї інформації буде залежати обсяг тих даних, які ми зможемо в нього вкласти. Чим більше тексту містить документ, тим більше даних ми зможемо в нього приховати. Самі дані ми будемо вкладати в RGB канали кольору кожного текстового символу з цього файла. Для цього нам спочатку треба розпарсити документ Word, отримати з нього всі необхідні дані – текст та інформацію про кольори кожного з символів в форматі RGB. Потім отримані складові кольору потрібно перевести в двійкову систему числення і замінити молодші біти складових кольору бітами нашого повідомлення. Більш детально пояснимо це на прикладі:

Це 1 байт нашого повідомлення:

**10 101 010**

Це RGB кольори одного символу:

R: 11110**00**

G: 00001**000**

B: 11001**000**

Замінивши 2 молодших біта у каналі R та 3 молодших біти у каналах Gт а B, отримаємо наступний результат:

R: 11110**10**

G: 00001**101**

B: 11001**010**

Дана операція не внесе в колір помітних людському оку спотворень. Натомість вона допоможе нам вкласти рівно 1 байт нашого повідомлення в колір кожного символу вхідного файла. Тобто максимальна кількість байт (або символів), яку ми можемо приховати, буде дорівнювати кількості символів документу з розширенням docx, включаючи проміжки, табуляції, символи повернення каретки та переводу на новий рядок.

Аналогічним чином виконується і витягнення даних з контейнера. Для того, щоб отримати повідомлення, потрібно, як і в першому випадку, розпарсити документ Word,

отримати кольори текстових символів у форматі RGB і прочитати останні біти кожного каналу. Вони і будуть складати один байт (або символ) прихованих даних. Проробивши ці дії для всіх інших кольорів, ми отримаємо повністю текст секретного повідомлення.

На кафедрі «Програмне забезпечення систем» Дніпровського державного технічного університету (м. Кам'янське) розроблено програмний засіб, який реалізує вищеописаний алгоритм. Також у процесі роботи було написано власний парсер docx-документів, який на відміну від вже існуючих повністю задовольняє вимогам задачі і містить в собі лише необхідні функції, такі як зчитування тексту зі збереженням форматування, зчитування кольорів символів, які використовуються у файлі, і т. ін.

Сама структура додатку нескладна. На головній формі знаходиться 4 пункти меню. Нас будуть цікавити перші два з них: «Приховання даних» та «Витяг даних». Перший пункт відкриває форму, за допомогою якої можна вкласти дані в документ (рис.1).

Маємо декілька груп полів. У першій групі треба вибрати документ формату docx, в який ми збираємося приховати повідомлення, у другій – ввести його текст. Сам текст можна також імпортувати з іншого файла, який має розширення docx або txt. Для цього необхідно поставити відповідний чекбокс, а потім в діалоговому вікні вибрати потрібний нам документ. Зазначимо, що обсяг контейнера для вкладення повинен бути не менший, ніж кількість символів у самому повідомленні. В іншому випадку нам видасть відповідну помилку.

Другий пункт «Витяг даних» відкриває форму, за допомогою якої можна витягти вже приховані дані з контейнера (рис.2). Для цього нам необхідно вибрати файл з прихованим текстом і натиснути кнопку «Отримати повідомлення».

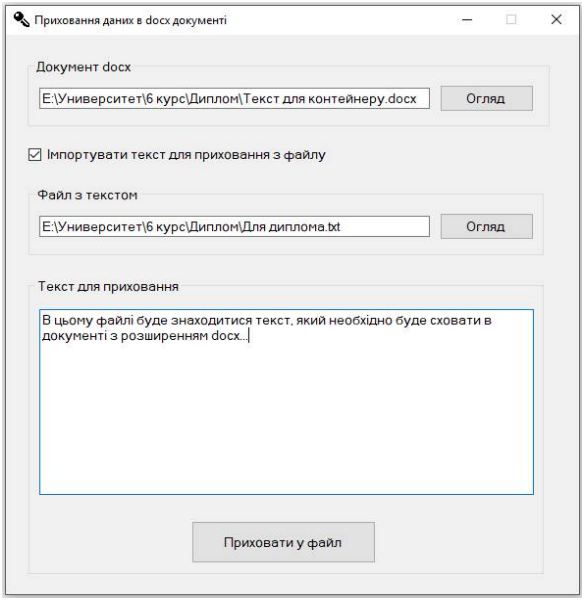


Рисунок 1 – Форма для вкладення даних у контейнер

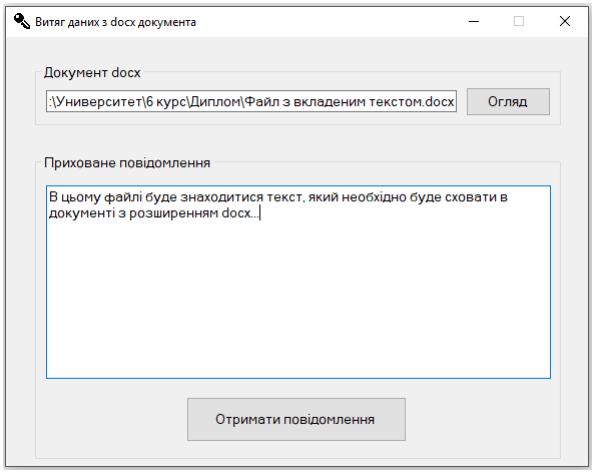


Рисунок 2 – Форма для витягнення даних з контейнера

**Висновки.** Оскільки даний проект орієнтований на сферу захисту даних, кожен, хто зацікавлений даною областю, отримає необхідний результат при використанні даного ПЗ. В процесі роботи розглянуто стеганографічні способи захисту інформації. На базі одного з них (метод LSB) з використанням колірної моделі RGB побудовано алгоритм вкладення прихованих даних в документ Microsoft Word з розширенням docx. Також засобами мови програмування C# було створено програмне забезпечення, яке повністю реалізує даний алгоритм.

Результати, отримані в цій роботі, будуть корисні у науково-технічній сфері та сфері захисту даних саме тому, що вони дадуть змогу як початківцям, так і професіоналам приховувати будь-які повідомлення без використання інших програмних засобів.

ЛІТЕРАТУРА

1. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф.Конахович, А.Ю.Пузыренко. – К.: МК-Пресс, 2006. – 288с.
2. Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Туринцев. – М.: «Солон-Пресс», 2002. – 272с.
3. .Домарев В.В. Безопасность информационных технологий. Системный подход / В.В.Домарев. – К.: ООО "ТИД "ДС", 2004. – 992с.

*Надійшла до редколегії 29.01.2018.*

УДК 004.52

DOI 10.31319/2519-2884.32.2018.176

МІНЯЙЛО Я.О., студент
БАБЕНКО М.В., к.т.н., доцент
ЖУЛЬКОВСЬКИЙ О.О., к.т.н., доцент

Дніпровський державний технічний університет, м. Кам'янське

## СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОТРИМАННЯ НОТНОЇ ГРАМОТИ З МУЗИЧНИХ ФАЙЛІВ ФОРМАТУ MIDI

**Вступ.** Питання, пов'язані з програмуванням музики, «програмованою музикою», «музикою на основі розрахунків» тощо обговорюються досить тривалий час. Завдяки інтелектуалізації персональних комп'ютерів, наявності вбудованих систем аналітичних обчислень, великої кількості діалогових засобів роботи з табличними, текстовими, графічними, музичними об'єктами і т. д., а також у зв'язку з розвитком спеціального програмного забезпечення виникли реальні можливості синтезу композиції з теорією інформації, об'єднання музичних параметрів з акустичними за допомогою серійного комбінування.

Музичне програмування, яке передбачає детальне з'ясування нюансів уявлень про способи функціонування гармонії і тенденції її розвитку, зробило істотний внесок в розвиток сучасних уявлень про музичну гармонію. Дана робота присвячена вибору оптимального шляху перетворення музичних файлів у нотну грамоту завдяки аналізу музичних файлів формату MIDI за допомогою аспектів музикознавства, що допускають формалізований підхід з можливістю навчання гри на фортепіано будь-якої мелодії, що запущена за допомогою програми.

MIDI (англ. *Musical Instrument Digital Interface*, цифровий інтерфейс музичних інструментів) – стандарт передачі інформації між електронними музичними інструментами, розроблений 1983 року, що уможливлює комунікацію електромузичних інструментів, комп'ютера та іншого MIDI-сумісного обладнання, здійснювати з одного інструменту управління іншими.

MIDI не передає звукової інформації, натомість MIDI працює з «повідомленнями», такими як висота та динаміка взятої на інструменті ноти, контрольні сигнали для таких параметрів як гучність, панорама, сигнали відліку часу для синхронізації темпу тощо. Як електронний протокол MIDI відзначається надзвичайно широким поширенням.

**Постановка задачі.** Засобами мови програмування C# необхідно реалізувати алгоритм отримання нотної грамоти з музичних файлів формату MIDI, при цьому процес отримання зробити дуже швидким, що дозволить музикантам в режимі реального часу перетворити будь-яку мелодію в нотну грамоту з можливістю друку. З розвитком технологій музичного програмування в широкому і вузькому розумінні цього процесу виник і новий об'єктивний метод вивчення творчості – моделювання (відтворення або імітація) деяких сторін досліджуваних об'єктів або процесів. Таким чином, для можливості навчання початківців грі на фортепіано можна зімітувати процес гри на інструменті будь-якої мелодії, запущеної через програму, а для професіональних музикантів дуже зручним буде наявність отриманої з музичних файлів нотної грамоти з можливістю регулювання тональності, часової сигнатури, показника тривалості, швидкості відтворення, з можливістю транспозиції та модуляції, а також вибору інструментів для програвання музичного файлу [1].

**Результати роботи.** Підсистема MIDI, на відміну від звукової підсистеми, надає додаткам три (а не два) класи об'єктів: пристрій введення (In), пристрій виведення (Out) і буферизований потік (Stream). Відповідно, є три класи інтерфейсних функцій для обслуговування цих об'єктів. Імена функцій мають відповідні префікси – midiInStart, midiStreamOut тощо. Було виявлено, що найчастіше однойменні функції з різними префіксами розрізняються тільки видами об'єктів, до яких вони належать. Якщо ж сенс і поведінка функцій різняться – вони будуть описуватися і згадуватися окремо.

При відкриванні об'єкта (потоку) підсистема повертає його ідентифікатор або ключ (handle), за яким потім відбувається вся інша робота з об'єктом. Ключ потоку виведення має тип HMIDISTRM і в деяких функціях може використовуватися замість ключа пристрою виведення; в таких випадках потрібно явне приведення до типу HMIDIOUT [2].

MIDI-інтерфейс містить два рівня уніфікації – апаратний, що описує правила з'єднання пристроїв і передачі сигналів, і протокольний, що описує протокол взаємодії і види переданих повідомлень. В даній роботі розглянуто насамперед програмування передачі MIDI-повідомлень в середовищі Windows. Зовні підсистема MIDI дуже схожа на підсистему цифрового звуку Audio/Wave і містить практично той же набір функцій і структур.

Створене програмне забезпечення дозволяє перетворювати музичні файли у нотну грамоту з функцією відтворення на фортепіано будь-якої мелодії, яка програється у додатку. На рис.1 зображено, як у програмі виконується програвання мелодії з отриманням нотної грамоти та навчання тому, як зіграти її на фортепіано. У програмному додатку реалізовані спеціальні засоби мови програмування C# для роботи з нотами і, насамперед, зроблено так, аби кожна нота мелодії, що програється, підсвічувалася. Користувач має можливість обирати кольори для нот, що витягуються. Нотна грамота, отримана з мелодій, буде зберігатися у форматі PNG з можливістю друку.

У файлах MIDI час вимірюється у так званих «імпульсах» і для роботи з часовою складовою створено окремий клас, що буде використовуватися для перетворення тривалості імпульсів у тривалість нот. Ноти з подібним часом імпульсу (наприклад, 10 імпульсів) будуть звучати як акорди. Для полегшення читання нотної грамоти можна зображати ноти у вигляді акордів. В MIDI використовується рівномірно-

темперований лад і 128 нот різної висоти (з номерами від 0 до 127). Частота нот задається за допомогою номера. Наприклад, нота з номером 60 – це «До» першої октави (частота 261 Гц).

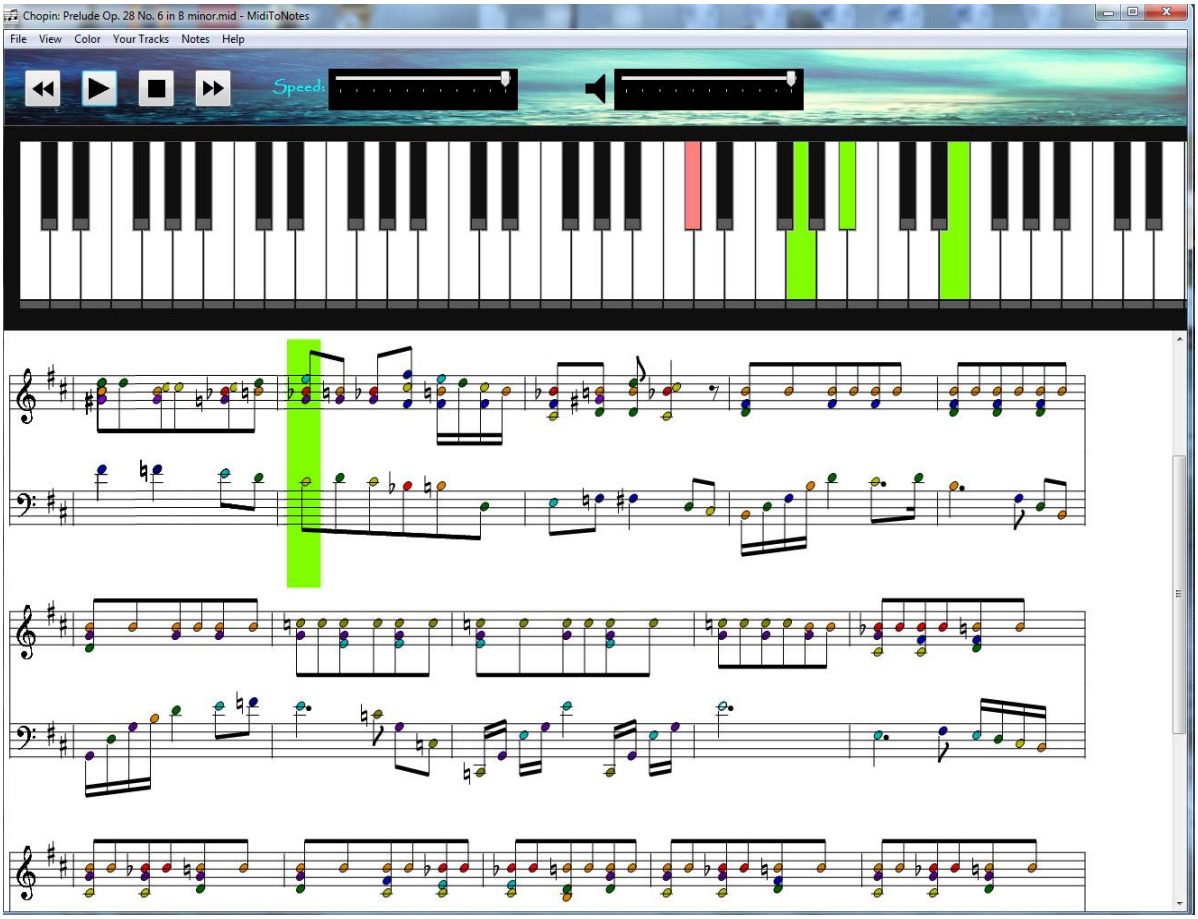


Рисунок 1 – Приклад роботи розробленого програмного забезпечення

Алгоритм отримання нотної грамоти виглядає наступним чином. Спочатку аналізується файл MIDI, отриманий на вході. Далі об’єкт класу, який отримує аналіз, після читання файла буде містити повний список MIDI-подій, музичний розмір пісні, всі доріжки пісні, що складаються з нот, номер, час початку та тривалість кожної ноти. Визначаємо тривалість ноти, потім переходимо до іншої і т.д. В такий спосіб нотна грамота виглядає правильно, ноти та усі музичні символи розміщуються послідовно та збалансовано. Окремий клас відповідає за створення клавіатури фортепіано, за допомогою чого з’являється можливість демонстрації гри на інструменті тієї чи іншої мелодії, що програється.

Для реалізації алгоритму створено класи, які відповідають за музичні (нотні) символи (музичний розмір, ключові знаки, тактові смуги, нотні позначення і т.д.), для кожного типу символів – окремий клас, а також класи, які служать для роботи з форматом MIDI, наприклад, один з класів, що створює символ музичного розміру (дріб, що позначує ритмічну впорядкованість та величину такту в музиці) на початку кожного нотного стану. Біля зображення музичного розміру в нотній грамоті позначаються ключові знаки, за які відповідає окремий клас. Також розроблено клас, який являє собою вертикальні смуги, що обмежують такти. Часом початку тактової смуги є початок нового так-

ту. Крім того, створено клас музичних ключів, що відповідає за створення зображення скрипкового ключа або басового ключа, які можуть бути нормального чи маленького розмірів. Були розроблені класи для роботи з іншими музичними символами, у тому числі і з музичними паузами, що мають початковий час та тривалість. Окремий клас відповідає за можливі помилки при аналізі файлів. Конструктор приймає зсунення файла (в байтах), де сталася помилка, і рядок, що описує помилку. Для читання низькорівневих двійкових даних з файла розроблено клас, який дає можливість прочитати наступний байт у файлі, читати рядки ASCII фіксованої довжини, читати задану кількість байтів у файлі. Один з класів містить доступні налаштування для модифікації нотної музики та звуку.

У додатку є можливість змінювати положення нот (вертикальне чи горизонтальне), збільшувати чи зменшувати нотні знаки. Деякі мелодії складаються з єдиної музичної доріжки, а інші з декількох доріжок (треків). Користувач має можливість обрати необхідний інструмент, за допомогою якого буде програватися доріжка (для кожної доріжки можна обрати свій інструмент).

MIDI є вираженим клавішно-орієнтованим протоколом, тому процес вилучення нот кодується двома простими повідомленнями – взяти ноту (Note On) і зняти ноту (Note Off). Виконавець при натисканні клавіші задає відразу три параметра: момент початку звучання, динаміку і висоту тону. Тривалість звуку визначається за моментом відпускання клавіші. При об'єднанні нот в доріжках створюється єдиний трек. Щоб об'єднати окремі треки розроблено нескладний алгоритм, подібний до сортування злиттям.

MIDI-повідомлення – це потік даних в реальному часі. Компоненти повідомлень в протоколі MIDI представлені байтами. Компонент, що описує тип повідомлення, називається статус-байтом, компонент, що уточнює повідомлення – байтом даних. Якщо уточнюючої інформації забагато, вона може бути представлена кількома байтами даних. Отже, кожне MIDI-повідомлення складається з одного статус-байта і, якщо необхідно, одного або декількох байтів даних. Для того, щоб пристрій міг безпомилково відрізнити статусний байт від байту даних, прийнято, що кожен старший біт статусного байту має значення «1», а кожний старший біт байту даних – «0». У байті даних решта 7 біт відведено для кодування значення того чи іншого параметра, що дозволяє закодувати 128 різних значень. У статусному байті наступні три біта кодують тип повідомлення, а останні 4 біта – один з 16 каналів повідомлення або тип системного повідомлення. Кількість байтів даних жорстко закріплено за кожним повідомленням. Для системних ексклюзивних повідомлень зроблено виняток – їх довжина жорстко не задається. Вона визначається спеціальним статус-байтом, який поміщається в кінець повідомлення.

У процесі розробки програмного забезпечення було виявлено, що для прийому коротких повідомлень у програмі досить передбачити обробник асинхронних подій типу DATA, який буде отримувати повідомлення від підсистеми MIDI кожен раз, коли на вхід інтерфейсу надійде чергове MIDI-повідомлення. Оскільки довжина коротких повідомлень не перевищує 3 байтів, вони передаються оброблювачу в числі параметрів функції або повідомлень, упакованих в змінну типу DWORD, і для їх прийому не потрібно виділення будь-яких буферів в пам'яті. Прийом повідомлень починається відразу ж після звернення до функції Start. Для виведення коротких повідомлень застосовується функція ShortMsg, в якій повідомлення передається цілком, упакованим в змінну типу DWORD [3].

Для прийому і виведення довгих повідомлень необхідно використовувати механізм буферизації за тією ж схемою, що і для запису/відтворення цифрового звуку.

**Висновки.** У сучасному електронному та комп'ютерному музичному інструментарії найбільш повно втілилися накопичені інформаційні технології в музиці. Все це вимагає, з одного боку, підготовки музикантів, які знаються на сучасних музично-комп'ютерних технологіях, а з іншого – підготовки фахівців технічного профілю, що мають основи загального музичної освіти і володіють знаннями в області програмування звуку, звукосинтезу, аудіоінжинірінга, звукотембрального програмування, моделювання музично-творчих процесів і професійно володіють комп'ютерними програмами, фахівців, здатних займатися моделюванням як одним з перспективних методів об'єктивного дослідження музичної творчості.

Створене програмне забезпечення задовольняє нагальній потребі як початківців, так і професійних музикантів отримувати нотну інтерпретацію будь-яких мелодій у форматі MIDI в режимі реального часу з можливістю імітації гри цих мелодій на музичному інструменті.

ЛІТЕРАТУРА

1. Горбунова И.Б. Основы музыкального программирования: учеб. пос. / Горбунова И.Б., Заливадный М.С., Кибиткина Э.В. – СПб.: Изд-во РГПУ им. А.И.Герцена, 2012. – 195с.
2. Горбунова И.Б. Музыкально-компьютерные технологии: к проблеме моделирования процесса музыкального творчества: монография / Горбунова И.Б., Чибирев С.В. – СПб.: Изд-во РГПУ им. А. И. Герцена, 2012. – 160с.
3. Светлов М.Г. Системы искусственного интеллекта в интерактивной музыке, аудиовизуальных инсталляциях и перформансах / М.Г.Светлов // Современное музыкальное образование – 2010: междунар. науч.-практ. конф.: материалы / под общ. ред. И.Б.Горбуновой. – СПб., 2011. – С.131.

*Надійшла до редколегії 29.01.2018.*