Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод нанесення цифрового водяного знаку на тривимірні об’єкти

(тема)

Виконав:

студент II курсу, групи СПм-20-2

Задорожний О.В.
(прізвище, ініціали)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: к.т.н., доц Мартовицький В.О.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри ЕОМ _____ Волк М.О.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Задорожному Олексію Вадимовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод нанесення цифрового водяного знаку на тривимірні об'єкти

затверджена наказом по університету від " 24 " березня 2021 р. № 413 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022 р.

3. Вхідні дані до роботи Тип обладнання – процесор AMD Ryzen 7 5800H, тестові тривимірні моделі, мова програмування – Python

4. Перелік питань, що потрібно опрацювати у роботі
Дослідження методів вбудови цифрових водяних знаків у 3D-модель
Дослідження методів атаки на 3D-моделі
Розробка програмного забезпечення
Оцінка візуального спотворення 3D-моделі
Оцінка якості відтворення цифрових водяних знаків
Оцінка візуального спотворення 3D-моделі
Оцінка якості відтворення цифрових водяних знаків після різноманітних атак

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	30.03.22-05.04.22	
2	Аналіз методів вбудови цифрових водяних	06.04.22-16.04.22	
	знаків у тривимірну модель		
3	Розробка програмного забезпечення для	17.04.22-28.04.22	
	вбудови цифрових водяних знаків у модель		
4	Тестування атак на моделі	29.04.22-05.05.22	
5	Аналіз отриманих результатів	05.05.22-10.05.22	
6	Оформлення матеріалів кваліфікаційної	11.05.22-13.05.22	
	роботи		
7	Подання кваліфікаційної роботи на	14.05.22-18.05.22	
	рецензування		

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ к.т.н., доц Мартовицький В.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 67 с., 15 рис., 1 дод., 15 джерел.

ЦИФРОВИЙ ВОДЯНИЙ ЗНАК, СТЕГАНОГРАФІЯ, 3D-МОДЕЛЬ, СПЕКТР.

Метою кваліфікаційної роботи є дослідження методів вставки цифрових водяних знаків у 3D-модель.

У ході виконання кваліфікаційної роботи було представлено метод вставки водяного знаку трикутної геометричної моделі в спектральну область. Спочатку в роботі наводиться аналіз існуючих методів вставки цифрового в 3D-модель, потім пропонується модифікована схема цифрового водяного знаку в спектральній області геометричної сітки.

Також, розроблено програмний додаток, який реалізує запропонований метод. Нарешті, дається огляд результатів, отриманих після вставки та відтворення цифрових водяних знаків. Показано стійкість алгоритму водяного знаку проти атак шумом та згладжуванням.

ABSTRACT

Master’s thesis: 67 pages, 15 figures, 1 appendices, 15 sources.

FIREWALL, GATE, INTERNET, PROTOCOL, ROUTER, SERVER, WI-FI, WIRELESS NETWORK, WLAN.

The major goal of this thesis is to study the methods of inserting digital watermarks into a 3D model.

In order to achieve this goal, the method of inserting a watermark of a triangular geometric model into the spectral domain was presented. First, the thesis analyzes the existing methods of inserting digital watermark into a 3D model, then offers a modified scheme of digital watermarking in the spectral domain of the geometric mesh.

Also, a software application has been developed that implements the proposed method. Finally, an overview of the results obtained after inserting and reproducing digital watermarks is given. The stability of the watermark algorithm against noise and smoothing attacks is shown.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЦВЗ – цифровий водяний знак

ДКП – дискретне косинус-перетворення

ДВП – дискретне вейвлет-перетворення

ДПФ – дискретне перетворення Фур'є

ПКЛ – перетворення Карунена-Лоєва

ВСТУП

Все більшого значення в нашому світі, що швидко змінюється, набуває захист інформації. Давно існують два напрямки рішення цієї задачі: криптографія та стеганографія. Метою криптографії є приховування змісту повідомлень за допомогою їх шифрування. На відміну від цього, стеганографія приховує сам факт існування таємного повідомлення.

В даний час популярність досліджень в області стеганографії викликана двома причинами: обмеження на використання засобів криптографії в низці країн світу і поява проблеми захисту прав власності на інформацію, представлену в цифровому вигляді. Перша причина спричиняє за собою велику кількість досліджень приховання факту передачі інформації, друга – численні роботи в області цифрових водяних знаків (ЦВЗ).

Із швидким розвитком веб-технологій, використання цифрових мультимедіа (аудіо, відео, зображення тощо) набуло широкого поширення. За рахунок розвитку цих речей, можна просто отримати та відтворити інтелектуальну власність. Тому потрібен захист вмісту, тому для цього існує така техніка, як цифрові водяні знаки, яка є одним із найефективніших способів захисту цифрової власності об’єкта.

Більшість великих інтернет-магазинів перед викладанням продукції автора накладають цифрові водяні знаки на неї. Як продукція виступають постановочні фотографії, панорами, обкладинки та вкладки музичних альбомів та відеофільмів. ЦВЗ містять інформацію, що однозначно підтверджує авторство або права на комерційне використання зображення, що захищається, яка може бути рахована для вирішення спірних правових ситуацій. Для маркування комерційної продукції цифровими водяними знаками потрібно передбачити такий момент, що в мережах зазвичай викладаються цифрові моделі, які проходять стиснення за певним алгоритмом з метою зменшення обсягу. Зазвичай застосовується стиснення з

втратами, при використанні якого розпаковані дані відрізняються від вихідних, але відмінність не є істотною з точки зору їх подальшого використання. Тому потрібно передбачити, щоб інформація, що вбудовується, була стійка до такого стиску. Тому задача створення методів і алгоритмів, використання яких при побудові стеганографічних систем захисту авторських прав для моделей може гарантувати цілісність ЦВЗ є актуальною.

У методах дослідження використовувалися: методи теоретичного та емпіричного дослідження, апарати обчислювальної математики, методи проектування та програмування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основи стеганографії

Цифрова стеганографія – це напрямок класичної стеганографії, заснований на приховуванні або впровадженні додаткової інформації в цифрові об'єкти, викликаючи деякі спотворення цих об'єктів. Дані об'єкти є мультимедіа файлами (зображення, відео, аудіо, текст) та внесення спотворень, що знаходяться нижче за поріг чутливості середньостатистичної людини, не призводить до помітних змін цих об'єктів. Сьогодні стеганографія дозволяє не лише успішно вирішувати основне завдання – потай передавати інформацію, а й цілу низку інших актуальних завдань, у тому числі вбудова прихованої інформації з метою захисту авторських прав на інтелектуальну власність, представлену в цифровому вигляді [1]. Ця інформація, що приховується, називається цифровим водяним знаком (ЦВЗ), який являє собою спеціальну мітку, що містить інформацію, однозначно підтверджує авторство або права на комерційне використання об'єкта, що захищається. Вона непомітно впроваджується у зображення чи інший сигнал з метою тим чи іншим чином контролювати його використання.

Цифровий водяний знак – це дані або мітка, прикріплена до цифрового об’єкту, яку можна ідентифікувати (витягнути), щоб заявити права на об’єкт. Лише конкретне програмне забезпечення та системи можуть розпізнати унікальну інформацію, яку утворює ЦВЗ (наприклад, дату та місце її створення, авторські права).

Існуючі методи, що вирішують задачу захисту авторського права шляхом вбудовування ЦВЗ, можна розділити на дві групи: група методів, які приховують інформацію в просторовій області зображення та методи, що вбудовують ЦВЗ в частотну область. Методи першої групи вбудовують інформацію безпосередньо в первинну область даних зображення, що робить

їх нестійкими до багатьох спотворень, особливо до компресії з втратами (наприклад JPEG-компресія). Це призводить до часткового чи навіть повного знищення вбудованого ЦВЗ. Більш стійкими до різного роду спотворень та компресії є методи другої групи. До відомих методів відносяться методи на основі використання дискретного косинус перетворення (ДКП), дискретного перетворення Фур'є (ДПФ), вейвлет-перетворення, перетворення Карунена-Лоєва та ін. Найбільш поширеними перетвореннями в стеганографії є ДКП та вейвлетперетворення, тому що крім можливості використання в стеганографічних перетвореннях, вони ефективно використовуються під час ущільнення зображень.

Основною вимогою вбудовування ЦВЗ є та, що стеганосистема повинна забезпечувати незмінність вбудованої інформації при спотворенні чи компресії зображення-контейнера та мінімальний вплив методу вбудовування ЦВЗ на якість самого зображення. Серед стеганосистем, які вирішують ці задачі, виділяють декілька типів: конфіденційні, напівконфіденційні, напіввідкриті та відкриті стеганосистеми. Така класифікація визначає, яка інформація потрібна системі для того, щоб виявити ЦВЗ – оригінал зображення, ЦВЗ, секретний ключ чи додаткова інформація. Стеганоситеми перших двох типів вимагають наявності оригіналу зображення чи ЦВЗ, та знання секретного ключа. Напіввідкриті стеганосистеми виявляють ЦВЗ за допомогою секретного ключа, який залежить від оригіналу зображення. Відкриті стеганосистеми для своєї роботи, окрім секретного ключа, не вимагають ні знання оригінального зображення, ні вбудованого ЦВЗ. Слід відзначити, що хоча більшість існуючих на сьогодні стеганосистем відносяться до конфіденційного або напівконфіденційного типу, перспективними є дослідження та розробка відкритих систем цифрових водяних знаків.

До стеганографічних методів, що представляють відкриті стеганосистеми та приховують інформацію в частотну область зображення, відносять відомі методи Коха і Жао, Бенгама-Мемона-Ео-Юнг та Фрідріха.

Основними проблемами при реалізації цих методів є суттєве руйнування чи знищення ЦВЗ при високих коефіцієнтах ущільнення зображення та афінних перетвореннях, а також пов’язане з цим помітне погіршення якості зображення. Тому актуальними є дослідження, спрямовані на розробку відкритих стеганосистем, в яких вирішувались би вказані проблеми.

1.2 Вимоги до систем вбудови цифрового водяного знаку

Зберігання зображень у цифровому форматі спрощує їх зберігання та розповсюдження, але також збільшує ризик порушення авторських прав, несанкціонованої зміни та розповсюдження. З метою захисту інтелектуальної власності та визначення змін, розробляються та застосовуються цифрові водяні знаки. До таких стеганографічних методик пред'являються особливі вимоги:

- якість вихідного зображення не повинна бути серйозно порушена, помітність прихованих даних повинна бути мінімальною;
- приховані дані повинні зберігатися у різних форматах, тобто утримуватися не лише у заголовку, а у всьому тілі цифрового об'єкта;
- приховані дані повинні бути стійкими до навмисних спроб видалення;
- необхідна наявність надлишкового коду для корекції помилок, оскільки деградація даних під час передачі/модифікації неминуча.

Основне завдання будь-якої стегосистеми – вбудувати повідомлення в контейнер так, щоб сторонній спостерігач не міг помітити різницю між оригінальним та модифікованим контейнерами. Зазвичай система будується так, щоб забезпечити певний компроміс її базових характеристик, до яких відносяться непомітність, стійкість, захищеність, пропускна здатність утвореного стеганоканалу та обчислювальна складність реалізації.

Аналіз існуючої літератури показав, що немає чіткого розмежування між поняттями стійкості (robustness) та безпеки (security). Часто ці поняття

вживаються взаємозамінно. Безпека – це базова характеристика, основним завданням якої є захист від умисних атак порушника, а робастність повинна забезпечувати захист від інших видів атак. Підкреслимо, що поняття стійкості не включає в себе атаки на методи вбудовування, що ґрунтуються на знаннях алгоритму вбудовування або вилучення. Під стійкістю мається на увазі, стійкість до нецільових модифікацій, або узагальнені операції з зображеннями.

Термін "робастність" (robustness - англ.) утворений від robust – міцний, грубий (англ.). Мається на увазі, що робастні процедури повинні "витримувати" помилки, які тими чи іншими способами можуть потрапляти в вихідні дані або спотворювати передумови використовуваних ймовірносно-статистичних моделей. Термін "робастний" спочатку використовувався фактично як звуження терміна "стійкий" на алгоритми статистичного аналізу даних класичного типу (не включаючи теорію вимірювань, статистику нечислових та інтервальних даних). Робастність – це здатність прихованого повідомлення залишатися неушкодженим, навіть якщо стего-медіа піддають трансформації, лінійній та нелінійній фільтрації, масштабуванню, розмиванню, обрізанню та іншим атакам. Іншими словами – це стійкість ЦВЗ до різного роду перешкод і спотворень.

ЦВЗ можуть бути трьох типів: робастні, крихкі й напівкрихкі (semifragile).

Крихкі водяні знаки знищуються при невеликій зміні заповненого контейнера. Їх використовують для розпізнавання сигналів. Для захисту медіа інформації це надзвичайно важливо, оскільки законний користувач або, навіть, сам автор може захотіти змінити дані, наприклад стиснути зображення. Також відзначимо, що крихкі ЦВЗ мають не тільки вказати на факт модифікації контейнера, але й також вигляд та розташування цієї модифікації.

Напівкрихкі ЦВЗ є стійкими до одних атак і нестійкими до інших. Загально кажучи, всі ЦВЗ можна віднести до цього типу. Проте напівкрихкі

ЦВЗ навмисно проектуються так, щоб бути нестійкими відносно певних операцій. Наприклад, вони допомагають стискати зображення, але забороняють вирізати або вставляти в нього сторонній фрагмент.

Невидимість (невідчутність, imperceptibility) – характеристика, що відповідає за нездатність людським зором без використання спеціальних засобів виявити приховане повідомлення. Скрита інформація непомітна, якщо людина не може відрізнити носій з прихованою інформацією від носія без неї. Загальноприйнята схема експерименту, яку часто називають "сліпий тест", заснована на тому, що суб’єктам пропонують в довільному порядку серед великої кількості носіїв з та без вбудованої інформації обрати, які саме носії містять приховані дані.

За наявністю ключа стеганографічні методи ділять на три групи: безключеві, з ключем та гібридні (змішані). Таким чином, захищеність безключевої стеганосистеми базується лише на секретності стеганографічних перетворень, які використовуються. Ключова стеганосистема в свою чергу поділяється на підсистеми з відкритим та закритим ключем. Стегосистема з відкритим ключем повинна мати закритий канал зв’язку для передачі стегоключа і повинна забезпечувати вищий рівень захищеності повідомлення, ніж система без ключа. Також така система потребує більше затрат на передачу стегоключа. Стеганосистема з відкритим ключем працює аналогічно з криптографічними алгоритмами, проте необхідно зазначити, що стеганоключ приховує місце вбудовування даних в контейнері, а не шифрує їх. Гібридні стегосистеми можуть використовувати і відкритий, і секретний ключ.

Складність вбудовування і вилучення – кількість операцій та дій, що мають бути виконані для вбудовування і викриття прихованого повідомлення. Стеганосистема повинна мати прийнятну обчислювальну складність реалізації. До того ж реалізація стеганографічної системи передачі інформації може бути асиметричною за складністю, наприклад складний стеганокодер і простий стеганодекодер і навпаки.

Перераховані вимоги конкурують одна з одною і не можуть бути оптимальними разом. Якщо необхідно приховати велике повідомлення в зображенні, то неможливо вимагати повної невидимості й хорошої стійкості. Необхідно знайти компроміс. З іншого боку, якщо потребується стійкість до великих спотворень, то повідомлення, що має бути надійно сховане, не може бути дуже довгим. Це можна побачити на рисунку на рисунку 1.1.

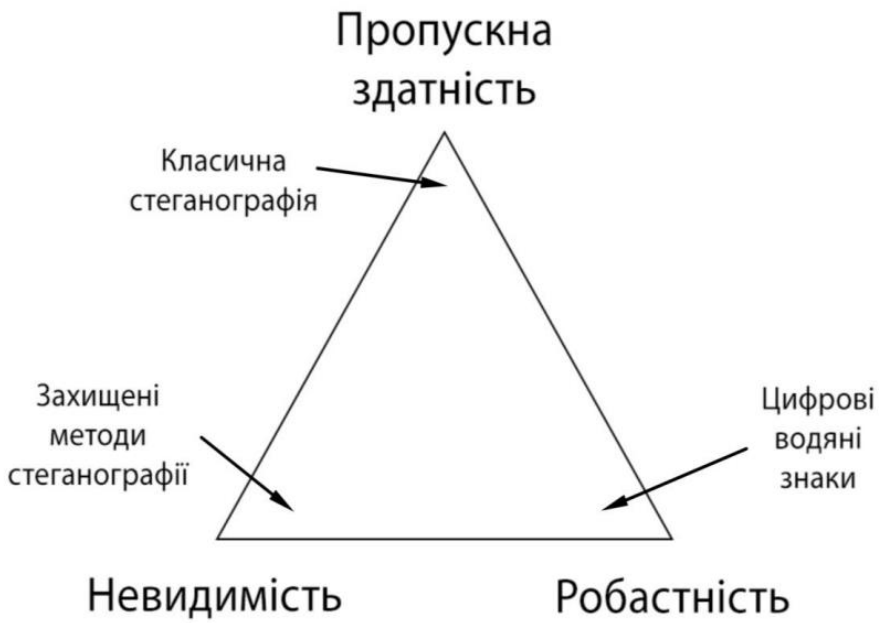

Рисунок 1.1 – Трикутник ключових характеристик стеганосистем

1.3 Сфери застосування цифрових водяних знакі

Вбудовування ЦВЗ у медіафайли може бути використане для таких цілей:

- вбудова інформації і її прихованої передачі. Цей напрямок використовується з метою захисту конфіденційної інформації від несанкціонованого доступу та безпечної її передачі через комп'ютерні мережі. Також подібний напрямок є привабливим, коли уряд країни накладає серйозні обмеження на використання засобів шифрування;

- вбудова ЦВЗ для захисту авторських прав на інтелектуальну

власність, представлену в цифровому форматі. Правовласник або видавець може впровадити ЦВЗ, що містить інформацію про авторство в продукт, що захищається. Впроваджений ЦВЗ може бути використаний на підтвердження прав власності. Найбільші досягнення стеганографії були досягнуті саме в цій галузі, і вона перебуває у постійному розвитку на даний момент;

- маркування ідентифікаційними номерами для відстеження шляхів розповсюдження нелегальних копій продукту за допомогою техніки унікального підпису для кожної легальної копії. У цьому випадку власник може впроваджувати різні ЦВЗ для різних замовників. ЦВЗ може містити інформацію про серійний номер, що однозначно ідентифікує покупця, який порушив ліцензійну угоду та надав продукт для незаконного розповсюдження або копіювання;

- вбудова для захисту від копіювання медіафайлу. Впроваджений ЦВЗ може безпосередньо контролювати цифрові записуючі або друкувальні пристрої. Детектор ЦВЗ на записувальному пристрої визначає, чи може інформація надана пристрою бути скопійована;

- моніторинг широкомовних каналів. Таким чином, можна перевірити за допомогою автоматизованої системи, чи виконується контракт на трансляцію комерційної інформації з впровадженим ЦВЗ;

- вбудова для перевірки цілісності переданих даних. Впровадження ЦВЗ дозволяє перевірити дані щодо зміни чи пошкодження як навмисного, і випадкового характеру. Також можливе визначення, в якій саме частині даних було здійснено зміни;

- вбудова для індексації частин файлу. Допустимо якщо вбудувати ЦВЗ у відеопослідовність, то можна полегшити завдання пошуковому движку;

- вбудова ЦВЗ для підпису медичних знімків або нанесення легенди на карту. Метою є зберігання різнорідно поданої інформації у цілому. Впровадження інформації допоможе уникнути плутанини та забезпечить зручність зберігання інформації.

2 АНАЛІЗ МЕТОДІВ СТЕГАНОГРАФІЇ

У 1989 був отриманий перший патент на спосіб прихованого вкладення інформації в зображення шляхом модифікації молодшого біта (LSB). У разі детектор аналізує лише значення цього біта кожному за пікселя, а око людини, навпаки, приймає лише старші 7 біт. Науково підтверджено факт, що система людського зору найменш чутлива до змін інтенсивності в синій області спектра. Таким чином, можна з великою впевненістю заміняти молодший біт байта, який відповідає за інтенсивність синього каналу, за обраною закономірністю. Людському оку буде важко відрізнити оригінальне чисте зображення від зображення з вбудованим прихованим повідомленням [2]. Даний метод є типовим представником методів вбудовування у просторову область зображення, при якому для вбудовування використовуються безпосередні зміни значень параметрів яскравості та кольоровості зображень. Існує багато модифікацій цього методу, але на даний момент багато з них застаріли.

2.1 Стеганографічна система

Стеганографічна система (стегосистема) – це сукупність методів та інструментів для створення прихованого каналу для передачі даних. При побудові такої системи було вирішено, що:

- ворог знає роботу стеганографічної системи. Невідомим для противника є ключ, за допомогою якого можна дізнатися про факт існування та змісту таємного повідомлення;
- при виявленні противником наявності прихованого повідомлення він повинен змогти витягти повідомлення до того часу поки він володіє ключем;
- противник не має технічних та інших переваг.

Повідомлення – це загальний термін для будь-якої секретної

інформації, яка передається, у формі листа чи цифрового файлу.

Контейнер – так називається будь-яка інформація, яка використовується для приховування секретного повідомлення. Порожній контейнер – це контейнер, який не містить секретного повідомлення. Заповнений контейнер (стегоконтейнер) – це контейнер, який містить секретне повідомлення.

Стеганографічний канал (стегоканал) – канал передачі стегоконтейнера.

Ключ (стегоключ) – секретний ключ, необхідний для приховування стегоконтейнера. Ключі в стегосистемах бувають двох типів: секретні та відкриті. Якщо стегосистема використовує секретний ключ, то він має бути створений або до початку обміну повідомленнями, або переданий захищеним каналом. Стегосистема, що використовує відкритий ключ, повинна бути влаштована таким чином, щоб неможливо було отримати з нього закритий ключ. У цьому випадку відкритий ключ можна передавати незахищеним каналом.

Стегосистема відповідає за вбудовування та виділення повідомлень з іншої інформації. Нижче наведено основні компоненти стегосистеми:

- прекодер – це пристрій, призначений для перетворення прихованого повідомлення до вигляду, зручного для вбудови на сигнал-контейнер;
- контейнер – це інформаційна послідовність, в якій зберігається повідомлення;
- стегокодер – це пристрій, призначений для здійснення вкладення прихованого повідомлення в інші дані з урахуванням їхньої моделі;
- стегодетектор – пристрій, призначений для визначення наявності стегоповідомлення;
- декодер – пристрій, який відновлює приховане повідомлення. Цей вузол може бути відсутнім, якщо нам потрібно лише встановити факт наявності в об'єкті вбудованого раніше ЦВЗ.

Далі розглянемо докладніше поняття контейнера. У сучасній цифровій

стеганографії контейнером може виступати музичний файл (найпопулярніші формати WAV і MP3), зображення (формати JPEG та BMP), відео (формати AVI та MPEG), текстові файли (формати DOC та PDF) та тривимірні об’єкти (OBJ, STL). Стегоконтейнер повинен бути візуально відмінним від порожнього контейнера. Контейнери поділяються на дві категорії: потокові та фіксовані. Використовуваний контейнер має значний вплив на надійність і стабільність стегосистеми, а також на її здатність виявити факт передачі секретного повідомлення.

Наступна бітова послідовність постійно передається в потоковий контейнер. Оскільки повідомлення вбудовується в режимі реального часу, кодер не може дізнатися, чи буде достатньо місткості контейнера для відправки повного повідомлення. Один великий контейнер може містити кілька повідомлень. Генератор псевдовипадкових послідовностей з рівномірним розподілом інтервалів між вибірками визначає інтервали між вбудованими бітами. Найскладнішою частиною є синхронізація та визначення початку та кінця серії. Якщо контейнер даних містить біти синхронізації, заголовки пакетів та іншу інформацію, за ними можуть слідувати приховані дані. З точки зору гарантування таємності передачі, проблема забезпечення синхронізації стає перевагою. Розміри та властивості фіксованого контейнера також відомі заздалегідь. Це дає змогу найкращим чином об’єднати дані. Контейнери фіксованої довжини, з іншого боку, обмежені за розміром, і вбудоване повідомлення може не завжди поміщатися у файлі контейнера. Інша проблема полягає в тому, що відстані між бітами, що приховуються, рівномірно розподілені між найкоротшим і найбільшим визначеними інтервалами, тоді як справжній випадковий шум має експоненційний розподіл довжини інтервалу. Звичайно, псевдовипадкові експоненціально розподілені числа можуть бути згенеровані, але цей метод, як правило, занадто складний в обчисленні. Насправді, контейнери фіксованої довжини є найпоширенішими і легкодоступними.

Контейнер може бути вибраним, згенерований випадковим чином або

нав’язаним. Вибраний контейнер визначається вбудованим повідомленням, а в гіршому випадку – його функцією. Цей тип контейнера робить стеганографію більш помітною. За обставин, коли особа, яка надає контейнер, підозрює ймовірну таємну кореспонденцію та бажає запобігти цьому, може з’явитися нав’язаний контейнер. Насправді, найчастіше зустрічається випадковий контейнер.

Існують різні алгоритми застосування ЦВЗ для кожної форми контейнера, кожен з яких спеціалізується на цьому типі стегоконтейнера. Стегоконтейнери можуть поставлятися в різних форматах, що обмежує реалізацію стегоалгоритму.

2.2 Класифікація методів стеганографії

На даний момент існує безліч різних варіантів вибору області вбудовування ЦВЗ. Класифікація алгоритмів за способами вбудови наведено на рисунку 2.1.

Рисунок 2.1 – Класифікація алгоритмів вбудови ЦВЗ

Для вбудови інформації використовується область роздільної здатності або частотна область. Ці підходи з'явилися пізніше за попередній і

продовжують розвиватися. Методи, що використовують для приховування даних частотну область, є стійкішими до різних можливих зовнішніх впливів на зображення-контейнер. У цій групі використовуються досить різноманітні трансформації:

- дискретне косинус-перетворення (ДКП);
- вейвлет-перетворення (ДВП);
- дискретне перетворення Фур'є (ДПФ);
- перетворення Карунена-Лоєва (ПКЛ);
- сингулярне розкладання.

Ці методи використовують переваги, якими володіє представлення зображення кінцевим набором коефіцієнтів. Такі методи мають добрі характеристики робастності.

2.3 Техніки вбудови цифрових водяних знаків

У сфері ЦВЗ, вбудова водяних знаків у цифрові зображення привернула велику увагу дослідників з двох причин: перша — це його доступність, а інша – це передача достатньої кількості додаткової інформації, яку можна було б використовувати для вбудовування ЦВЗ. Техніка вбудови ЦВЗ у зображення завжди працює в двох областях: просторовій та області перетворень. Технології просторової області працюють безпосередньо з вершинами. Найбільш часто використовуваним методом просторової області є LSB. Методи області перетворення вставляють водяний знак шляхом зміни коефіцієнтів просторової області. Найбільш часто використовуваними методами з області перетворення є ДКП, ДВП і ДПФ. Для досягнення надійності та непомітності методи області перетворення є більш ефективними, ніж методи просторової області.

2.3.1 Техніки вбудови цифрових водяних знаків у просторовій області

Просторова область представляє модель у вигляді вершин. Вбудова ЦВЗ відбувається шляхом зміни позиції деяких вибраних вершин. Сильні сторони систем ЦВЗ просторової області:

- простота;
- дуже низька обчислювальна складність;
- менше витрат часу.

Вбудова ЦВЗ методами просторової області простіше, при чому їх обчислювальна швидкість висока порівняно з методами з частотної області, але вони менш стійкі до атак. Методи просторової області можна легко застосувати до будь-якої моделі. Найважливішим методом просторової області є LSB.

Метод найменш значущого біту (LSB) LSB – це найпростіша техніка вбудови ЦВЗ в просторовій області для вбудови знака в найменш значущі біти деяких вибраних вершин моделі. Основна перевага цього методу полягає в тому, що він легко виконується на зображеннях. І це забезпечує високу прозорість сприйняття. При вбудові ЦВЗ за допомогою LSB якість зображення не погіршується. Основним недоліком техніки LSB є її погана стійкість до звичайних операцій обробки сигналів, оскільки за допомогою цієї техніки водяний знак можна легко знищити будь-якими атаками. Цей метод не вразливий до атак і шуму, але дуже непомітний.

Адитивні ЦВЗ. Найпростіший метод для вбудовування ЦВЗ в просторовій області – це додати псевдовипадковий шаблон шуму до координат вершин. Сигнал шуму зазвичай являє собою цілі числа (-1, 0, 1) або іноді числа з плаваючою точкою. Щоб гарантувати, що водяний знак може бути виявлений, шум створюється ключем, так що кореляція між номерами різних ключів буде дуже низькою.

Методика на основі модуляції SSM. Методи розширеного спектру – це методи, за допомогою яких енергія, що генерується на одній або кількох

дискретних частотах, навмисно поширюється або розподіляється в часі. Алгоритми ЦВЗ на основі SSM вбудовують інформацію шляхом лінійного поєднання основної моделі з невеликим псевдошумовим сигналом, який модулюється вбудованим водяним знаком.

Техніка на основі кореляції. У цій техніці шаблон псевдовипадкового шуму (PN) названий $W(x, y)$, додається до вершин моделі $I(x, y)$.

$$Iw(x, y) = I(x, y) + k * W(x, y), \tag{2.1}$$

де k – коефіцієнт посилення;

Iw – модель водяного знаку та положення x, y;

I – оригінальна модель.

При збільшенні коефіцієнту посилення збільшується надійність водяного знака, але якість отриманого зменшиться.

2.3.2 Техніки вбудови цифрових водяних знаків у частотній області

Порівняно з методами в просторовій області, методи в частотній області застосовуються ширше. Метою є вбудовування цифрових водяних знаків у спектральні коефіцієнти моделі. Найбільш часто використовуваними перетвореннями є дискретне косинусне перетворення (ДКП), дискретне перетворення Фур'є (ДПФ), дискретне вейвлетне перетворення (ДВП). Причина вбудови цифрових водяних знаків у частотній області полягає в тому, що характеристики зорової системи людини (ЗСЛ) краще фіксуються спектральними коефіцієнтами.

Дискретні косинусні перетворення (ДКП), як і перетворення Фур'є, представляє дані в рамках частотного, а не амплітудного простору. Це корисно, оскільки це більше відповідає тому, як людина сприймає світло, так що іншу частину можна визначити та викинути. Методи нанесення цифрових водяних знаків на основі ДКП є надійними порівняно з методами просторової

області. Такі алгоритми стійкі до простих операцій обробки. Однак їх важко реалізувати, а також вони є дорожчими з точки зору обчислень. У той же час, вони слабкі проти геометричних атак, таких як обертання, масштабування, обрізання тощо. Вбудову ЦВЗ в області ДКП можна класифікувати на глобальні ДКП і ті, що побудовані на основі блоків. Вбудовування в перцептивно значущу частину моделі має свої переваги, оскільки більшість схем стиснення видаляє незначну для сприйняття частину моделі.

Дискретні вейвлетні перетворення (ДВП). Вейвлет-перетворення – це сучасна техніка, яка часто використовується в цифровій обробці даних, стиснення, цифрових водяних знаків тощо. Перетворення засновані на малих хвилях, які називаються вейвлетами, різної частоти та обмеженої тривалості. Вейвлет-перетворення розкладає модель на три просторові напрямки: горизонтальний, вертикальний і діагональний. Отже, вейвлети більш точно відображають анізотропні властивості ЗСЛ. Величина коефіцієнтів ДВП більша в найнижчих діапазонах (LL) на кожному рівні розкладання і менша для інших діапазонів (HH, LH і HL). Дискретне вейвлетне перетворення (ДВП) в даний час використовується в широкому спектрі програм обробки сигналів, таких як стиснення аудіо та відео, видалення шуму в аудіо та моделювання розподілу бездротової антени. Енергія вейвлетів зосереджена в часі і добре підходять для аналізу перехідних сигналів, що змінюються в часі. Оскільки більшість реальних сигналів, які зустрічаються в реальному житті, мають різний характер у часі, вейвлетне перетворення дуже добре підходить для багатьох застосувань [3]. Однією з головних викликів проблеми ЦВЗ є досягнення кращого компромісу між робастністю та сприйнятливістю. Робастність може бути досягнута шляхом збільшення міцності вбудованого цифрового водяного знаку, але видиме перетворення також буде збільшено [3]. Однак ДВП є більш доцільним, оскільки воно забезпечує як одночасну просторову локалізацію, так і частотне розповсюдження цифрового водяного знака всередині основної моделі [4]. Основна ідея дискретного вейвлет-перетворення в процесі моделі полягає в багатодиференційованому

розкладанні моделі на меши з різними просторовими областями та незалежними частотами[5].

Дискретне перетворення Фур'є (ДПФ) перетворює безперервну функцію в її частотні компоненти. Воно стійке до геометричних атак, таких як обертання, масштабування, обрізання, перенос тощо. ДПФ показує незмінність переносу. Просторові зсуви в моделі впливають на фазове представлення моделі, але не на відображення величини, або кругові зсуви в просторовій області не впливають на величину перетворення Фур'є.

2.3.3 Переваги й недоліки техніки вбудови цифрових водяних знаків у частотній області

Вейвлет-перетворення краще розуміє ЗСЛ, ніж ДКП. Модель можна відображати з різними рівнями деталізації та послідовно обробляти від низької до високої деталізації. Але обчислювальна складність ДВП більша в порівнянні з ДКП [6]. Як зазначив Фейг (1990), для обчислення ДКП для блоку 8x8 потрібно лише 54 множення, на відміну від вейвлет-розрахунку залежно від довжини використовуваного фільтра, який становить принаймні 1 множення на коефіцієнт.

ДПФ, у свою чергу, перетворює безперервну функцію в її частотні компоненти. Воно стійке до геометричних атак, таких як обертання, масштабування, обрізання, перенос тощо. ДПФ показує незмінність переносу. Просторові зсуви в моделі впливають на фазове представлення моделі, але не на відображення величини, або кругові зсуви в просторовій області не впливають на величину перетворення Фур'є.

ДПФ є інваріантом обертання, масштабування та переносу. Тому його можна використовувати для відновлення від геометричних викривлень, тоді як просторова область, ДКП і ДВП не є інваріантними до переносу, і тому його важко витримати геометричні перетворення.

3 КОНЦЕПЦІЯ МЕТОДІВ ПОБУДОВИ ТРИВИМІРНИХ МОДЕЛЕЙ

3.1 Модель у комп’ютерній графіці

У комп’ютерній графіці тривимірна модель зазвичай представлена полігональною сіткою, яка являє собою набір полігональних граней, націлених на наближення до реальної форми тривимірного об’єкту. Складні, багатополігональні моделі можуть бути поділені на окремі об’єкти геометрії, так звані полігональні сітки (Mesh). Основною причиною розділення моделі є той факт, що у кожної ділянки об’єкта може бути різний матеріал, що по-різному відбиває або поглинає світло, має різні текстури тощо.

3D-моделі зберігають тривимірну інформацію у вигляді звичайного тексту або бінарних даних для 3D-моделі. Існують багато типів 3D-файлів, оскільки кожна програма має свій власний формат файлу, який оптимізовано для цього конкретного програмного забезпечення. Наприклад, Blender має BLEND, AutoCAD має DWG, Clo має .zprj тощо.

Тривимірна модель складається з безлічі точок, які з'єднуються між собою гранями і утворюють полігони. Вершина – це точка, яка має свої координати в тривимірній системі, тобто X, Y, Z. Свою назву вона отримала через те, що є крайньою точкою плоского багатокутника, або полігону. Грань, або ребро – це відрізок, який з'єднує дві вершини. У тривимірній графіці гранню називають обмежувач полігонів. Основною складовою в тривимірній графіці вважається полігон – плоский багатокутник, безліч яких і утворює тривимірну фігуру. Сукупність полігонів несе інформацію про розмір і форму 3D-моделі, а обрана текстура дозволяє передати достовірну інформацію про зовнішній вигляд об'єкта і являє собою зображення на поверхні фігури.

Існує кілька видів тривимірних моделей:

- полігональна модель;

- NURBS поверхні.

NURBS поверхні мають більш високий рівень точності, так що їх найчастіше використовують інженери, машинобудівники і архітектори. А ось полігональні моделі частіше використовуються для створення 3D-зображень в мультиплікації, кінематографі та комп'ютерних іграх. Вони складаються з численних найпростіших геометричних фігур, які також називають примітивами.

Крім того, існує три різні типи 3D-моделювання:

- каркасне моделювання;
- поверхневе моделювання;
- твердотільне моделювання.

Каркасне моделювання – це найпростіший вид. Дротяні або каркасні моделі – це ті, що створюються цим способом відтворення. Вони складаються з ліній, дуг та сегментів. Цей тип зображення не передає всієї інформації про об’єкт: не можна зрозуміти обсяг або структуру поверхні такої моделі, але можна вивчити її пристрій і функціональність. Головною перевагою каркасного моделювання є те, що для зберігання тривимірних моделей, виготовлених таким чином, не потрібно багато оперативної пам’яті. Каркасна візуалізація найчастіше використовується в спеціалізованих програмах для створення передбаченої траєкторії руху пристрою або інструменту.

Другий вид 3D-моделювання – це поверхневе моделювання. На відміну від каркасного, тут є не тільки сегменти, лінії і дуги, а й поверхні, що утворюють контур об'єкту, що відображається.

Останній, найточніший і достовірний тип 3D-моделювання, називається твердотільне моделювання. В результаті його використання можна отримати справжній зразок готового об'єкта, який передає всі дані про нього. Модель, створена завдяки цьому способу візуального відтворення, містить лінії, межі, текстуру і дані про обсяг і масу тіла. Хоча зображення і займають найбільший обсяг пам'яті комп'ютера в порівнянні з іншими, але

він повністю описує готовий об'єкт. Твердотільне моделювання використовується всюди: при створенні техніки, промислових деталей, меблів, ювелірних виробів, кіно і комп'ютерних ігор.

Найпопулярніші формати тривимірних моделей:

- OBJ: є нейтральним 3D-форматом, якщо використовується як варіант ASCII. Однак, якщо він використовується в бінарному форматі, він є пропрієтарним. 3D-принтінг, графіка та 3D-сканування використовують цей формат файлу частково завдяки його здатності зберігати геометрію, а також інформацію про колір і текстуру. Цей формат файлу зберігає інформацію про колір і текстуру в окремому файлі з розширенням .MTL. OBJ не підтримує анімацію, але є одним з найпопулярніших форматів обміну для 3D-графіки;
- STL: формат файлу 3D, який є найбільш популярним для 3D-друку. Це нейтральний формат 3D-файлів, який зберігає лише інформацію про геометрію;
- FBX: це пропрієтарний формат 3D-файлів. Він широко використовується в індустрії кіно та відеоігор. Він підтримує геометрію, зовнішній вигляд (колір і текстура), а також анімацію. FBX є найбільш популярним для анімації і використовується як формат обміну між різними програмами, такими як Maya, 3DSMax, AutoCAD, Roman’s CAD та іншими;
- 3DS: є одним із форматів файлів, які використовуються програмним забезпеченням Autodesk 3ds. Протягом тривалого часу він був загальновизнаним стандартом для передачі інформації між 3D програмами. В результаті він підтримується майже всіма програмними пакетами 3D і може бути відкритий більшістю інструментів. Однак нині він використовується в основному для простої геометрії, оскільки зберігає лише найнеобхідніше – інформацію про сітку, анімацію об’єкта, розташування камери, освітлення, матеріал, колір;
- gLTF/GLB: нейтральний формат із відкритим вихідним кодом. Це перший дійсно чітко визначений стандарт для 3D. Цей формат файлу підтримує геометрію, матеріали, текстури, кольори та анімацію. Це включає

в себе PBR (Physical Based Rendering), тому тіні та світло будуть виглядати більш реалістичними. gLTF заснований на JSON, тому він зберігає деякі дані у зовнішніх файлах, таких як текстури (JPEG або PNG), шейдери (GLSL) або дані геометрії та анімації (BIN);

- USDZ/USD: формат, розроблений компаніями Apple і Pixar. Це пропрієтарний формат 3D-файлу, який в основному використовується для доповненої реальності на пристроях iOS. Цей формат файлу є найпопулярнішим для 3D-комерції, оскільки за допомогою цього формату можна розміщувати та приміряти 3D-моделі на пристроях iPhone;

- STEP (.STP): формат 3D-файлів, який використовується для машинобудування та оборонної промисловості. Це нейтральний формат 3D-файлів, який може зберігати всю геометрію, включаючи топологію, типи матеріалів, текстури та інші складні дані про об’єкт;

- COLLADE: це нейтральний формат 3D-файлів, який зберігає геометрію, зовнішній вигляд, сцену та анімацію. Це також один з небагатьох форматів, який підтримує фізику та кінематику. COLLADE з часом став менш популярним через його нездатність йти в ногу з новими технологіями.

3.2 Використання 3D-моделей

3D-моделювання використовується в різних галузях, таких як кіно, анімація, ігри, дизайн інтер’єру та архітектура. Вона також використовуються в медичній промисловості для інтерактивного уявлення про анатомію. Велика кількість програмного забезпечення також використовується для побудови цифрового представлення механічних моделей або деталей до того, як вони будуть фактично виготовлені. У таких галузях використовується програмне забезпечення, пов’язане з CAD/CAM, і за допомогою цього програмного забезпечення можна не тільки конструювати деталі, але й зібрати їх та спостерігати за їх функціональністю. 3D-моделювання також використовується в галузі промислового дизайну,

коли продукти моделюються перед тим, як представляти їх клієнтам. У індустрії медіа 3D-моделювання використовується в дизайні сцени/декорації. Також, можна використовувати 3D-моделювання в програмуванні.

Технологія цифрового виготовлення, яку також називають 3D-друком або адитивним виробництвом, створює фізичні об’єкти з геометричного зображення шляхом послідовного додавання матеріалів. Технологія 3D-друку – це технологія, яка швидко розвивається. На сьогоднішній день у світі широко використовується 3D друк для масової кастомізації, виробництва будь-яких типів проектів з відкритим кодом у сфері сільського господарства, в охороні здоров'я, автомобільній промисловості, локомотивній промисловості та авіаційній промисловості [7]. Технологія 3D-друку дозволяє друкувати об’єкт шар за шаром нанесенням матеріалу безпосередньо з моделі автоматизованого проектування (CAD).

3.3 3D-модель як стегоконтейнер

У зв’язку з масовим розвитком обміну 3D-моделями в інтернеті останнім часом особливу увагу приділено передачі та захисту таких даних.

В останні роки графічні 3D-моделі стали більш доступними для загальних кінцевих користувачів завдяки використанню вдосконалених пристроїв сканування та мови моделювання віртуальної реальності (VRML) для графічного опису. Більше того, у зв’язку зі стрімким зростанням інтернету та розвитком технологій проектування та обробки цифрового контенту, багато цінних матеріалів можуть бути представлені в цифрових формах для демонстрації та доступу через інтернет. Через особливості легкого копіювання та модифікації цифрового вмісту, необхідно розробити різноманітні методи цифрового підпису або цифрових водяних знаків для різних цілей захисту, таких як автентифікація моделі та заява про право власності. Цифрові підписи призначені для одержувача електронних документів для перевірки особи відправника та перевірки оригінальності

документів. Схеми цифрових водяних знаків зазвичай розроблені для того, щоб відправник перевірив право власності на авторські права (надійний водяний знак) або для одержувача, щоб перевірити автентифікацію отриманого носія (крихкий водяний знак). Основна відмінність між методами цифрового підпису та водяних знаків полягає в тому, що перша додає невелику частину інформації (цифровий підпис), передану з оригінальними документами, тоді як друга вбудовує невидиму інформацію (водяні знаки) в оригінальний носій.

3.4 Огляд існуючих методів вбудови цифрових водяних знаків у 3D-модель

Зі збільшенням можливостей сканування, обробки та візуалізації 3D-даних захист тривимірних моделей привертає все більше уваги. Однак для тривимірних моделей існує небагато методів вставки водяних знаків, на відміну від відносно зрілої теорії та практики вставки водяних знаків у зображеннями, аудіо та відео. Ця ситуація в основному викликана труднощами, які виникають під час роботи з довільною топологією та нерегулярною вибіркою тривимірних сіток, а також складними можливими геометричними та топологічними атаками на моделі з водяними знаками.

Існуючі методи, що стосуються тривимірних моделей, можна розділити на дві основні категорії, залежно від того, чи цифровий водяний знак вбудований у просторову область (шляхом зміни геометрії чи зв’язку) чи в спектральну область (шляхом зміни певних спектральних коефіцієнтів). Більшість ранніх методів належать до першої категорії, автори намагалися вставити ЦВЗ в різні просторові примітиви. Ці примітиви включають відстань від вершини до центру мас моделі [8], середній напрямок нормалей групи граней, проекцію вершини на її протилежний край, відношення між висотою трикутника і довжини протилежного ребра, локальна щільність тріангуляції, взаємне розташування вершини до її сусідів [9] тощо. Зазвичай

такі методи чутливі до атак з’єднання, таких як спрощення та перебудова. Щоб подолати цей недолік, деякі алгоритми пропонують етап повторної дискретизації на вхідній сітці вилучення водяного знаку, щоб відновити ту саму зв’язність, що й вихідна сітка, але це неминуче робить метод не сліпим. Друга категорія алгоритмів спочатку розкладає сітку в спектральній області перетворення, а потім цифровий водяний знак вставляється в низькій, середній чи високочастотній частині [10]. Ці методи зазвичай забезпечують кращу непомітність і кращу стійкість до геометричних атак, тоді як проблема з’єднання все ще залишається актуальною. Наприклад: лапласівський спектральний аналіз тривимірних сіток чутливий до зміни зв'язності; техніка децимації граней для встановлення представлення також залежить від зв’язності геометричної сітки; більшість інструментів вейвлет-аналізу вимагають на вхід сітку, в якій більшість вершин мають валентність рівною шести.

4 МЕТОД НАНЕСЕННЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКУ НА ТРИВИМІРНИЙ ОБ'ЄКТ

У даній роботі було представлено метод стиснення та вставки цифрового водяного знаку в геометричну модель у спектральну область геометричної сітки. Було розроблено програмну реалізацію, а також проаналізовано результати.

4.1 Геометричний розклад та спектр сітки

У цьому розділі визначається, як отримати спектр геометричної сітки за допомогою оператора Лапласа. Це можна розглядати як еквівалент ДКП для сіток. Таубін вперше заснував цю дослідницьку лінію, коли досліджував фільтрацію [11]. Спочатку потрібно визначити локальний оператор Лапласа для сіток, а потім вивести рівняння перетворення.

4.1.1 Геометричний розклад

Таубін запропонував оригінальний спосіб застосування аналізу Фур'є до дискретних тривимірних сіток за допомогою оператора Лапласа, а Карні та Готсман адаптували його для цілей стиснення. Сітка складається з множини V з N вершин ($|V| = N$) і множини $E \subset V^2$ ребер. Множина V зберігає геометричну інформацію про сітку, тоді як множина E зберігає інформацію про зв'язність сітки. Геометрія задається в декартових координатах $V = (X, Y, Z) \subset R^3$. Іншими словами, геометрія визначається як три вектори розмірності $N(X, Y, Z)$, визначених у кожному вузлі графу сітки. Також, можливо додати атрибути для вершин або граней, такі як колір, прозорість, текстура тощо. Але в даній роботі зосереджується увага на

геометричних даних, визначених графом зв'язності сітки, і не звертається увага на атрибути, однак, атрибути для вершин також можуть бути розкладені за допомогою цієї декомпозиції.

Обозначимо $\{i^*\}$ вершини $v_i \in V$ як:

$$\forall v_j \in V,\ j \in \{i^*\} \Leftrightarrow (v_i, v_j) \in E \tag{4.1}$$

А також, позначаючи d_i ступінь вершини графу v_i (тобто: $d_i = \left|\{i^*\}\right|$), Таубін визначає матрицю Лапласа сітки як матрицю L розміром $N \times N$:

$$L_{ij} = \begin{cases} 1 & ,i = j \\ -d_i^{-1} & ,j \in \{i^*\}, d_i \neq 0 \\ 0 & otherwise \end{cases} \tag{4.2}$$

Власні вектори L утворюють ортогональний базис R^N, тому власні значення можна розглядати як псевдочастоти геометрії, визначеної над графом сітки [12]. Ці власні значення $e_i, 0 \le i \le N-1$, обмежені 0 і 2. Перетворені вектори геометрії виходять проектуванням трьох векторів X, Y, Z на базисні функції. Власні значення впорядковані за зменшенням величини.

Таким чином отримуємо таку систему проекції:

$$\begin{pmatrix} e_0 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & e_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & e_{N-1} \end{pmatrix} = B^{-1}LB \tag{4.3}$$

Стовпці B є базовими функціями (власними векторами L), а стовпці

B^{-1} – подвійними базовими функціями. В результаті перетворення, застосованого до V, отримуємо три вектори розмірності N, які називаються спектрами або псевдочастотами (P, Q, R):

$$\begin{cases} P = BX \\ Q = BY \\ R = BZ \end{cases} \tag{4.4}$$

Точна реконструкція виконується:

$$\begin{cases} X = B^{-1}P \\ Y = B^{-1}Q \\ Z = B^{-1}R \end{cases} \tag{4.5}$$

Але також можна фільтрувати V, використовуючи неповну матрицю. Це може мати місце при розгляді стиснення геометрії з втратами.

4.1.2 Спектр сітки

Сума степеней сигналу на трьох псевдочастотних осях визначає степінь спектру Лапласа V (коефіцієнти, відсортовані за власними значеннями):

$$S_i = \|P_i\|^2 + \|Q_i\|^2 \|R_i\|^2, \; 0 \le i \le N-1 \tag{4.6}$$

На рисунку 4.1 представлений спектр для виділеної частини моделі кролика.

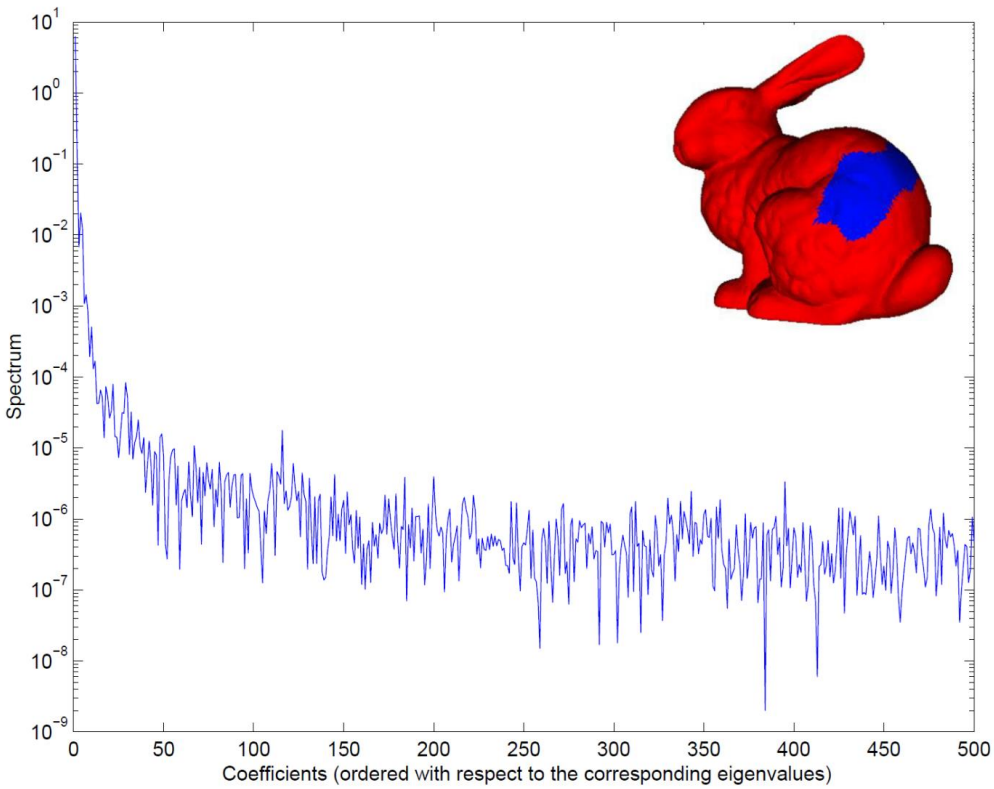

Рисунок 4.1 – Спектр (логарифмічна шкала) темно-сірої плями для моделі зайця

4.2 Алгоритм вставки цифрового водяного знаку

Схема вбудови водяних знаків є заміщуваною. В даній роботі прагнеться вставити 64-бітний цифровий підпис, як це часто вимагається у багатьох програмах. Використовуємо повторення як основну схему кодування каналу. Ця схема стає приватною за допомогою використання секретного ключа, який шифрує біти цифрових водяних знаків по всьому спектру сітки [13].

На рисунку 4.2 ліворуч можна побачити модель кролика з 100-вершинними частинами. Водяний знак дуже помітний. Праворуч: кролик з 500-вершинними частинами. 64-бітний водяний знак не може бути відрізнений.

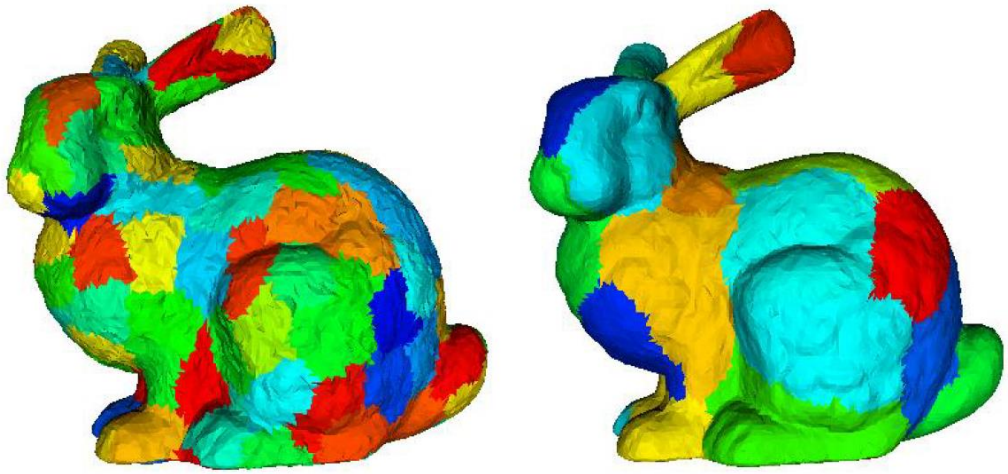

Рисунок 4.2 – Вставка водяного знаку з однаковим маркером інтенсивності, але поділеною на різну кількість частин

Розмір із 500 вершин для стиснення також добре пристосований до цілей вставки ЦВЗ.

4.2.1 Метод спектрального розкладу

Для того, щоб вставити цифровий водяний знак методом спектрального розкладу, випадковим чином змінюється взаємозв'язок між векторами P,Q,R. Маркер інтенсивності при процесі вставки водяного знаку визначається як індекс i_0, на якому починаємо вставляти біти у спектр. Таким чином, залишаємо низькі частоти незмінними, щоб забезпечити кращу невидимість ЦВЗ. Кожна сума $S_i, i \geq i_0$, потенційно зберігає один біт ЦВЗ. Повторюємо водяний знак якомога більше разів в межах коефіцієнтів $(N - i_0)$, що залишилися. Крім того, для того, щоб не придавати великого значення першим бітам цифрового водяного знаку під час процесу пошуку, шифруємо його за допомогою секретного ключа [14]. Орієнтуючись на один

набір з трьох коефіцієнтів (P_i, Q_i, R_i), сортуємо їх $(P_i, Q_i, R_i) \to (C_{\min}, C_{inter}, C_{\max})$ разом з:

$$\begin{cases} C_{\min} = \min(P_i, Q_i, R_i) \\ C_{\max} = \max(P_i, Q_i, R_i) \\ C_{inter} \text{ є спектральним коефіцієнтом, що залишився} \end{cases} \quad (4.7)$$

Інтервал $[C_{\min}; C_{\max}]$ довжини $\triangle = C_{\max} - C_{\min}$ ділиться на два рівні інтервали: $W_0 = [C_{\min}; C_{\max} + \triangle/2]$ та $W_1 = [C_{\min} + \triangle/2; C_{\max}]$.

4.2.2 Процес вставки цифрового водяного знаку

Якщо біт, який буде вставлений у i-тий набір коефіцієнтів дорівнює «0» й C_{inter} знаходиться в підмножині W_0, не змінюємо трійку векторів. Навпаки, якщо C_{inter} лежить у підмножині W_1, інвертуємо його відносно центру інтервалу $[C_{\min}, C_{\max}]$ (рисунок 4.3).

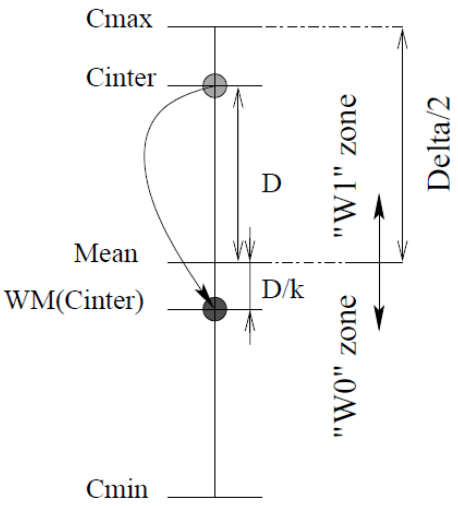

Рисунок 4.3 – Процес введення ЦВЗ у разі вставки «0»

Відстань перевороту поділяється на коефіцієнт $k = 10$, обраний для

забезпечення візуальної невидимості. Процедура точно симетрична у випадку введення «1». Таким чином, геометрична модифікація (якщо така є) випадковим чином розподіляється між трьома наборами псевдо-частот. Значення C_{inter} повинно бути перевернуто під межою. Схема аналогічна у випадку вставки «1».

4.2.3 Процес відтворення цифрового водяного знаку

Щоб відтворити цифровий водяний знак, просто зчитаємо чи знаходиться C_{inter} у діапазоні W_0 або W_1. Дескремблування можливе лише для власника секретного ключа. Як було сказано раніше, водяний знак повторюється кілька разів для підвищення надійності. Використовуємо просту стратегію голосування більшістю, щоб визначити значення вбудованого біта з наборів усіх його вбудованих реплік. Вийшло, що повторення є простою, але ефективною стратегією для кодування каналу для вставки водяних знаків, а перемішування ЦВЗ різко збільшує продуктивність етапу відтворення. Без перемішування, тільки перші біти були відтворенні, тоді як у випадку перемішування, всі біти мають однакову ймовірність відтворення.

В результаті, було побудовано графік залежності візуального спотворення від маркеру інтенсивності водяного знаку (рисунок 4.4). Для моделі кролика було виявлено, що водяний знак був невидимим на рівні деградації $D_{vis} = 2.5 \times 10^{-6}$. Поріг спотворення не такий саме, як для стиснення, оскільки процес вставки не спотворює сітку таким же чином. По-друге, було вимірювано відтворення цифрового водяного знаку при стисненні моделі, щоб продемонструвати надійність водяних знаків проти квантування спектральних коефіцієнтів.

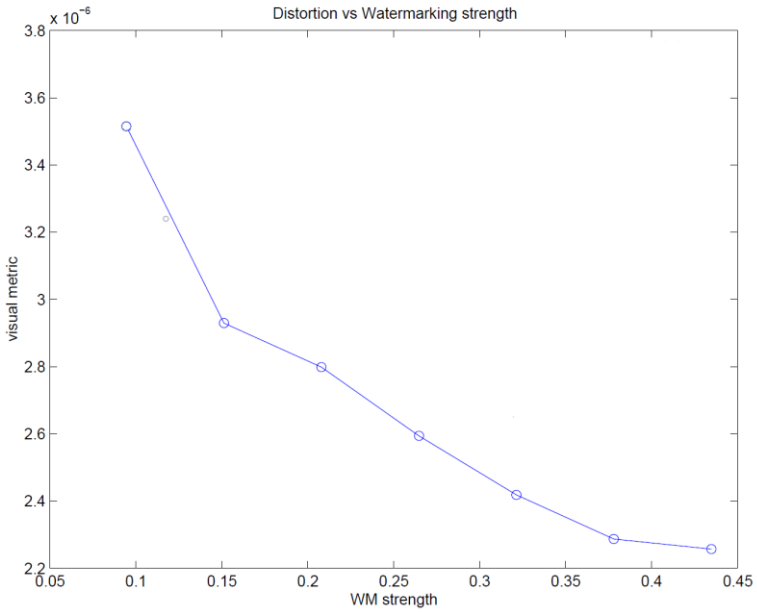

Рисунок 4.4 – Графік візуального спотворення від маркеру інтенсивності

Із графіку видно, що із збільшенням маркеру інтенсивності, збільшується візуальне спотворення моделі.

4.3 Стійкість ЦВЗ до спектрального стиснення

В якості атаки на вбудований ЦВЗ, спектральні дані стискаються методом, який був описаний раніше. Інтенсивність стиснення зображується як кількість бітів квантування. Як видно на рисунку 4.5, цифровий водяний знак, вбудований у модель кролика, повністю відновлюється після квантування 14 біт, що є нижньою межею для цілей стиснення. Було показано, що перекриття має особливе значення під час вбудовування ЦВЗ, хоча воно не впливає на відтворення.

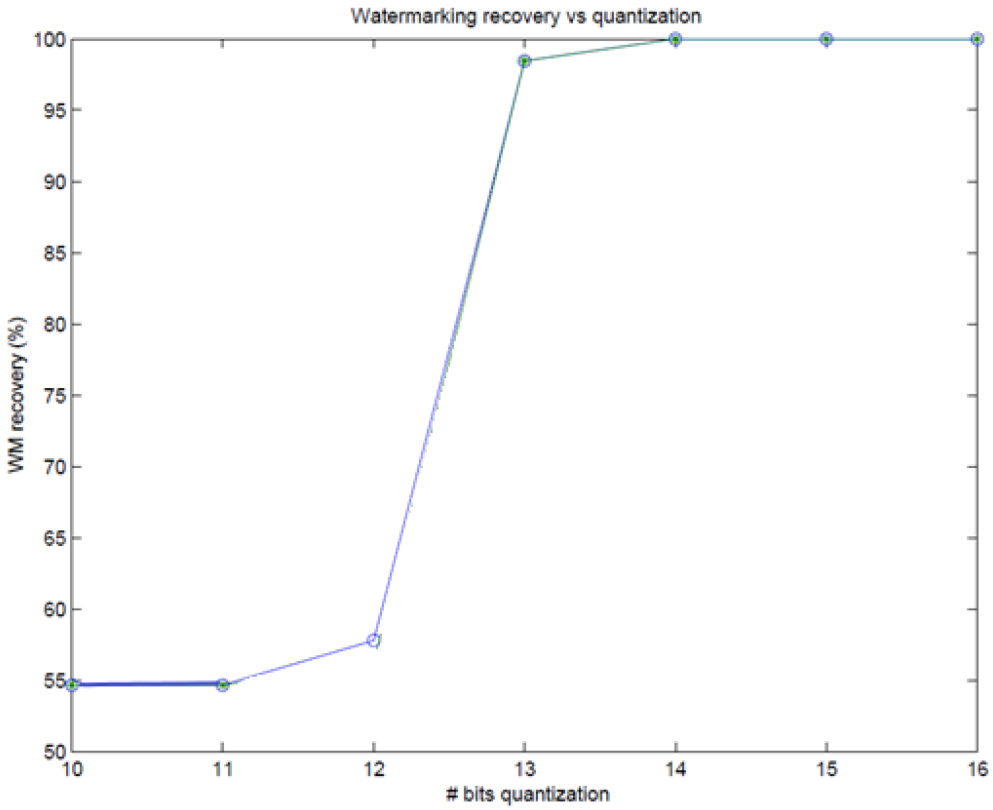

Рисунок 4.5 – Результати декодування ЦВЗ для атаки стисненням на моделі кролика

Ця схема стійка до класичного квантування для спектральних коефіцієнтів (для квантування від 14 біт).

4.4 Метрики для оцінки візуального спотворення

Для того, щоб оцінити метод вставки шляхом спектрального розкладу, необхідно визначити математичні метрики для оцінки результатів. Спочатку, потрібно визначити геометричний сенс Лапласіана для кожної вершини. Цей Лапласіан повинен буде штрафувати збільшення відстані вершини до центру її сусідів, тобто барицентричне положення. Нехай l_{ij} – відстань від вершини v_i до вершини v_j. Геометричний локальний Лапласіан визначається як:

$$GL(v_i) = v_i - \frac{\sum_{j \in \{i^*\}} l_{ij}^{-1} v_j}{\sum_{j \in \{i^*\}} l_{ij}^{-1}} \tag{4.8}$$

Локальний Лапласіан представляє локальну гладкість сітки у вершині v_i. Візуальна відстань між сіткою M та іншою сіткою M' з такою ж зв'язністю повинна поєднувати глобальні відстані між вершинами та відстані між локальними Лапласіанами. Така візуальна відстань D_{vis} потім враховує необроблену інформацію, подібну до пікового співвідношення сигналу до шуму (PSNR) і різницю локальної гладкості:

$$D_{vis}(M, M') = \frac{1}{2|V|}\left(\sum_{i=0}^{N-1} \|v_i - v'_i\| + \sum_{i=0}^{N-1} \|GL(v_i) - GL(v'_i)\|\right) \tag{4.9}$$

Ця метрика має перевагу в тому, щоб розрізняти випадкове додавання шуму на вершинах і погану якість реконструкції. Наприклад, око краще пристосовується до гладкого об'єкта, а не до шумової сітки. Показник збільшується у разі локальних порушень, що зазвичай призводить до поганої якості для людського ока. На жаль, порядок величини візуальної метрики залежить від сітки. Налаштування допустимого стиснення або спотворення водяних знаків проводиться вручну для кожної моделі.

4.5 Програмна реалізація

Для розробки програми була обрана мова програмування Python, оскільки це потужна і водночас проста мова, яка має набір корисних бібліотек. Для програмної реалізації було використано наступні бібліотеки:

- Numpy: розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами;

- SciPy: відкрита бібліотека високоякісних наукових інструментів, що містить модулі для оптимізації, інтегрування, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів, розв'язування звичайних диференціальних рівнянь та інших задач, які розв'язуються в науці і при інженерній розробці;

- PyMesh – це швидка платформа прототипів, орієнтована на обробку геометрії. Вона надає набір загальних функцій обробки сітки та інтерфейсів з низкою найсучасніших пакетів із відкритим кодом, щоб безперешкодно об’єднати їх потужність в єдиному середовищі розробки. PyMesh надає легко читаний мінімалістичний інтерфейс та працює з нативними структурами даних Python.

Для вставки ЦВЗ була реалізована функція insert, яка представлена в лістингу 4.1.

Лістинг 4.1 – Алгоритм вставки ЦВЗ

```
# Scramble the data
scrambled_data = scramble(data, secret)

mesh = pymesh.load_mesh(filename_in)

if partitions == -1:
    partitions = mesh.num_vertices/500

# Partition the mesh into patches
patches, mapping = partitioning.mesh_partitioning(filename_in,
mesh, partitions)
```

Спочатку мішаємо вхідні дані за допомогою секретного ключа, потім завантажуємо модель за допомогою бібліотеки PyMesh і ділимо її на частини за допомогою функції mesh_partitioning, основна частина якого представлена в лістингу 4.2.

Лістинг 4.2 – Алгоритм розділу моделі на частини

```
for i in range(partitions):
        faces = []
        vertices = []
        # Initialize all the mappings at -1
        indexes_mapping = np.zeros(mesh.num_vertices) - 1

        for j, partition in enumerate(partitions_list):

            # Find all the faces that should be added to this
specific patch
            if int(partition) == i:

                # Check if the vertices corresponding to this
face have already been added to this parttion
                i1 = indexes_mapping[mesh.faces[j][0]]
                i2 = indexes_mapping[mesh.faces[j][1]]
                i3 = indexes_mapping[mesh.faces[j][2]]

                # If the vertices have not been added to this
partition, add them and update the mapping
                if i1 == -1 :

vertices.append(mesh.vertices[mesh.faces[j][0]])
                    i1 = len(vertices) - 1
                    indexes_mapping[mesh.faces[j][0]] = i1
                if i2 == -1 :

vertices.append(mesh.vertices[mesh.faces[j][1]])
                    i2 = len(vertices) - 1
                    indexes_mapping[mesh.faces[j][1]] = i2
                if i3 == -1 :

vertices.append(mesh.vertices[mesh.faces[j][2]])
                    i3 = len(vertices) - 1
                    indexes_mapping[mesh.faces[j][2]] = i3
                # Add the faces to the list of faces for this
partition with the correct indexes
                faces.append([i1, i2, i3])

        # Save the patch and the mapping
        mapping.append(indexes_mapping)
        patches.append(pymesh.form_mesh(np.array(vertices),
np.array(faces)))
```

Перед безпосередньою вставкою ЦВЗ у частину моделі, необхідно обчислити власні вектори B за допомогою функції compute_eigenvectors, що представлена в лістингу 4.3.

Лістинг 4.3 – Функція підрахунку власних векторів

```
def compute_eigenvectors (num_vertices, patch):

    laplacian = np.zeros((num_vertices, num_vertices))

    # Get the list of neighbors and the valence of each vertice
    star, d = utils.get_neighbors_and_valence(num_vertices,
patch)

    # Compute the Laplacian
    for i in range(num_vertices):
        laplacian[i][i] = 1
        for j in star[i]:
            laplacian[i][int(j)] = -1.0/d[i]

    # Get the eigenvectors and eigenvalues of the laplacian
    eigenValues, eigenVectors = np.linalg.eig(laplacian)

    # Sort the eigenvectors regarding to their eigenvalues
    idx = eigenValues.argsort()
    eigenValues = eigenValues[idx]
    eigenVectors = eigenVectors[:,idx]

    return np.transpose(eigenVectors)
```

Після цього, можна отримати спектральні коефіцієнти P,Q,R псевдочастот шляхом множення власних векторів на вектори X,Y,Z даної сітки (лістинг 4.4).

Лістинг 4.4 – Підрахунок спектральних коефіцієнтів

```
B = compute_eigenvectors(patch.num_vertices, patch.faces)
# Get the spectral coefficients
P = np.matmul(B, patch.vertices[:, 0])
Q = np.matmul(B, patch.vertices[:, 1])
R = np.matmul(B, patch.vertices[:, 2])
```

Далі, формується степінь спектру Лапласа S як сума квадратів коефіцієнтів псевдочастот. Також, рахується квантиль, який являє собою процентиль даних уздовж степеню спектру Лапласа S. Таким чином, частоти сітки більші за підрахований квантиль не модифікуються. Коефіцієнти

спектру відсортовані за власними значеннями. Модифікація частоти спектру представлена в лістингу 4.5. В кожній частоті вставляється один біт цифрового підпису.

Лістинг 4.5 – Модифікація частоти спектру зі вставкою одного біту ЦВЗ

```
minimum, intermediate, maximum = sorted([[P[i], 0], [Q[i], 1],
[R[i], 2]], key=get_key)

# Order P, Q, R -- spectra
C_min = minimum[0]
C_inter = intermediate[0]
C_max = maximum[0]

# Remember to which of P, Q and R corresponds C_min, C_inter and
C_max
min_index = minimum[1]
inter_index = intermediate[1]
max_index = maximum[1]

Mean = 0.5*C_max + 0.5*C_min

# We want to use 4 intervals
if C_inter < Mean :
    Min = C_min
    Max = Mean
else:
    Min = Mean
    Max = C_max

Mean = (Min + Max)/2

# Change C_inter from interval if needed
if data[count%len(data)]:
    if C_inter < Mean :
        C_inter = Mean + (Mean - C_inter)/k
else:
    if C_inter > Mean :
        C_inter = Mean - (C_inter - Mean)/k

# Reassign P, Q and R to their new value
values = np.zeros(3)
values[min_index] = C_min
values[inter_index] = C_inter
values[max_index] = C_max

[P[i], Q[i], R[i]] = values
```

Отримавши спектральні коефіцієнти, можна порахувати нові координати вершин, після чого різні частини об’єднуються в одну сітку.

Для відновлення ЦВЗ, знову обчислюється максимальна частота, з якої повинні зчитуватися дані. Всі інші частоти не слід враховувати, для них додаються 0.5 до списку даних, щоб вона не вплинула на отримані дані, але порядок залишився правильним. Інші частоти зберігають біт даних, який треба зчитати. Функція зчитування представлена в лістингу 4.6.

Лістинг 4.6 – Алгоритм зчитування одного біту ЦВЗ із частоти

```
C_min, C_inter, C_max = sorted([P[i], Q[i], R[i]])

Mean = 0.5*C_max + 0.5*C_min

# We use 4 intervals, so we want to know in which half C_inter
is located
if C_inter < Mean :
    Min = C_min
    Max = Mean
else:
    Min = Mean
    Max = C_max

Mean = (Min + Max)/2

# Read the data bit
if C_inter >= Mean:
    data.append(1)
else:
    data.append(0)
```

4.6 Атаки

Атаки виконуються шляхом випадкового додавання шуму або згладжування на модель, змінюючи амплітуду або кількість ітерацій і маркер інтенсивності [1]. Модель повинна бути спочатку позначена водяним знаком за допомогою функції generate_sample.

Для додавання шуму, написана функція noise (лістинг 4.7). Вона повертає копію моделі, до якої був доданий шум до всіх координат

пропорційно значенню амплітуди. Для шуму використовуються наступні значення амплітуд: 0.01, 0.02, 0.03, 0.05, 0.07, 0.1.

Лістинг 4.7 – Функція додавання шуму на модель

```
def noise(mesh, amplitude):
    new_vertices = []

    # Go through each vertice and modify each of its 3
coordinates relatively to @amplitude
    for vertice in mesh.vertices:
        new_vertices.append([vertice[0] + vertice[0] * random()
* amplitude, vertice[1] + vertice[1] * random() * amplitude,
vertice[2] + vertice[2] * random() * amplitude])

    return pymesh.form_mesh(numpy.array(new_vertices),
mesh.faces)
```

Для згладжування моделі використовується функція smoothing (лістинг 4.8).

Лістинг 4.8 – Функція згладжування моделі

```
def smoothing(mesh, iterations):
    neighbors, _ =
utils.get_neighbors_and_valence(mesh.num_vertices, mesh.faces)
    vertices = mesh.vertices

    # At each iteration, move each vertice to the center of its
neighbors
    for _ in range(iterations):
        for i in range(mesh.num_vertices):
            new_value = compute_center(neighbors[i], vertices)
            vertices[i] = new_value

    return pymesh.form_mesh(vertices, mesh.faces)
```

На кожній ітерації кожна вершина переміщуються в центр її сусідніх вершин.

4.7 Тестування програмного забезпечення

Для вставки ЦВЗ була обрана модель кролика (рисунок 4.6).

Рисунок 4.6 – Модель кролика, яка використовується для вставки ЦВЗ

Дана модель складається з 14007 вершин і 27930 трикутників.

4.7.1 Вставка цифрового водяного знаку в модель

Цифровий водяник знак для вставки в модель – це 64-бітне число. Вставка ЦВЗ у модель була проведена з різними значеннями кількості частин моделі та маркером інтенсивності. Модель була поділена на 100, 90, 80, 70, 60, 50, 40, 30 частин, для кожної ітерації була обраний маркер інтенсивності 1, 2, 3, 5, 7, 10, 20, 30, 50, 70, 100. Таким чином, було створено багато моделей зі вставленою ЦВЗ з різними параметрами. На кожну модель були проведені атаки двох видів: шум та згладжування. Атаки шумом були проведені з різними значенням амлітуди: 0.01, 0.02, 0.03, 0.05, 0.07, 0.1, а атаки згладжуванням були проведені з різною кількістю ітерацій: 1, 2, 3, 5, 7, 10.

В процесі роботи програми, був засічений час, за який вставляється й

відтворюється ЦВЗ для кожної моделі, а також було пораховано кількість помилок. Також, для вихідної моделі та моделі зі вставленою ЦВЗ була порахована середнє квадратичне відхилення, відстань Гаусдорфа та локальне згладжування (local smoothness).

Пропонується проаналізувати модель з різним маркером інтенсивністю, що поділена на 60 частин. У наведених нижче графіках по осі X представлено маркер інтенсивності, по осі Y – дані, що аналізуються.

По-перше в залежності від інтенсивності, змінюється місткість даних, що може бути вставлено, тобто кількість бітів ЦВЗ. Чим більше значення інтенсивності, тим більше місткість даних. Наприклад, для маркеру 1, можливо вставити лише 410 бітів даних, у той час як для маркеру інтенсивності 100 можливо вставити 38407 бітів даних ЦВЗ (рисунок 4.7). Отже, будь-який маркер інтенсивність зможе зберігти 64-бітне число.

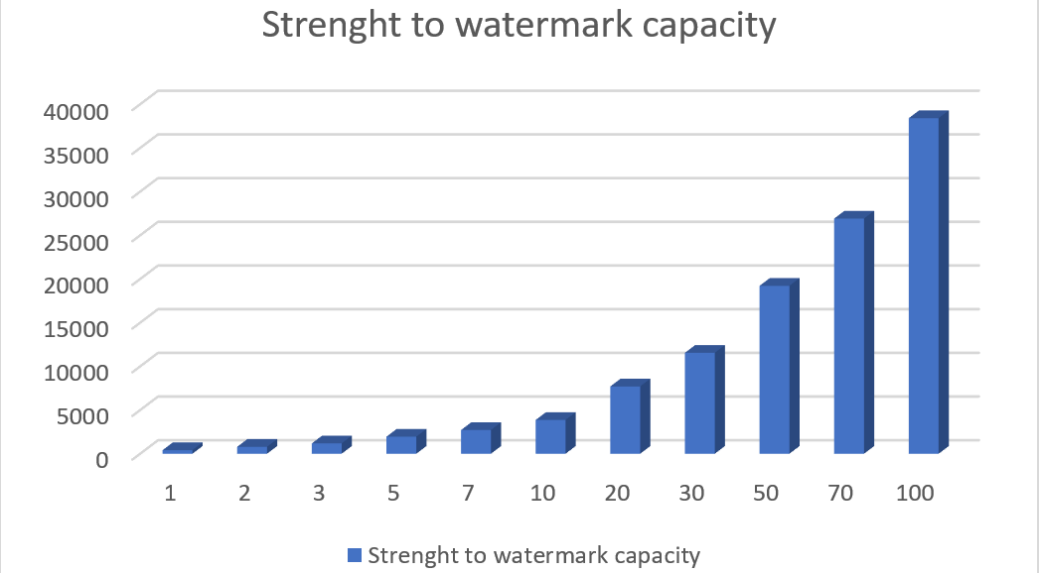

Рисунок 4.7 – Графік залежності місткості даних від маркеру інтенсивності

Час вставки й відтворення ЦВЗ приблизно однаковий. Для вставки це в середньому 89 мс, для відтворення – в середньому 25 мс, але для інтенсивності 1 час сильно відрізняється – 319 мс та 260 мс відповідно (рисунок 4.8).

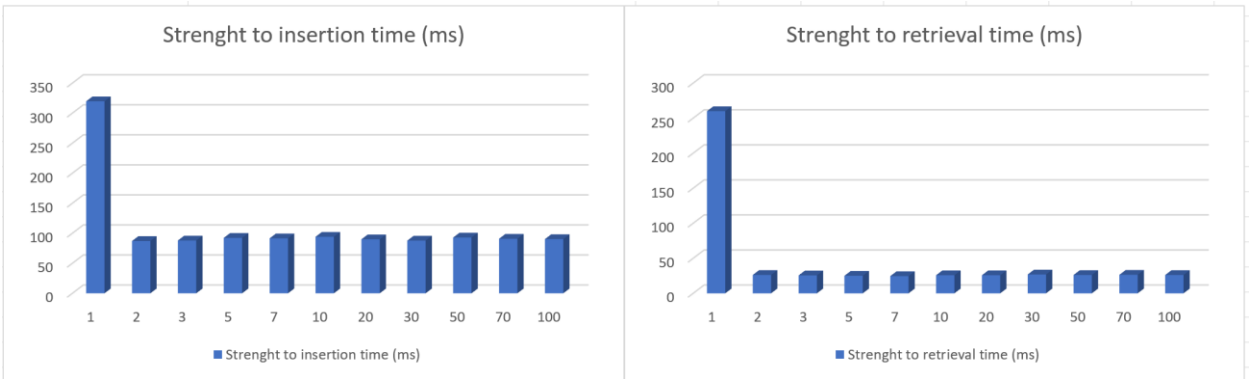

Рисунок 4.8 – Час вставки й відтворення ЦВЗ в залежності від інтенсивності

З ростом інтенсивності, а отже й місткості, модель все більше змінюється візуально, а кількість помилок, зроблених під час відтворення ЦВЗ зменшується. Тому слід знайти таке оптимальне значення інтенсивності, щоб кількість помилок наближалась до нуля, а візуальне спотворення було мінімальне. Для оцінки візуального спотворення було використано три математичні методи: обчислення середньо-квадратичного відхилення, відстані Гаусдорфа та локальне згладжування між двома моделями (рисунок 4.9).

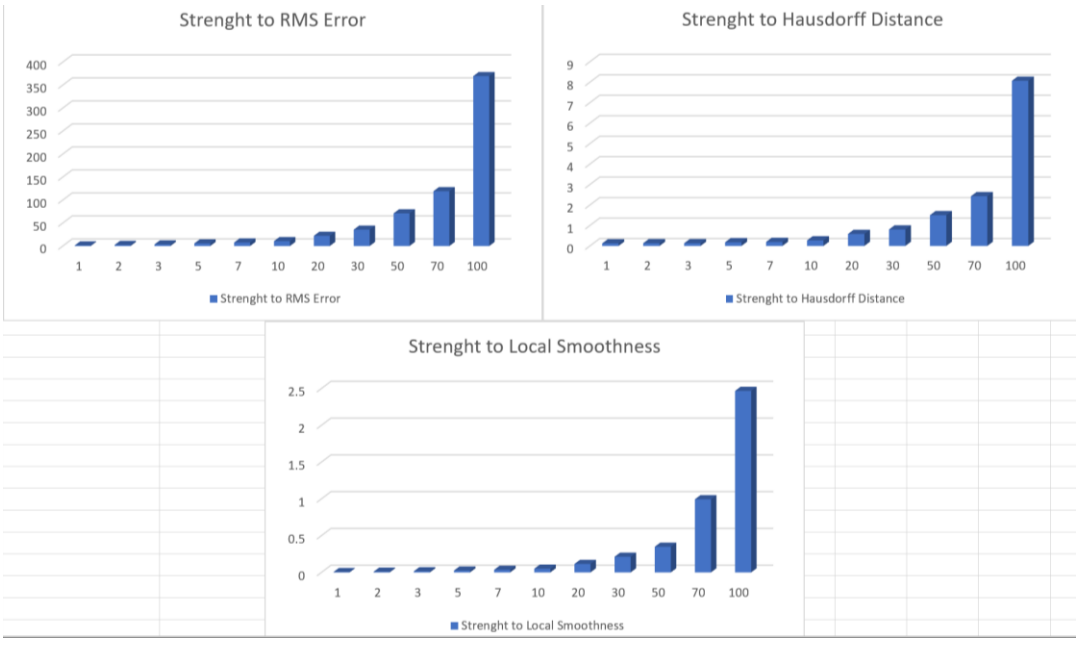

Рисунок 4.9 – Графіки візуального спотворення моделей

Кількість помилок при відтворенні ЦВЗ починає наближатися до нуля при значенні маркеру інтенсивності 20 і більше. Для маркеру 7 та 10 виявляється лише 1 помилка з 64 бітів, тоді як для маркеру 1 кількість помилок сягає 17, що є 26.5% від вихідних даних (рисунок 4.10).

Рисунок 4.10 – Кількість помилок при відтворенні ЦВЗ в залежності від маркеру інтенсивності

4.7.2 Атаки на модель

На моделі із вбудованою ЦВЗ було здійснено атаки шумом та згладжуванням.

Для шуму було обрано амплітудні коефіцієнти 0.01, 0.02, 0.03, 0.05, 0.07, 0.1. Після додавання шуму намагаємось відтворити ЦВЗ, при цьому рахуємо кількість помилок. Результат можна побачити на рисунку 4.11.

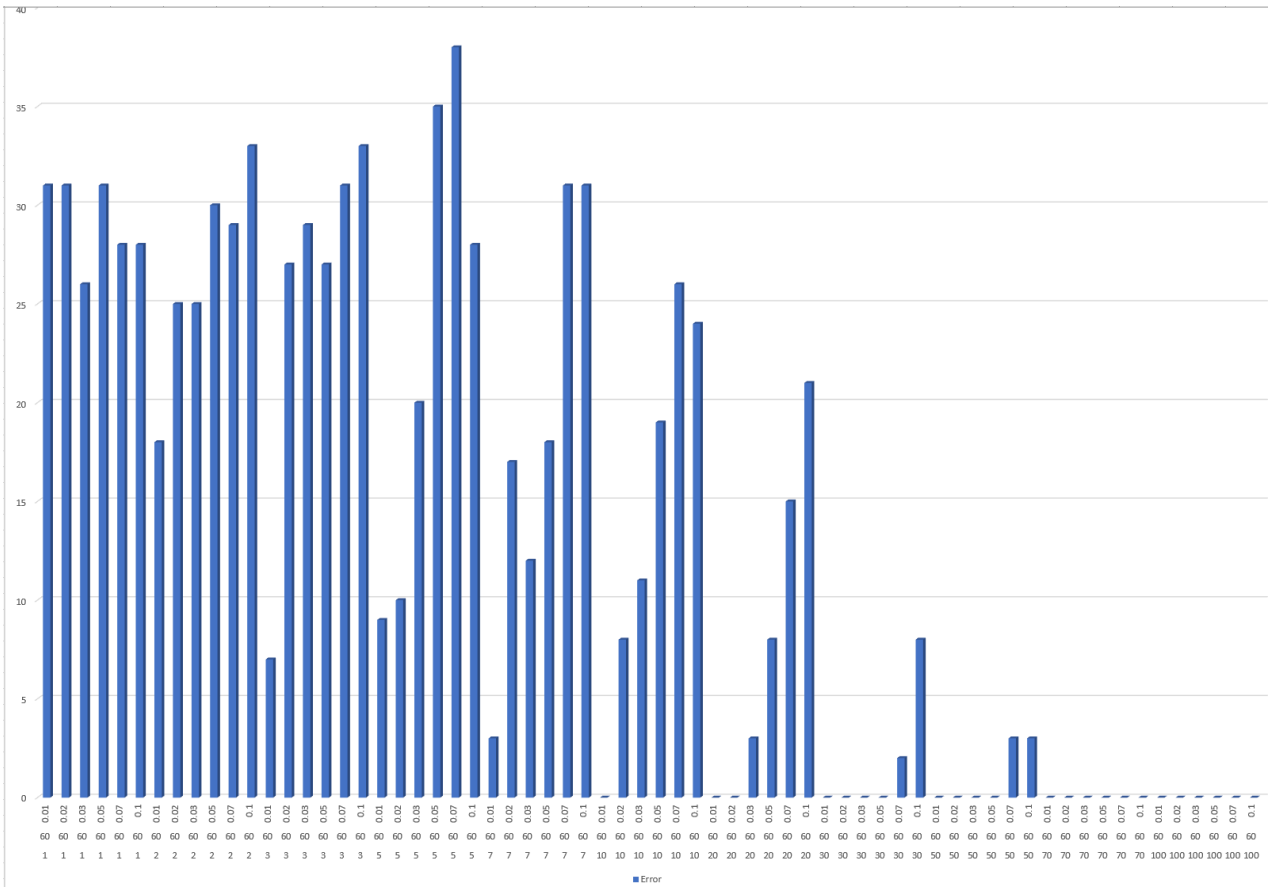

Рисунок 4.11 – Графік помилок в залежності від маркеру інтенсивності та амплітуди шуму для моделі, атакованою шумом

Із графіку видно, що чим більша амплітуда шуму, тим більше помилок здійснюється при відтворенні. Однак, для маркеру інтенсивності 100, кількість помилок сягає нулю для всіх проведених атак шумом. Слід зазначити, що навіть з маркером інтенсивністю 20, вдається відтворити ЦВЗ без помилок із невеликими значеннями амплітуди (0.01, 0.02), що є непоганим результатом, враховуючи невелике візуальне спотворення цієї моделі після вставки цифрового підпису.

Для атак згладжуванням використовується різна кількість ітерацій: 1, 2, 3, 5, 7, 10. Така ж сама методика атак була використана, як і під час додавання шуму: спочатку робимо згладжування, потім намагаємось відтворити ЦВЗ, рахуючи кількість помилок. Результат можна побачити на рисунку 4.12.

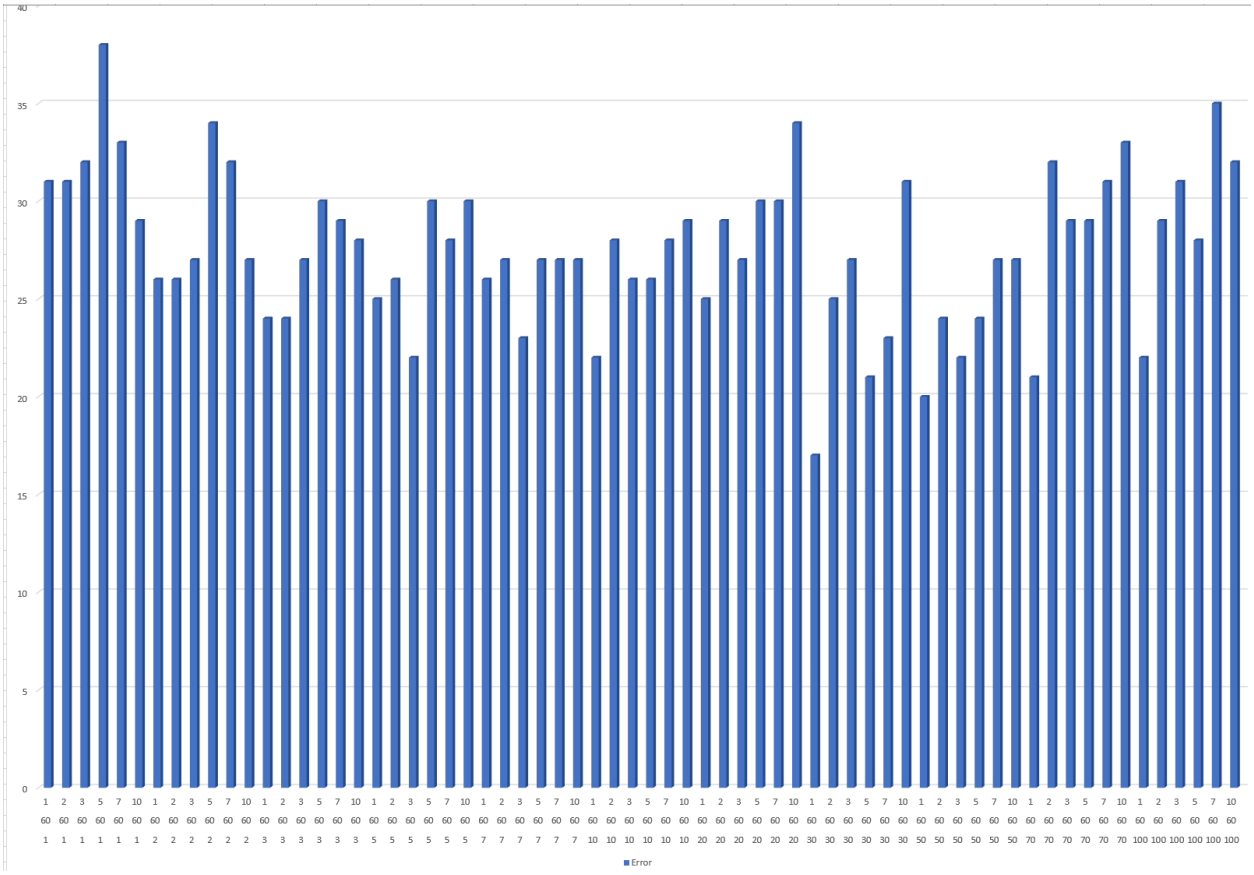

Рисунок 4.12 – Графік помилок в залежності від маркеру інтенсивності та кількості ітерацій для моделі, атакованою згладжуванням

Із графіку видно, що кількість помилок стало набагато більше навіть з однією ітерацією згладжуванням. В середньому кількість помилок сягає 40-45%, що є недопустим навіть зі значенням маркеру інтенсивності 100, що є недопустимим результатом. Можна зробити висновок, що даний метод вставки ЦВЗ є вразливим до атак згладжуванням.

4.8 Аналіз отриманих результатів

Проаналізувавши весь спектр отрманих даних після вставки ЦВЗ методом спектрального розкладу з різними значеннями маркеру інтенсивності й кількістю частин, на які поділяється модель при вставці, можна зробити висновок, що інтенсивність та розподіл моделі відіграє

ключову роль в даному алгоритмі. Із збільшенням маркеру інтенсивності, збільшується надійність відтворення ЦВЗ без помилок, навіть після атак шумом, але в противагу збільшується візуальне спотворення моделі. У результаті, було обрано найбільш вдалу комбінацію параметрів для вставки 64-бітного ключа в модель, яка складається 14007 вершин. Це розділ на 40 частин та маркером інтенсивність 5, що означає, що кожна частина моделі буде складатися з 350 вершин (рисунок 4.8). З цими показниками можна вставити й відтворити ЦВЗ без жодної помилки, водночас з мінімальними візуальними спотвореннями. Однак, для більшої надійності, пропонується взяти модель з маркером інтенсивності 7, тому що кількість помилок може змінюватися при проводені повторного проведення експерименту, а отже відсутність помилок не гарантована. Тим не менш, різниця в кількості помилок буде мінімальною. Також, для кожної моделі необхідно індивідуально обирати кількість, на яку ця модель буде ділитися, щоб кожна частина складалася з 350-500 вершин, оскілька, як було емпірично досліджено, така кількість є найбільш прийнятий для цілей вставки ЦВЗ.

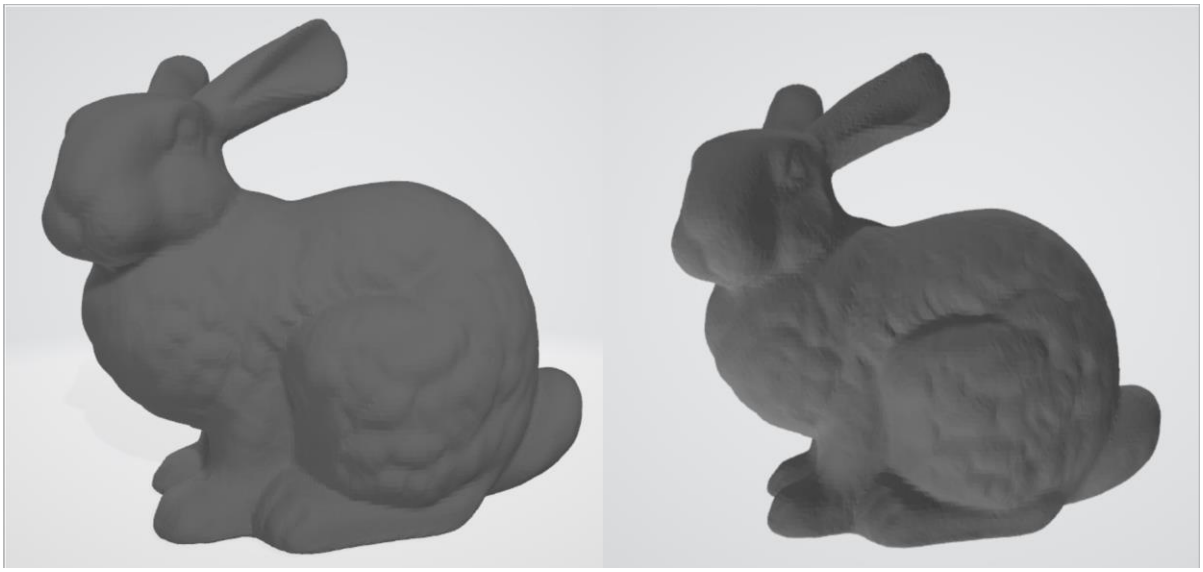

Рисунок 4.13 – Порівняння оригінальної моделі та моделі зі вставленою ЦВЗ з маркером інтенсивності 5 та розподілом на 40 частин

ВИСНОВКИ

Отже, в даній кваліфікаційній роботі були розібрані існуючі методи та механізми цифрової стеганографії, проведено порівняльний аналіз, вироблені критерії для алгоритму, а також розроблений програмний додаток для вбудови ЦВЗ у тривимірну модель методом спектрального розкладу. Крім того, були проаналізовані основні методики атак на модель, щоб виявити стійкість пропонованого методу.

Було представлено спектральний розклад моделі на основі діагоналізації Лапласа, а також схему вбудови ЦВЗ поверх цього представлення. Дана схема цифрових водяних знаків стійка до афінного перетворення, стиснення спектральної геометрії, а також атак шумом, оскільки зміна моделі відбувається в спектральній області. Емпірично було досліджено, що даний метод не є стійким до атак згладжуванням, тому що така модифікація призводить до порушення з’єднання вершин, що може призвести до некорректного з’єднання частин моделі після вставки ЦВЗ у спектр. Розподіл моделі – важливий момент процесу вставки ЦВЗ. Зокрема, не можна зробити жодних припущень щодо топології отриманих частин після розбиття, тому подальше з’єднання частин із зміненим спектром в єдину модель може призвести до візуальних дефектів.

Подальший розвиток цього методу передбачає покращення алгоритму розбиття моделі на частини, що може поліпшити результат відтворення цифрового водяного підпису після атак згладжуванням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Яремчук, Ю. Є., та В. В. Карпінець. Аналіз стійкості стеганографічного перетворення до вбудовування цифрових водяних знаків у зображення [Текст] / Ю. Є. Яремчук. – К. : Інформаційні технології та компютерна інженерія, 2007. №1 – 212-217 с.

2. Абазіна, Є.С., та Єрунов А.О. Цифрова стеганографія: стан і перспективи [Текст] / Є.С. Абазіна. – К. : Системи управління, зв'язки й безпека, 2016. – №2.

3. Evelyn Brannock, Michael Weeks, Robert Harrison. Watermarking with Wavelets: Simplicity Leads to Robustness [Текст] / Brannock Evelyn. – Computer Science Department Georgia State University. – IEEE, 2008. – 587-592 pp.

4. G. Bouridane. A, M. K. Ibrahim. Digital Image Watermarking Using Balanced Multi wavelets [Текст] / Bouridane G. – Transaction on Signal Processing. – IEEE. 2006, 1519-1536 pp.

5. Cox, I.J.; Miller, M.L.; Bloom, J.A. Digital Watermarking [Текст] / I.J. Cox. – Morgan Kaufmann, 2001.

6. Задорожний О.В., Мартовицький В.О. Методи вбудовування цифрових водяних знаків [Текст] / О.В. Задорожний, В.О. Мартовицький. – Харків: Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління, 2022.

7. Beknazarova S.S., Maxammadjonov M.M. 3D modeling and the role of 3d modeling in our life [Текст] / S.S. Beknazarova, M.M. Maxammadjonov. – Computer Science – ISSN 2413-1032, 2016. – pp. 28-31.

8. Z. Yu, H.H.S. Ip, and L.F. Kwok, "A Robust Watermarking Scheme for 3D Triangular Mesh Models" [Текст] / Yu Z., Ip H.H.S., Kwok L.F., Pattern Recognition, vol. 36, no. 11, 2003, pp. 2603-2614.

9. A.G. Bors, "Watermarking Mesh-based Representations of 3-D Objects Using Local Moments" [Текст] / Bors A.G., IEEE Transactions on Image

Processing, vol. 15, no. 3, 2006, pp. 687-701.

10. R. Ohbuchi, A. Mukaiyama, and S. Takahashi, “A Frequency-domain Approach to Watermarking 3D Shapes” [Текст] / Ohbuchi R., Mukaiyama A., Takahashi S., Computer Graphics Forum, vol. 21, no. 3, 2002, pp. 373-382.

11. G. Taubin, T. Zhang, G. Golub, Optimal surface smoothing as filter design [Текст] / Taubin G. Zhang G, Golub, IBM Technical Report RC- 20404.

12. Jing L, Yinghui W, Wenjuan H, Ye L A New Watermarking Method of 3D Mesh Model [Текст] / L. Jing, W. Yinghui, H. Wenjuan, TELKOMNIKA Indonesian Journal of Electrical Engineering. – 2014, pp. 1610-1617.

13. El-seoud SA, Rumman NA, Tajeddin IATF, Hatatneh KF, Gutl C, Robust Digital Watermarking for Compressed 3D Models based on Polygonal Representation [Текст] / SA El-seoud, NA Rumman, IATF Tajeddin, KF Hatatneh, C Gutl. International Journal of Computer Applications, 2013. p. 61-114.

14. Jang H.U., Choi H.Y., Son J et al. Cropping-resilient 3D mesh watermarking based on consistent segmentation and mesh steganalysis [Текст] / H.U. Jang, H.Y. Choi, Son J et al. In: Multimedia tools and applications, 2017, pp 1–28.

15. Bors AG, Luo M. Minimal surface distortion function for optimizing 3D watermarking [Текст] / A. G. Bors, M. Luo. – IEEE Trans Image Process, 2013, pp. 1822–1835.