УДК 004.056.5

Карпінець В.В., Яремчук Ю.Є., Іванішина Д.О.

ВИРІШЕННЯ ПРОБЛЕМИ ЗАХИСТУ АВТОРСЬКИХ ПРАВ ВЕКТОРНИХ ЗОБРАЖЕНЬ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ

У роботі розроблено сервіси та додатки для захисту векторних зображень в інформаційно-комунікаційних системах, що дають можливість підтверджувати авторство без необхідності наявності оригіналу зображення чи самого цифрового водяного знаку (ЦВЗ). При цьому розроблені сервіси та додатки забезпечують достатній рівень стійкості і незначні спотворення при вбудовуванні ЦВЗ. Особливістю розроблених сервісів та додатків є те, що вони складаються з окремих незалежних програмних блоків, які можна використовувати при вбудовуванні та при витягуванні ЦВЗ, що скорочує розмір програмного коду та підвищує надійність їх роботи.

Ключові слова: захист авторських прав, захист векторних зображень, цифрові водяні знаки, підтвердження авторства.

Вступ. На сьогодні в інформаційно-комунікаційних системах все більшого поширення отримують цифрові зображення векторного формату, що використовуються для проектування архітектурних об'єктів, інтер'єрів, розробки приладів, реклами, логотипів, створення шрифтів, географічних карт тощо, на створення яких витрачається багато часу та коштів. Наприклад, зараз дуже поширене використання векторних карт, зокрема існує достатньо Інтернет-сервісів, які надають доступ до деталізованих карт усього світу, крім того, кожен може скористатися системою GPS-навігації за допомогою спеціального пристрою чи мобільного телефону, які також використовують карти у векторному форматі. В зв'язку з цим, виникає проблема, пов'язана з можливістю нелегального копіювання та розповсюдження векторних зображень, які мають свого правовласника [1].

Для вирішення задачі захисту авторських прав цифрових векторних зображень в інформаційно-комунікаційних системах використовуються різні сервіси та додатки, що дають змогу маркувати об'єкти захисту цифровими водяними знаками (ЦВЗ) для подальшого виявлення неправомірного використання зображення .

В зв'язку з цим актуальним є розробка сервісів та додатків для стійкого захисту векторних зображень в інформаційно-комунікаційних системах, що забезпечують зручне та надійне підтвердження авторства.

При розробці сервісів та додатків для захисту векторних зображень важливим є вибір стеганографічного методу вбудовування ЦВЗ, що є основою захисту.

В роботі [2] запропоновано метод вбудовування ЦВЗ у векторні зображення, який дозволяє вилучати ЦВЗ без наявності оригіналу векторного зображення чи самого ЦВЗ, що дозволяє значно спростити процедуру підтвердження авторства. При цьому метод забезпечує достатньо високий рівень стійкості ЦВЗ до навмисних спотворень зображення та незначний вплив вбудовування ЦВЗ на якість зображення [3].

Постановка задачі. Розробити сервіси та додатки вбудовування ЦВЗ у векторні зображення для захисту їх авторського права, що дозволять вилучати ЦВЗ без наявності оригіналу векторного зображення чи самого ЦВЗ. При цьому вони мають забезпечувати стійкість ЦВЗ до зловмисних атак, спрямованих на знищення чи підміну ЦВЗ при забезпеченні збереження високого рівня якості векторного зображення внаслідок вбудовування ЦВЗ

Розробка сервісів та додатків для захисту векторних зображень в інформаційно-комунікаційних системах

Розробку сервісів та додатків будемо здійснювати на основі методу, представленого у роботі [2].

Для розробки сервісів та додатків згідно методу, представленого у роботі [2] розробимо загальну структуру програмної реалізації методу у порядку виконання всіх етапів вбудовування чи вилучення цифрового водяного знаку.

Для забезпечення виконання всіх етапів вбудовування ЦВЗ в зображення згідно запропонованого методу пропонується використовувати незалежні програмні модулі. Це пояснюється тим, що більшість дій, які виконуються для вилучення ЦВЗ з зображення згідно методу, аналогічні діям, що виконуються для вбудовування ЦВЗ. Таким чином існує можливість використання багатьох програмних модулів як для вбудовування так і для вилучення ЦВЗ.

Виходячи зі структури методу, для вбудовування ЦВЗ в зображення програму пропонується будувати з таких блоків:

1. блок зчитування значень координат точок з файлу векторного зображення;
2. блок формування матриць розміром 8x8 для кожного масиву X та Y;
3. блок виконання двовимірного дискретного косинусного перетворення (ДКП) для кожної матриці розміром 8x8;
4. блок зчитування файлу ЦВЗ та перетворення у вектор двійкових даних;
5. блок визначення придатних для вбудовування матриць коефіцієнтів ДКП з використанням P_h;
6. блок вбудовування бітів ЦВЗ шляхом зміни коефіцієнтів матриць ДКП;
7. блок виконання оберненого двовимірного ДКП матриць з вбудованим ЦВЗ;
8. блок формування двох одновимірних масивів X та Y з отриманих матриць ДКП;
9. блок запису змінених координат точок у файл векторного зображення.

Для виконання вилучення ЦВЗ з векторного зображення програма буде складатися з наступних блоків:

1. блок зчитування значень координат точок векторного зображення з файлу векторного зображення з ЦВЗ та формування двох масивів для X та Y координати;
2. блок формування матриць розміром 8x8 для кожного масиву X та Y;
3. блок виконання двовимірного ДКП для кожної матриці розміром 8x8;
4. блок визначення придатних для вбудовування матриць коефіцієнтів ДКП з використанням P_h;
5. блок вилучення бітів ЦВЗ з матриць коефіцієнтів ДКП
6. блок перетворення вилученої двійкової бітової послідовності у вихідний формат ЦВЗ та збереження у файл.

Враховуючи послідовність етапів вбудовування та вилучення цифрового водяного знаку згідно запропонованого методу, пропонується загальна структура програми для приховування та вилучення ЦВЗ з використанням вищеописаних програмних блоків, яка показана на рисунку 1.

Вбудовування ЦВЗ по-різному буде впливати на окремі елементи, тому програма має враховувати можливість вбудовування ЦВЗ окремо в ці типи об’єктів. Тому у випадку, коли вбудовування ЦВЗ буде проводитись тільки у деякі об’єкти, блок формування масивів X та Y має записати у них координати точок тільки вибраних об’єктів, що зменшить об’єм даних для подальших обчислень і, відповідно, час вбудовування чи вилучення ЦВЗ.

Блок виконання прямого двовимірного ДКП також однаковий для програм вбудовування та вилучення ЦВЗ. При розробці блоку слід використати прискоренедвовимірне ДКП на основі множення матриць, як зазначалося в пункті 3.5., що значно підвищить швидкість обчислення матриць ДКП коефіцієнтів.

Блок зчитування файлу ЦВЗ програми вбудовування ЦВЗ має забезпечувати перетворення даних ЦВЗ у двійковий формат, оскільки вбудовування ЦВЗ згідно запропонованого методу проводиться по одному біту.

Блок визначення придатних для вбудовування матриць коефіцієнтів ДКП з використанням P_h є однаковим для програми вбудовування та вилучення ЦВЗ. Згідно запропонованого методу на етапі визначення придатних матриць важливим є вибір значення величини P_h, оскільки вона впливає на кількість придатних матриць, а отже і на максимальний розмір ЦВЗ, а також на рівень спотворення зображення внаслідок

вбудовування ЦВЗ. Оскільки оптимальне значення P_h може відрізнятися для кожного конкретного випадку залежно від розміру, типу та деталізації карти, а також від розміру ЦВЗ, виникає задача вибору цього значення. Тому важливим є розробка цього блоку з можливістю автоматичного підбору мінімального значення величини P_h для вбудовування усіх бітів ЦВЗ, що дозволить забезпечити мінімальний рівень спотворення для конкретного випадку.

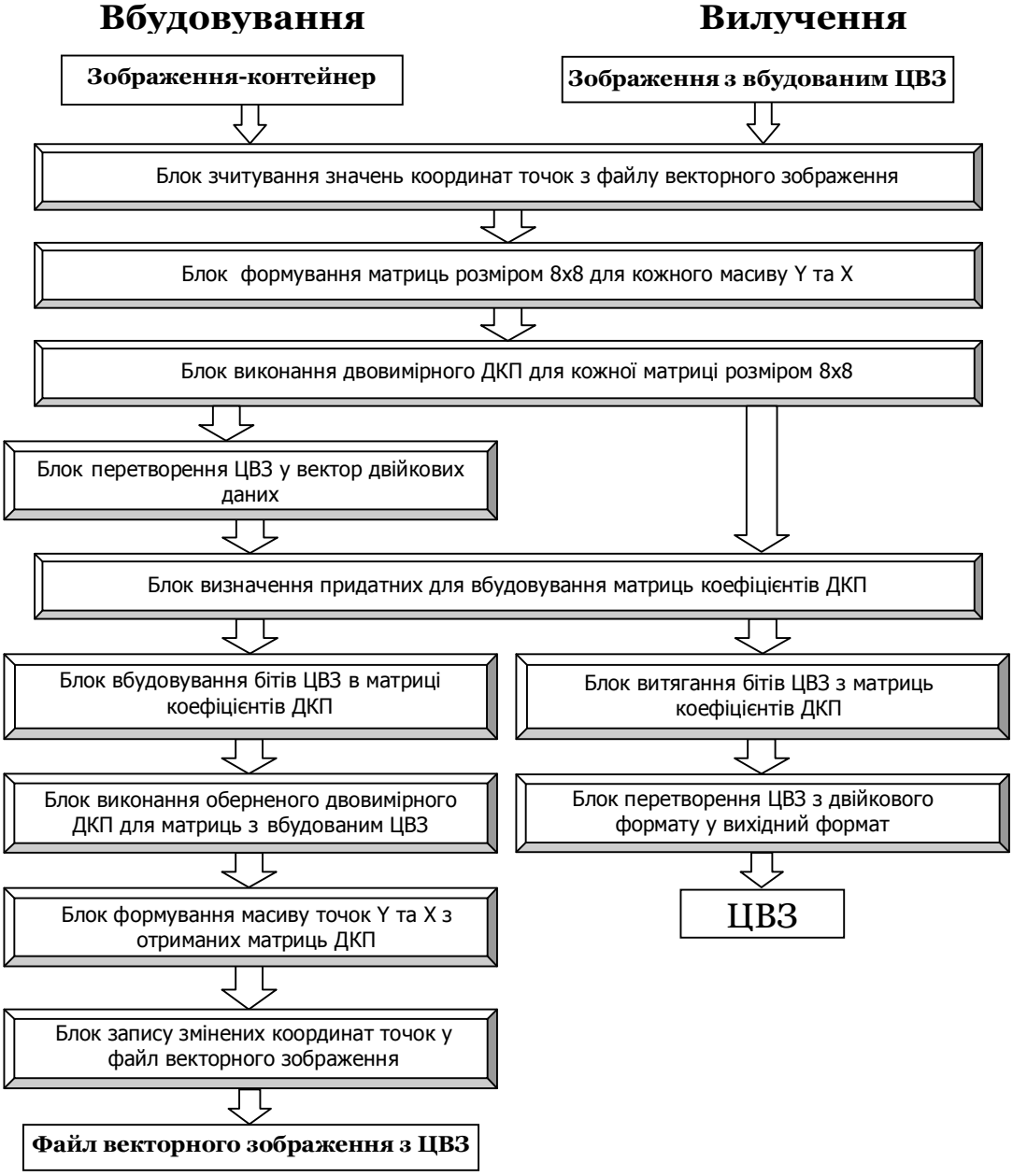

Рис. 1. Загальна структура програми для вбудовування та вилучення ЦВЗ

При розробці блоку вбудовування бітів ЦВЗ слід врахувати те, що всі операції відбуваються паралельно для масивів координати x та y окремо. Тому вбудовувати біти пропонується таким чином: перший біт – в матрицю коефіцієнтів ДКП для координат x, а наступний біт вбудовувати в масив з коефіцієнтами координат y, і т. д.

Також згідно методу на етапі вбудовування ЦВЗ важливим є вибір величини P для забезпечення правильного розпізнавання бітів ЦВЗ після вбудовування. Тому як і у випадку з P_h, важливим є забезпечення функції підбору мінімального значення величини P, при якій буде можливим відновити усі біти ЦВЗ.

Використовуючи можливості мови Ruby було розроблено програмні засоби вбудовування та вилучення ЦВЗ. Програмні засоби були розроблені згідно структури методу, запропонованого в роботі [3].

На рис. 2 показано зовнішній вигляд програми для вбудовування ЦВЗ на основі розглянутого методу.

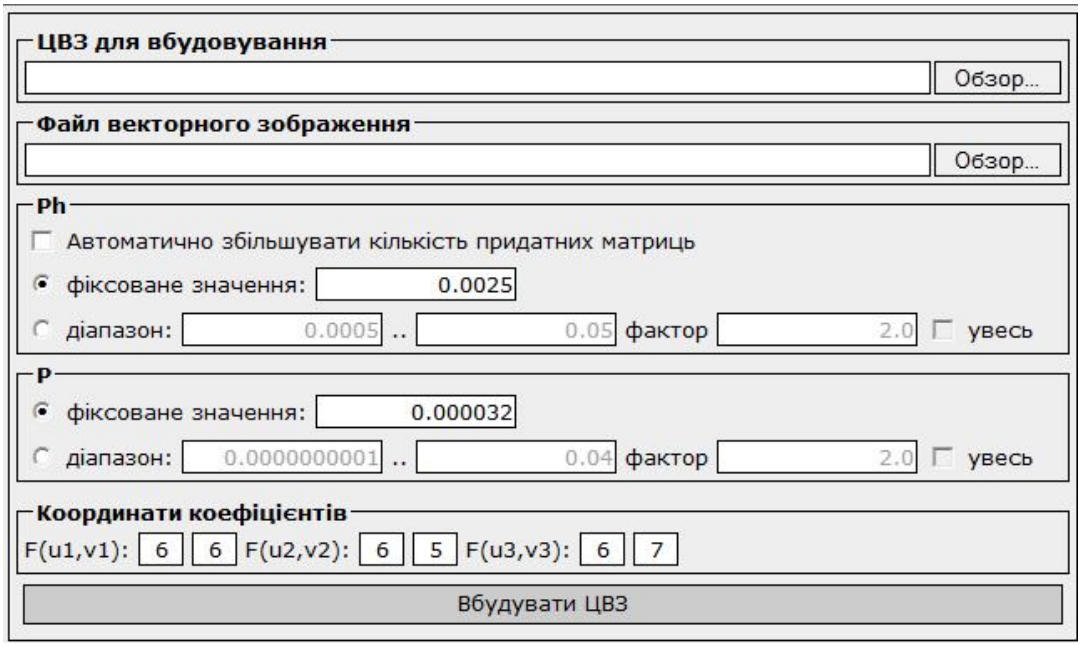

Рис. 2. Інтерфейс програми для вбудовування ЦВЗ

Як видно з рисунку, програма дозволяє обрати цілий ряд параметрів, які будуть визначати спосіб знаходження оптимальних значень коефіцієнтів P_h та P, а також позиції коефіцієнтів $F_i(u_1,v_1)$, $F_i(u_2,v_2)$ та $F_i(u_3,v_3)$.

Інтерфейсна частина програми передає програмі, написаній мовою Ruby, ті параметри, які визначив користувач. Далі відбувається аналіз отриманих параметрів та підбір оптимальних значень величин P_h та P для приховування ЦВЗ в файл векторного зображення.

Сам процес зчитування даних з файлу та формування масиву матриць зведений до виклику програмного методу `each_matrix` класу `File`, основний фрагмент коду якого має такий вигляд:

```
class File
  def self.each_matrix(file_name)
    enum = Enumerator.new do |yielder|
      xs = []; ys = []
      File.foreach(file_name) do |line|
        if line =~ /^\[(POLYLINE|POLYGON)\]/ .. line =~ /^\[END\]/ and line =~
/^Data\d+=/
          line.scan(/(\d+\.\d+),(\d+\.\d+)/).each do |x,y|
```

Завдяки принципу відкритих класів Ruby цим програмним методом був розширений функціонал класу `File` – одного зі стандартних класів мови Ruby. Робота цього програмного методу полягає в послідовному зчитуванні рядків файлу та їх аналізу. Якщо поточна стрічка файлу містить перелік координат і ці координати належать блоку опису координат полігона чи полілінії, то метод `each_matrix` проводить розбір такого рядку, отримання з неї пар координат X та Y для чергової точки та додавання до тимчасового масиву. Як тільки цей масив міститиме 64 координати, то такий масив одразу ж перетворюється на повноцінний об’єкт класу `Matrix`, викликом одного з його конструкторів. А оскільки було також

розширено можливості класу `Matrix`, то тепер об'єкти цього класу можуть виконувати і пряме та обернене ДКП.

Після обробки усього вмісту файлу з картою, який було обрано в інтерфейсі програми користувачем, починається процес пошуку оптимальних значень величин P_h та P для приховування ЦВЗ.

Увесь процес пошуку оптимальних величин P_h та P для вбудовування ЦВЗ знаходиться в модулі `BestParam`, який в коді програми описується таким чином:

```
module BestParam
  def self.set(matrixes, dwm, params_H, params_P)
    suitable_matrixes = nil
        check_P = lambda do
      if Matrix.dct_P >= Matrix.dct_H
        Informer.| :skip_P
        next false
      end
      srand Matrix::SEED
      max_diff, total_diff, recovered_count, all_are_ok =
...
```

Цей блок має один-єдиний публічний програмний метод `set`, призначенням якого є аналіз масиву матриць, який отримано під час зчитування та аналізу файлу карти, та проведення над ним ДКП і приховування даних ЦВЗ-файлу. Якщо дані ЦВЗ файлу повністю можна приховати в цьому масиві матриць (а відтак, і в файлі векторного зображення), то програма намагається провести ще одну спробу зробити те саме, однак з меншими величинами P_h та P. Крок такого зменшення задається користувачем в інтерфейсі програми. Також користувач визначає необхідні йому діапазони пошуку значень величин P_h та P.

Найменші знайдені значення величин P_h та P, при яких програма дозволяє приховати усі біти ЦВЗ і в подальшому відновити їх (для цього кожна спроба приховування автоматично контролюється програмою і відбувається спроба відновлення прихованого біту інформації) вважаються оптимальними. Якщо такі оптимальні значення величин P_h та P для даного векторного зображення та ЦВЗ були знайдені, то програма виконує вбудовування ЦВЗ в файлі векторного зображення.

Вбудовування бітів ЦВЗ відбувається за допомогою програмного методу `hide_matrixes` класу `File`. Ключовий фрагмент коду програми цього методу має такий вигляд:

```
class File
  def self.hide_matrixes(file_name1, matrixes)
    ext_length = File.extname(file_name1).length
    file_name2 = file_name1.dup.insert(-ext_length-1, "_dwm")
    point_number = 0
    File.open(file_name2, "w") do |f2|
      File.foreach(file_name1) do |line|
        if line =~ /^\[(POLYLINE|POLYGON)\]/ .. line =~ /^\[END\]/ and line =~
/^(Data\d+=)/
```

Робота цього програмного методу полягає в зворотній процедурі відносно методу each_matrix. Програма послідовно зчитує дані з вхідного файлу і записує їх у вихідний. Однак, якщо серед цих даних знаходяться координати полігону чи полілінії, то їхні координати будуть братись не з вхідного файлу, а з масиву матриць (з який перед цим було проведено приховування ЦВЗ з оптимальними значеннями величин P_h та P). Таким чином, користувач, поряд з вхідним файлом карти, отримає інший файл, який буде відрізнятись лише координатами певних координат точок, в які вбудовано ЦВЗ.

Для вилучення ЦВЗ було розроблено окрему програму, яка простіша від програми для вбудовування та менша за обсягом коду. На рисунку 3 представлено інтерфейс програми для вилучення ЦВЗ.

Робота програми для вилучення ЦВЗ з файлу векторного зображення простіша в порівнянні з програмою приховування ЦВЗ, оскільки відсутня необхідність в підборі оптимальних значень величин P_h та P та виконанні оберненого ДКП.

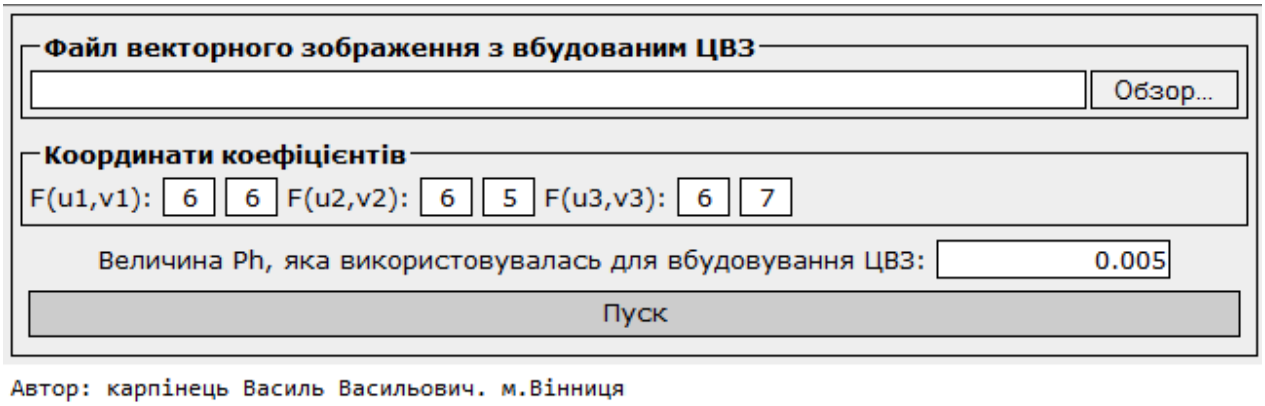

Рис. 3. Інтерфейс програми вилучення ЦВЗ

```
# encoding: windows-1251
$stderr = $stdout
Dir["./lib/**/*.rb"].each {|lib| require lib}
START_TIME = Time.now
map = $stdin.gets.chomp
Matrix.dct_H = $stdin.gets.chomp.to_f
[:dct_A, :dct_B, :dct_C].each do |el|
  Matrix.send "#{el}=", $stdin.gets.chomp.split(',').map(&:to_i)
end
...
```

Програмний метод `File.each_matrix` застосовується і у випадку вилучення ЦВЗ, оскільки потрібно знову отримати масив матриць і провести над ними ДКП.

Отримавши масив матриць програма знаходить серед них ті, які задовольняють умові, заданій користувачем в інтерфейсі, а саме значення величини P_h. Провівши перевірку серед таких матриць на предмет того, на яку величину (додатну чи від'ємну) відрізняється значення коефіцієнта $F_i(u_1,v_1)$ від середнього значення коефіцієнтів $F_i(u_2,v_2)$ та $F_i(u_3,v_3)$, програма визначає який біт був прихований в результаті перетворення, здійсненого над даною матрицею. Після цього отримані розпізнані біти ЦВЗ записуються у файл формату ЦВЗ.

Було проведено оцінювання швидкодії розроблених додатків вбудовування та вилучення ЦВЗ. Результати показали, що розроблені додатки дозволяють достатньо швидко виконувати вбудовування ЦВЗ, зокрема, 3,5 секунди для вбудовування 768 біт ЦВЗ у векторне зображення, що складається з 74157 точок, та проводити вилучення ЦВЗ – 1,39 секунди відповідно. Додатки дозволяють вбудовувати ЦВЗ будь-якого формату у векторні зображення різних типів, а також вилучати ЦВЗ без наявності оригіналу зображення чи самого ЦВЗ.

Висновки. Розроблено сервіси та додатки для захисту векторних зображень в інформаційно-комунікаційних системах, що дають можливість при вилученні ЦВЗ підтверджувати авторство без необхідності наявності оригіналу зображення чи самого ЦВЗ. При цьому забезпечується достатній рівень стійкості, і незначні спотворення при вбудовуванні ЦВЗ. Особливістю розроблених сервісів та додатків є те, що вони складаються

з окремих незалежних програмних блоків, які можна використовувати при вбудовуванні та при вилученні ЦВЗ, що скорочує розмір програмного коду та підвищує надійність їх роботи.

ЛІТЕРАТУРА

1. В.О. Хорошко, О.Д. Азаров, М.Є. Шелест, Ю.Є. Яремчук. Основи комп'ютерної стеганографії. Навчальний посібник. – Вінниця: ВДТУ. – 2003. – 143 с.

2. Карпінець В. В., Яремчук Ю. Є. Зменшення відхилень координат точок внаслідок вбудовування цифрових водяних знаків у векторні зображення // Правове, нормативне, та метрологічне забезпечення системи захисту інформації в Україні – 2010. – № 2(21). – С.69-78.

3. Карпінець В. В., Яремчук Ю. Є. Аналіз впливу цифрових водяних знаків на якість векторних зображень // Сучасний захист інформації. – 2011. – №1. – С.72-82.

Надійшла: 15.12.2011

Рецензент: д.т.н., проф. Щербак Л.М.