Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ПРИХОВУВАННЯ ДАНИХ У ПРОСТОРОВІЙ ОБЛАСТІ НЕРУХОМИХ ЗОБРАЖЕНЬ МЕТОДОМ БЛОКОВОГО ВБУДОВУВАННЯ, МЕТОДОМ КВАНТУВАННЯ ТА МЕТОДОМ «ХРЕСТА»

Методичні рекомендації
до лабораторної роботи з дисципліни «Стеганографія»
для студентів спеціальності 125 «Кібербезпека»

Харків – 2019

УДК 004.415.24 (075.8)
П 77

Рецензенти:

В. А. Краснобаєв – доктор технічних наук, професор, професор кафедри електроніки і управляючих систем Харківського національного університету імені В. Н. Каразіна;

О. Г. Толстолузька – доктор технічних наук, старший науковий співробітник, професор кафедри теоретичної та прикладної системотехніки Харківського національного університету імені В. Н. Каразіна.

Затверджено до друку рішенням Науково-методичної ради
Харківського національного університету імені В. Н. Каразіна
(протокол № 1 від 30.10.2019 р.)

Приховування даних у просторовій області нерухомих зображень методом
П 77 блокового вбудовування, методом квантування та методом «хреста» : методичні рекомендації до лабораторної роботи з дисципліни «Стеганографія» для студентів спеціальності 125 «Кібербезпека» / уклад. О. О. Кузнецов, М. О. Полуяненко, Т. Ю. Кузнецова. – Харків : ХНУ імені В. Н. Каразіна, 2019. – 48 с.

Методичні рекомендації розроблено для студентів факультету комп'ютерних наук за спеціальністю «Кібербезпека». Матеріали видання мають допомогти студентам усвідомити специфіку безпеки інформаційних і комунікаційних систем та особливості професійної наукової діяльності у галузі захисту інформації. Передбачається, що в результаті навчання студенти оволодіють початковими навичками роботи з дисципліни «Стеганографія», вироблять ставлення до використання методів та принципів приховування даних, набудуть практичних вмінь та навичок щодо розробки стеганографічних систем.

УДК 004.415.24 (075.8)

ЗМІСТ

1. МЕТА І ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ «ПРИХОВУВАННЯ ДАНИХ У ПРОСТОРОВІЙ ОБЛАСТІ НЕРУХОМИХ ЗОБРАЖЕНЬ МЕТОДОМ БЛОКОВОГО ВБУДОВУВАННЯ, МЕТОДОМ КВАНТУВАННЯ ТА МЕТОДОМ "ХРЕСТА"»

Мета роботи: закріпити теоретичні знання за темою «Приховування даних у просторовій області нерухомих зображень методом блокового вбудовування, методом квантування та методом "хреста"», набути практичних вмінь та навичок щодо розробки стеганографічних систем, дослідити властивості стеганографічних методів, що засновані на низькорівневих властивостях зорової системи людини (ЗСЛ).

Лабораторна робота виконується у середовищі символьної математики MathCAD версії 12 або вище. Допускається виконання лабораторної роботи із використанням інших середовищ або мов програмування, що вивчалися студентами під час навчання.

Завдання до лабораторної роботи

Завдання 1. Реалізація алгоритмів вбудовування та вилучення повідомлень методом блокового вбудовування

Реалізувати у середовищі символьної математики MathCAD (або в іншому середовищі / мові програмування) алгоритми приховування та вилучення даних у просторовій області зображень методом блокового приховування. Застосовуючи розроблену програмну реалізацію, виконати стеганографічне кодування інформаційного повідомлення, тобто сформувати заповнений контейнер (стеганограму). Виконати зорове порівняння пустого та заповненого контейнера та переконатися у відсутності помітних похибок. Переконатися в автентичності вилученого повідомлення.

Завдання 2. Реалізація алгоритмів вбудовування та вилучення повідомлень методом квантування

Реалізувати у середовищі символьної математики MathCAD (або в іншому середовищі / мові програмування) алгоритми приховування та вилучення даних у просторовій області зображень методом квантування. Застосовуючи розроблену програмну реалізацію, виконати стеганографічне кодування інформаційного повідомлення, тобто сформувати заповнений контейнер (стеганограму). Виконати зорове порівняння пустого та заповненого контейнера та переконатися у відсутності помітних похибок. Переконатися в автентичності вилученого повідомлення.

Завдання 3. Реалізація алгоритмів вбудовування та вилучення повідомлень методом Куттера–Джордана–Боссена (методом «хреста»)

Реалізувати у середовищі символьної математики MathCAD (або в іншому середовищі / мові програмування) алгоритми приховування та

вилучення даних у просторовій області зображень методом «хреста». Застосовуючи розроблену програмну реалізацію, виконати стеганографічне кодування інформаційного повідомлення, тобто сформувати заповнений контейнер (стеганограму). Виконати зорове порівняння пустого та заповненого контейнера та переконатися у відсутності помітних похибок. Переконатися в автентичності вилученого повідомлення (або зафіксувати можливі помилки). Провести експериментальні дослідження ймовірнісних властивостей методу «хреста», отримати емпіричні залежності ймовірності правильного вилучення даних та частки внесених при цьому похибок у контейнер-зображення.

Додаткове завдання. Удосконалення алгоритмів вбудовування та вилучення повідомлень методом Куттера–Джордана–Боссена (методом «хреста») застосуванням завадостійкого кодування

Реалізувати у середовищі символьної математики MathCAD (або в іншому середовищі / мові програмування) алгоритми завадостійкого кодування інформаційних даних для покращення ймовірнісних властивостей стеганографічного методу вбудовування даних Куттера–Джордана–Боссена (методу «хреста»).

2. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ З ОРГАНІЗАЦІЇ САМОСТІЙНОЇ РОБОТИ

1. Вивчити теоретичний матеріал лекції «Приховування даних у просторовій області зображень методом блокового вбудовування, методом квантування та методом "хреста"».

2. Вивчити матеріал основного джерела літератури (Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография):

a. метод блокового вбудовування (ст. 97–98);

b. метод квантування (ст. 103–106);

c. метод Куттера–Джордана–Боссена (ст. 106–110).

3. Вивчити матеріал додаткових джерел:

a. структура лінійних блокових кодів, стандартне розташування, коди Хеммінга (Р. Блейхут. Теория и практика кодов, контролирующих ошибки, ст. 61–73);

b. принципи побудови та властивості генераторів псевдовипадкових послідовностей (Иванов М. А., Чугунков И. В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей, ст. 5–64).

4. Вивчити основні команди у середовищі символьної математики MathCAD щодо роботи із зображеннями.

5. Підготувати відповіді на контрольні запитання.

6. Підготувати бланк звіту з лабораторної роботи.

Допуск до виконання лабораторної роботи здійснюється за результатами письмового опитування (контрольної роботи).

3. ЗАГАЛЬНОТЕОРЕТИЧНІ ПОЛОЖЕННЯ ЗА ТЕМОЮ ЛАБОРАТОРНОЇ РОБОТИ

3.1. Метод блокового приховування

Метод блокового приховування за своєю суттю є подальшим розвитком методів модифікації найменш значущих бітів (НЗБ, LSB – Least Significant Bit). Він має певні переваги у порівнянні із методами модифікації НЗБ, але також наслідує і певні їхні недоліки.

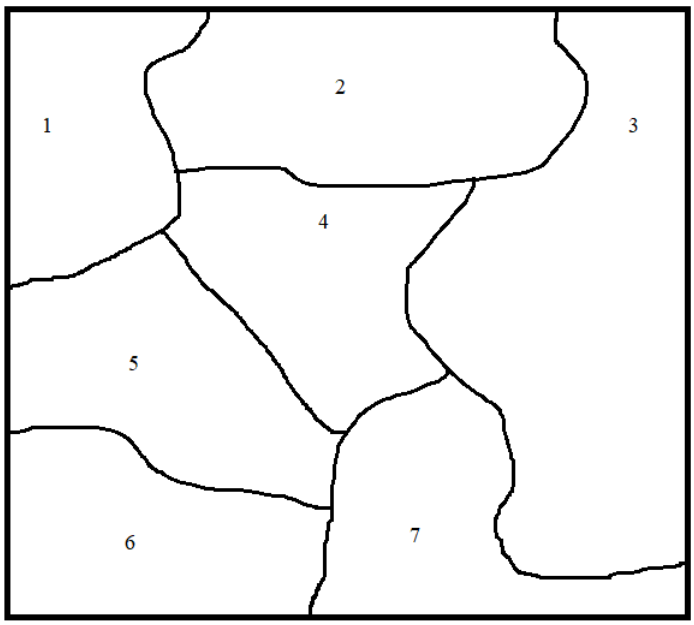


Рис. 1. Розбиття контейнера-зображення на блоки, N = 7 (варіант)

Для реалізації методу блокового приховування зображення-контейнер розбивається на *N* блоків, що не перетинаються. Це розбиття тримається в секреті від зловмисника, тобто правило розбиття контейнера на блоки задається секретним ключем.

Правило розбиття може бути довільним і реалізуватися, наприклад, таким чином (див. рис. 1, 2). Цифрами на рисунках вказано номери блоків.лючем.

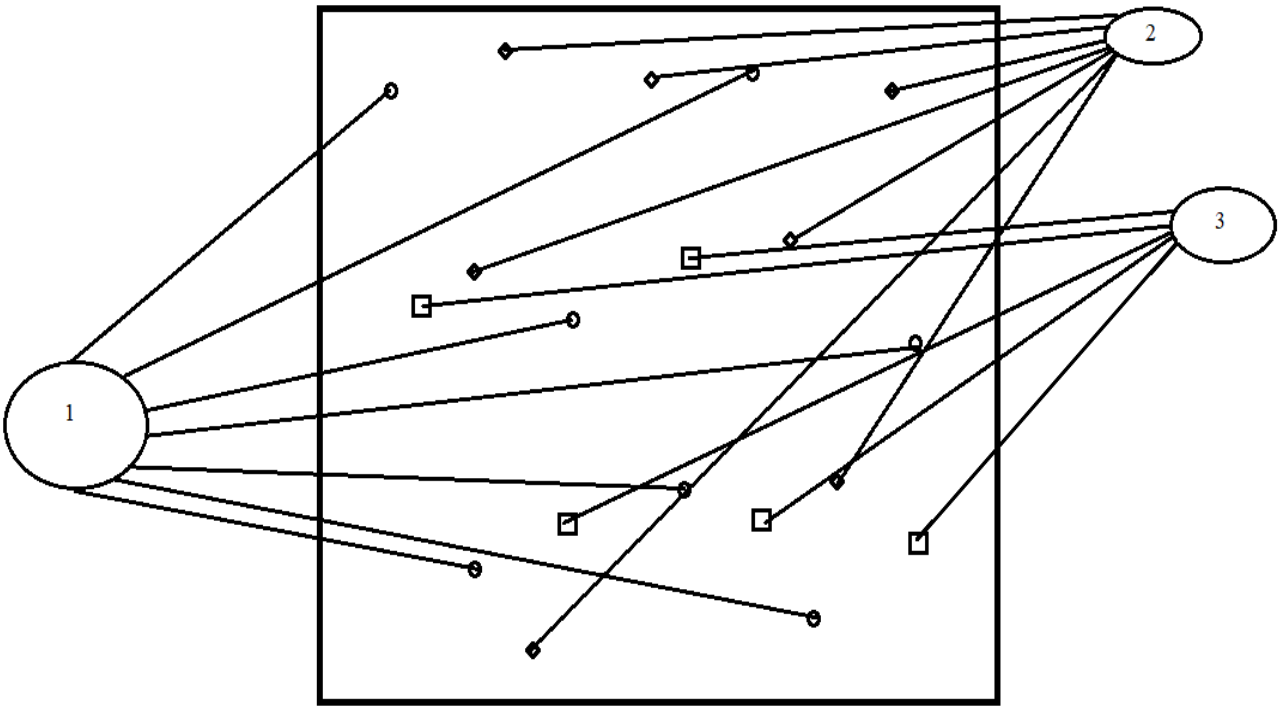


Рис. 2. Розбиття контейнера-зображення на блоки, N = 3 (варіант)

У кожному блоці буде приховано тільки один біт, отже, кількість блоків N визначає обсяг прихованої в контейнері інформації, а пропускну спроможність стеганографічного каналу можна розрахувати як відношення

$$C = \frac{N}{V},$$

де V – бітовий обсяг контейнера.

Як приклад, розглянемо вихідне зображення 100*100 пікселів у форматі BMP24, що представлено масивом 300*100 байтів (300 стовпців і 100 рядків). Розіб'ємо зображення на блоки таким чином: кожен байтовий стовпець відповідає одному блоку. Тоді в цьому контейнері ми зможемо приховати 300 бітів, пропускна спроможність буде дорівнювати

$$C = \frac{N}{V} = \frac{300}{300 \cdot 100 \cdot 8} = 0{,}00125.$$

3.2. Метод квантування зображення

До методів приховування в просторовій області можна також віднести метод квантування зображення, заснований на міжпіксельній залежності, яку можна описати деякою функцією Θ. У найпростішому випадку можна обчислити різницю між суміжними пікселями c_i і c_{i+1} (або c_{i-1} і c_i) і задати її як параметр функції $\Theta : \Delta_i = \Theta(c_i - c_{i+1})$, де Δ_i – дискретна апроксимація різниці сигналів $c_i - c_{i+1}$.

Оскільки Δ_i – ціле число, а реальна різниця $c_i - c_{i+1}$ – дійсне число, то виникають помилки квантування $\delta_i = \Delta_i - \varepsilon_i$. Для сигналів, що сильно корелюються, ця помилка є близькою до нуля: $\delta_i \approx 0$.

При цьому методі приховування інформації проводиться шляхом коригування різницевого сигналу Δ_i. Стеганоключ є таблицею, що кожному можливому значенню Δ_i ставить у відповідність певний біт, наприклад:

Δ_i	–4	–3	–2	–1	0	1	2	3	4
b_i	1	0	1	1	0	0	1	0	1

Щоб приховати і-й біт повідомлення, обчислюється різниця Δ_i. Якщо при цьому b_i не відповідає секретному біту, який необхідно приховати, то значення Δ_i замінюється найближчим Δ_i, для якого така умова виконується. При цьому відповідним чином коригуються значення інтенсивностей пікселів, між якими обчислювалася різниця Δ_i. Вилучення секретного повідомлення здійснюється відповідно до значення b_i^*, що відповідає різниці Δ_i^*.

3.3. Метод Куттера–Джордана–Боссена

М. Куттер (M. Kutter), Ф. Джордан (F. Jordan) і Ф. Боссен (F. Bossen) запропонували алгоритм вбудовування в канал синього кольору зображення, що має RGB-кодування, оскільки до синього кольору зорова система людини є найменш чутливою. Розглянемо алгоритм передачі одного біта секретної інформації в запропонованому методі.

Нехай m_i – біт, який підлягає вбудовуванню; $C = \{R, G, B\}$ – зображення-контейнер; p = (x, y) – псевдовипадковий піксель контейнера, в який буде виконуватися вбудовування.

Секретний біт m_i вбудовується в канал синього кольору шляхом модифікації яскравості

$$\lambda_{x,y} = 0.29890 \cdot R_{x,y} + 0.58662 \cdot G_{x,y} + 0.11448 \cdot B_{x,y};$$

$$B'_{x,y} = \begin{cases} B_{x,y} - \upsilon \cdot \lambda_{x,y}, & при\ m_i = 0 \\ B_{x,y} - \upsilon \cdot \lambda_{x,y}, & при\ m_i = 1 \end{cases} = B_{x,y} + (2 \cdot m_i - 1) \cdot \upsilon \cdot \lambda_{x,y},$$

де υ – константа, що визначає «енергію» вбудованого сигналу. Її величина залежить від призначення стеганосистеми. Чим більше υ, тим вище стійкість вбудованої інформації до спотворень, проте і тим сильніше її помітність.

Одержувач отримує біт, не маючи первинного зображення, тобто «наосліп». Для цього виконується прогнозування значення первинного, немодифікованого пікселя на основі значень сусідніх пікселів. Для отримання оцінки пікселя запропоновано використовувати значення декількох пікселів, розміщених в тому ж стовпці і в тому ж рядку масиву графічного контейнера. Використовують «хрест» пікселів розміром 7x7. Оцінка $\hat{B}^*_{x,y}$ виходить у вигляді

$$\hat{B}^*_{x,y} = \frac{1}{4 \cdot \sigma} \cdot \left[\sum_{i=-\sigma}^{+\sigma} B^*_{x+i,y} + \sum_{j=-\sigma}^{+\sigma} B^*_{x+i,j} - 2 \cdot B^*_{x,y} \right],$$

де σ – кількість пікселів зверху (знизу, зліва, справа) від оцінюваного пікселя (в разі хреста 7x7 $\sigma = 3$).

Під час вилучення вбудованого біта обчислюється різниця δ між поточним ($B^*_{x,y}$) і прогнозованим ($\hat{B}^*_{x,y}$) значеннями інтенсивності пікселя p = (x, y):

$$\delta = B^*_{x,y} - \hat{B}^*_{x,y}.$$

Знак δ означатиме вбудований біт: якщо $\delta < 0$, то $m_i = 0$; якщо $\delta > 0$, то $m_i = 1$.

Функції вбудовування та вилучення в цьому методі є несиметричними, тобто функція вилучення не є зворотною функцією до функції вбудовування. Хоча, як зазначають автори методу, правильне розпізнавання біта повідомлення в разі застосування описаних процедур є високоймовірним, проте не стовідсотковим.

Для зменшення ймовірності помилок вилучення було запропоновано в процесі вбудовування кожен біт повторювати кілька разів (багаторазове вбудовування). Оскільки при цьому кожен біт був повторений τ разів, то виходить τ оцінок одного біта повідомлення. Секретний біт вилучається за результатами усереднення різниці між реальним і оціненим значеннями інтенсивності пікселя в отриманому контейнері:

$$\delta = \tau^{-1} \cdot \sum_{i=1}^{\tau} \left[B^*_{x,y} - \hat{B}^*_{x,y} \right].$$

Як і в попередньому випадку, знак усередненої різниці δ визначатиме значення вбудованого біта.

Цей алгоритм є стійким до багатьох відомих видів атак: НЧ фільтрації зображення, його компресії відповідно до алгоритму JPEG, обрізання країв.

4. ПИТАННЯ ДЛЯ ПОТОЧНОГО КОНТРОЛЮ ПІДГОТОВЛЕНОСТІ СТУДЕНТІВ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Метод блокового вбудовування та його зв’язок з лінійними блоковими кодами з контролем парності.

2. Ймовірнісні характеристики методу блокового вбудовування: ймовірність правильного вилучення повідомлень та ймовірність виникнення помилок.

3. Поняття контрастності зображення. Чутливість зорової системи людини до незначної зміни контрастності. Вбудовування даних у нерухомі зображення методом квантування.

4. Метод вбудовування даних у нерухомі зображення Куттера–Джордана–Боссена (метод «хреста»). Лінійне передбачення сигналів при отриманні даних методом «хреста».

5. ІНСТРУКЦІЯ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1. Реалізація алгоритмів вбудовування та вилучення повідомлень методом блокового вбудовування

1.1. Завантажуємо вихідні дані: контейнер – нерухоме зображення (в форматі *.bmp24); інформаційне повідомлення – текстовий документ (у форматі *.txt). Для цього в середовищі MathCAD виконуємо дії, аналогічні п. 1.1 інструкції до лабораторної роботи «Приховування даних у просторовій області нерухомих зображень шляхом модифікації найменш значущого біта».

1.2. Перетворимо масив інформаційних даних. Для цього в середовищі MathCAD виконуємо дії, аналогічні п. 1.2 інструкції до лабораторної роботи «Приховування даних в просторовій області нерухомих зображень шляхом модифікації найменш значущого біта».

1.3. Реалізуємо алгоритм вбудовування даних у просторову область зображень методом блокового вбудовування. Для цього скористаємося такою процедурою:

```
S1 := | for i ∈ 0.. cols(R) − 1
      |   b ← mod( Σ_{j=0}^{rows(R)−1} R_{j,i}, 2 )
      |   if M_b_i ≠ b
      |     P ← D_B(R_{0,i})
      |     P_0 ← P_0 ⊕ 1
      |     S1_{0,i} ← B_D(P)
      |   S1_{0,i} ← R_{0,i} if M_b_i = b
      |   for j ∈ 1.. rows(R) − 1
      |     S1_{j,i} ← R_{j,i}
      | S1
```

Наведена процедура реалізує поелементне вбудовування бітового масиву інформаційних даних M_b в біти парності окремих блоків зображення. При цьому зображення розбите на блоки за стовпчиками, тобто кожен стовпець масиву растрових даних каналу червоного кольору являє собою окремий блок зображення, в який вбудовується відповідний біт інформаційного повідомлення. Біт парності b для кожного блока обчислюється у друго-

му рядку процедури, за допомогою підсумовування за модулем 2 всіх елементів блока. Якщо біт парності поточного блока не збігається зі значенням вбудованого у цей блок інформаційного біта, проводиться модифікація (інвертування) найменш значущого біта в першому рядку блока (наступні три рядки процедури). При цьому змінюється значення біта парності, що після виробленої модифікації збігається зі значенням вбудованого інформаційного біта даних. Якщо значення біта парності спочатку збігалося зі значенням вбудованого інформаційного біта даних, то проводиться перезапис першого елемента поточного блока контейнера. Аналогічним чином перезаписуються і всі інші елементи блока контейнера, що не модифікуються (останні рядки процедури).

Таким чином, вбудовування даних здійснюється в біти парності окремих блоків зображення, при цьому модифікуються перші елементи блока. Результат вбудовування (заповнений контейнер) зберігається в масиві S1. Слід зазначити, що значення найменш значущого біта першого елемента блока не завжди збігатиметься зі значенням вбудованого інформаційного біта. Збігаються лише біт парності та інформаційний біт даних.

Для візуального перегляду результату вбудовування інформаційних даних виведемо вихідний масив растрових даних червоного кольору R та отриманий масив S1 зі зміненими бітами парності блоків. Для розглянутого прикладу маємо:

R =

	0	1	2	3	4
0	86	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	183
8	191	192	194	199	194
9	186	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	177	185	187	186	180
14	173	176	174	166	165
15	160	162	158	153	...

S1 =

	0	1	2	3	4
0	87	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	183
8	191	192	194	199	194
9	186	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	177	185	187	186	180
14	173	176	174	166	165
15	160	162	158	153	...

З представлених даних видно, що, наприклад, значення $R_{0,1}$, $R_{0,2}$, $R_{0,3}$ повністю ідентичні відповідним значенням $S_{0,1}$, $S_{0,2}$, $S_{0,3}$. Практично це означає, що значення бітів парності першого, другого і третього стовпців масиву R збіглися з вбудованими інформаційними бітами повідомлення. Навпаки, значення $R_{0,0}$ відрізняється на одиницю від відповідного значення масиву S1. Це означає зміну біта парності нульового блока контейнера в процесі вбудовування повідомлення. Відзначимо, що внесені спотворення знаходяться нижче порога зорової чутливості людини.

Графічну інтерпретація порожнього і заповненого контейнера (каналу червоного кольору в градаціях сірого) наведено на наступному рисунку, з якого слідує, що візуально внесені спотворення не помітні, що підтверджує висновок про чутливість органів зору людини.

R

S1

Отриманий заповнений масив S1 записуємо в канал червоного кольору контейнера. Виконуємо команду

«WRITERGB("Stego_Blok.bmp"):=augment(S1,G,B)».

В результаті виконання команди система MathCAD формує на фізичному носії новий файл з ім’ям «Stego_Blok.bmp».

Для графічного відображення вихідного (порожнього) і заповненого контейнера виконаємо вставку відповідних зображень:

"1.bmp"

"Stego_Blok.bmp"

Переконаємося у відсутності видимих спотворень.

1.4. Реалізуємо алгоритм вилучення даних з просторової області зображень методом блочного вбудовування. Для цього в тому ж вікні середовища

MathCAD виконуємо команди читання растрових даних нерухомого зображення з заданого файла (файла заповненого контейнера) у вигляді двовимірного масиву цілих чисел. Для розглянутого прикладу виконуємо команди:

«C1:=READRGB(“Stego_Blok.bmp”)»,
«R1:=READ_RED(“Stego_Blok.bmp”)»,
«G1:=READ_GREEN(“Stego_Blok.bmp”)»,
«B1:=READ_BLUE(“Stego_Blok.bmp”)».

Отримуємо такий результат:

R1 =

	0	1	2	3	4
0	87	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	...

R1

Далі обчислюємо біти парності для кожного блока даних контейнера і формуємо масив отриманих (витягнутих) інформаційних бітів. Для цього використовуємо таку процедуру:

$$\mathrm{M\_b1} := \left| \begin{array}{l} \text{for } i \in 0..\operatorname{cols}(\mathrm{R1}) - 1 \\ \quad \mathrm{M\_b1}_i \leftarrow \operatorname{mod}\left(\sum_{j=0}^{\operatorname{rows}(\mathrm{R})-1} \mathrm{R1}_{j,i}, 2 \right) \\ \mathrm{M\_b1} \end{array} \right.$$

За допомогою цієї процедури обчислюється для всіх стовпців (блоків) масиву R1 біт парності, який і є вбудованим інформаційним бітом. В результаті маємо лінійний бітовий масив M_b1, заповнений бітами парності окремих блоків масиву растрових даних каналу червоного кольору заповненого контейнера.

M_b1 =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

Порівняння даних масивів вбудованих та витягнутих (отриманих) бітів даних дозволяє підтвердити правильність виконання алгоритмів вбудовування-вилучення.

1.5, 1.6. Формування масиву цілих чисел, що відповідають ASCII кодуванню вбудованих символів повідомлення, та запис у текстовий файл витягнутих (отриманих) інформаційних даних здійснюється аналогічно п. 1.5, 1.6 інструкції до лабораторної роботи № 1.

Завдання 2. Реалізація алгоритмів вбудовування та вилучення повідомлень методом квантування

2.1, 2.2. Завантажуємо вихідні дані та перетворюємо масив інформаційних даних (відповідно до п. 1.1, 1.2).

2.3. Реалізуємо алгоритм вбудовування даних у просторову область зображень методом квантування. Для цього спочатку сформуємо випадковий секретний ключ – таблицю квантування d, скориставшись такою процедурою:

$$
\begin{aligned}
d := \;& \text{for } i \in 0..510 \\
& \quad d_{0,i} \leftarrow i - 255 \\
& \quad d_{1,i} \leftarrow \text{ceil}(\text{rnd}(2)) - 1 \\
& d
\end{aligned}
$$

Ця процедура псевдовипадковим чином (з використанням вбудованого датчика "rnd ()") заповнює таблицю квантування для всіх можливих значень перепадів яскравості зображення. Приклад заповненої таблиці має вигляд:

d =

	250	251	252	253	254	255	256	257	258	259	260
0	-5	-4	-3	-2	-1	0	1	2	3	4	5
1	0	0	0	1	0	1	1	1	0	1	...

Нульовий рядок масиву d заповнений усіма можливими (від –255 до +255) значеннями перепадів яскравості, перший рядок заповнений датчиком "rnd()". Функція "ceil()" округлює аргумент до найближчого цілого, функція "rnd()" формує рівномірно розподілені на заданій ділянці псевдовипадкові значення.

Для вбудовування даних з використанням секретного ключа d скористаємося процедурою, яка реалізує поелементне вбудовування бітового масиву інформаційних даних M_b в значення різниць елементів нульового і першого стовпця масиву червоного кольору контейнера. Іншими словами, кожен окремий біт вбудовується в одну різницю, номер біта задає номер рядка контейнера.

Поточне значення різниці b знаходиться в таблиці квантування. Значення вбудованого біта M_b_i порівнюється з бітовим значенням $d_{1,b+255}$ з другого рядка матриці квантування. При співпадінні цих значень рівень контрастності в даній позиції не змінюється.

```
S2 := | for i ∈ 0.. rows(R) − 1
      |   b ← R_{i,0} − R_{i,1}
      |   S2_{i,0} ← R_{i,0}  if M_b_i = d_{1,b+255}
      |   if M_b_i ≠ d_{1,b+255}
      |     j ← 1
      |     while M_b_i ≠ d_{1,b+255+j} ∧ j < 509
      |       j ← j + 1
      |     S2_{i,0} ← R_{i,0} + d_{0,b+255+j} − b
      |   for j ∈ 1.. cols(R) − 1
      |     S2_{i,j} ← R_{i,j}
      | S2
```

При неспівпадінні за заздалегідь заданим правилом (в даному випадку, за правилом «пошук праворуч») знаходиться найближча позиція, для якої значення M_b_i та $d_{1,b+255}$ збігаються (співпадають). Вбудовування інформації в такому випадку полягає в модифікації різниці (відповідно до знайдених значень з таблиці квантування). Інша частина зображення, що не бере участі в модифікації різниці, перезаписується з пустого контейнера без зміни.

Таким чином, вбудовування даних здійснюється в значення різниці між окремими елементами масиву R. Результат вбудовування (заповнений контейнер) зберігається в масиві S2. Для візуального перегляду результату вбудовування інформаційних даних виведемо вихідний масив растрових даних червоного кольору R і отриманий масив S2 зі зміненими значеннями різниць. Для розглянутого прикладу маємо:

R =

	0	1	2	3	4
0	86	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	183
8	191	192	194	199	194
9	186	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	177	185	187	186	180
14	173	176	174	166	165
15	160	162	158	153	...

S2 =

	0	1	2	3	4
0	88	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	132	122	117	118	118
5	150	147	148	148	150
6	169	164	167	170	173
7	193	195	193	189	183
8	191	192	194	199	194
9	187	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	183	185	187	186	180
14	174	176	174	166	165
15	160	162	158	153	...

З представлених даних видно, що, наприклад, значення різниць

$$b = R_{1,0} - R_{1,1},\ b = R_{2,0} - R_{2,1},\ b = R_{3,0} - R_{3,1}$$

повністю ідентичні відповідним значенням різниць

$$b = S2_{1,0} - S2_{1,1},\ b = S2_{2,0} - S2_{2,1},\ b = S2_{3,0} - S2_{3,1}.$$

Практично це означає, що значення відповідних бітів з другого рядка таблиці квантування збіглося в цих позиціях зі значенням вбудованих інформаційних бітів даних. Навпаки, значення різниць

$$b = R_{0,0} - R_{0,1},\ b = R_{4,0} - R_{4,1},\ b = R_{5,0} - R_{5,1}$$

відрізняються від відповідних значень різниць

$$b = S2_{0,0} - S2_{0,1},\ b = S2_{4,0} - S2_{4,1},\ b = S2_{5,0} - S2_{5,1}.$$

Це означає зміну поточної різниці відповідно до знайдених значень у таблиці квантування. Так, наприклад, значення різниці

$$b = R_{5,0} - R_{5,1} = 147 - 147 = 0$$

було змінено на значення різниці

$$b = S2_{5,0} - S2_{5,1} = 150 - 147 = 3.$$

Як видно з наведеної вище таблиці квантування d, для значення різниці

$$b = d_{0,255} = 0$$

відповідне значення з другого рядка дорівнює

$$d_{1,255} = 1.$$

Для вбудовування інформаційного біта зі значенням «0» за правилом «пошук праворуч» значення різниці модифікується на найближче знайдене праворуч значення, для якого значення з другого рядка таблиці квантування і значення вбудованого біта співпадуть. Очевидно, що це

$$d_{1,258} = 0,$$

і маємо відповідне значення різниці

$$b = d_{0,258} = 3,$$

що повністю підтверджує правильність роботи алгоритму вбудовування.

Слід зазначити, що в запропонованій реалізації модифікація різниці досягається лише зміною елемента контейнера в нульовому стовпці, тобто за рахунок модифікації значень $S2_{i,0}$. Практично це означає, що всі спотворення будуть зосереджені в одному стовпці. Абсолютне значення внесених спотворень визначається статистичними властивостями використовуваної як секретний ключ псевдовипадкової послідовності, тобто другого рядка таблиці квантування. Для ефективних криптографічних генераторів з рівноймовірним розподілом формованих значень внесені спотворення знаходяться нижче порога зорової чутливості людини.

Графічну інтерпретацію порожнього і заповненого контейнера (каналу червоного кольору в градаціях сірого) наведено на наступному рисунку, з якого слідує, що візуально внесені спотворення не помітні, що підтверджує висновок про чутливість органів зору людини до незначної зміни контрастності.

R

S2

Отриманий заповнений масив S2 записуємо в канал червоного кольору контейнера. Виконуємо команду

«WRITERGB("Stego_Kvant.bmp"):=augment(S2,G,B)».

В результаті виконання команди система MathCAD формує на фізичному носії новий файл з ім'ям «Stego_ Kvant.bmp».

Для графічного відображення вихідного (порожнього) і заповненого контейнера виконаємо вставку відповідних зображень:

"1.bmp"

"Stego_Kvant.bmp"

Переконаємося у відсутності видимих спотворень.

2.4. Реалізуємо алгоритм вилучення даних з просторової області зображень методом квантування. Для цього в тому ж вікні середовища MathCAD виконуємо команди читання растрових даних нерухомого зображення з заданого файла (файла заповненого контейнера) у вигляді двовимірного масиву цілих чисел. Для розглянутого прикладу виконуємо команди:

«C2:=READRGB("Stego_Kvant.bmp")»,
«R2:=READ_RED("Stego_Kvant.bmp")»,
«G2:=READ_GREEN("Stego_Kvant.bmp")»,
«B2:=READ_BLUE("Stego_Kvant.bmp")».

Отримаємо такий результат:

R2 =

	0	1	2	3	4
0	88	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	132	122	117	118	118
5	150	147	148	148	150
6	169	164	167	170	173
7	193	195	193	189	...

R2

Далі обчислюємо біти значення різниці b між елементами перших двох стовпців і знаходимо відповідне бітове значення з другого рядка таблиці квантування. Для цього використовуємо таку процедуру:

$$\text{M\_b2} := \left| \begin{array}{l} \text{for } i \in 0..\text{rows}(R2) - 1 \\ \quad \left| \begin{array}{l} b \leftarrow R2_{i,0} - R2_{i,1} \\ \text{M\_b2}_i \leftarrow d_{1,b+255} \end{array} \right. \\ \text{M\_b2} \end{array} \right.$$

Дана процедура виконує для всіх рядків масиву R2 обчислення значення різниці і відповідного йому біта даних, який і є вбудованим інформаційним бітом. В результаті маємо лінійний бітовий масив M_b2, заповнений відповідними бітами з другого рядка таблиці квантування.

M_b2 =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

Порівняння даних масивів вбудованих і витягнутих (отриманих) бітів даних дозволяє підтвердити правильність виконання алгоритмів вбудовування-вилучення.

2.5, 2.6. Формування масиву цілих чисел, що відповідають ASCII кодуванню вбудованих символів повідомлення і запис у текстовий файл витягнутих (отриманих) інформаційних даних здійснюється аналогічно п. 1.5, 1.6.

Завдання 3. Реалізація алгоритмів вбудовування та вилучення повідомлень методом Куттера–Джордана–Боссена (методом «хреста»)

3.1, 3.2. Завантажуємо вихідні дані і перетворюємо масив інформаційних даних (згідно п. 1.1, 1.2).

3.3. Реалізуємо алгоритм вбудовування даних у просторову область зображень методом квантування. Для цього спочатку реалізуємо функцію обчислення яскравості окремого пікселя із заданими координатами:

$$\lambda(x, y) := 0.29890 \cdot R_{x, y} + 0.58662 \cdot G_{x, y} + 0.11448 \cdot B_{x, y}$$

і встановлюємо параметри методу квантування:

$$\gamma := 0.05 \quad \sigma := 3,$$

а також визначаємо функцію модифікації окремого пікселя в такий спосіб:

$$SV(x, y, b) := \text{round}\left[B_{x, y} + (2 \cdot b - 1) \cdot \gamma \cdot \lambda(x, y)\right].$$

Зрозуміло, що значення $\lambda(x, y)$ визначається як повнокольорова яскравість пікселя за значеннями яскравості трьох колірних компонент з відповідним ваговим коефіцієнтом. Обрані параметри $\gamma = 0{,}05$ (енергія вбудованого біта) і $\sigma = 3$ (розмір ділянки передбачення) є найпростішими показниками надійної роботи стеганоалгоритма.

Функція вбудовування $SV(x, y, b)$ полягає в модифікації яскравості синього кольору заданого пікселя на частку його повноколірної яскравості, що задається параметром γ.

Для прикладу покажемо правильність роботи функції вбудовування.

Значення яскравості пікселя з координатами (7,7) відповідає

$$\lambda(7,7) = 176{,}42.$$

Відповідне значення яскравості синього кольору пікселя

$$B(7,7) = 208.$$

Можлива модифікація яскравості синього кольору пікселя згідно з функцією вбудовування приймає значення $208 \pm [0{,}05 \cdot 176{,}42] = 208 \pm 9$. Зрозуміло, що алгоритм обчислення функції $SV(7,7,0) = 199$ працює правильно.

Для вбудовування масиву інформаційних даних M_b скористаємося наступною процедурою. Вона реалізує поелементне вбудовування бітового масиву в значення яскравостей синього кольору за допомогою модифікації функцією вбудовування SV(x,y,b).

Ділянкою для вбудовування обрано діагональ масиву яскравостей синього кольору контейнера. Інші елементи контейнера (не з ділянки модифікації) підлягають перезапису з порожнього контейнера.

```
S3 := | for i ∈ 0.. cols(B) − 1
      |   for j ∈ 0.. rows(B) − 1
      |     S3[j,i] ← B[j,i]
      | for i ∈ σ.. rows(B) − σ − 1
      |   b ← SV(i, i, M_b[i−σ])
      |   S3[i,i] ← b   if 0 ≤ b ≤ 255
      |   S3[i,i] ← 255 if b > 255
      |   S3[i,i] ← 0   if b < 0
      | S3
```

Слід зазначити, що вбудовування починається не з пікселя з координатами (0, 0), а з пікселя, що має координати (σ, σ). Це виконано для можливості в подальшому здійснити передбачення методом «хреста».

Для візуального перегляду результату вбудовування інформаційних даних виведемо вихідний масив растрових даних синього кольору B і отриманий масив S2 зі зміненими значеннями різниць. Для розглянутого прикладу маємо:

S3 =

	0	1	2	3	4	5
0	135	128	123	122	119	124
1	155	142	141	134	133	122
2	174	159	151	152	141	136
3	162	160	151	151	147	147
4	172	161	159	164	164	160
5	184	181	190	184	183	183
6	198	197	200	203	206	207
7	225	222	226	222	218	206
8	223	226	222	224	219	215
9	220	221	230	229	224	221
10	224	228	231	225	229	221
11	221	217	227	231	232	233
12	222	224	228	230	231	231
13	215	211	218	217	211	208
14	209	207	204	198	197	197
15	200	196	192	190	189	...

B =

	0	1	2	3	4	5
0	135	128	123	122	119	124
1	155	142	141	134	133	122
2	174	159	151	152	141	136
3	162	160	151	145	147	147
4	172	161	159	164	158	160
5	184	181	190	184	183	191
6	198	197	200	203	206	207
7	225	222	226	222	218	206
8	223	226	222	224	219	215
9	220	221	230	229	224	221
10	224	228	231	225	229	221
11	221	217	227	231	232	233
12	222	224	228	230	231	231
13	215	211	218	217	211	208
14	209	207	204	198	197	197
15	200	196	192	190	189	...

З представлених даних видно, що, наприклад, значення

$$S3_{3,3} > B_{3,3},\ S3_{4,4} > B_{4,4}$$

відповідає вбудовуванню «1» в ці позиції.

Значення $S3_{5,5} < B_{5,5}$ відповідає вбудовуванню «0».

Слід зазначити, що величина внесених спотворень визначається введеним значенням $\gamma = 0{,}05$ (енергія вбудованого біта), як частки повноколірної яскравості пікселя, що припадає на модифікацію яскравості синього кольору.

Графічну інтерпретацію порожнього і заповненого контейнерів (каналів синього кольору в градаціях сірого) наведено на наступному рисунку, з якого видно, що візуально внесені спотворення не помітні, що підтверджує висновок про чутливість органів зору людини до незначної зміни синього кольору.

S3

B

Отриманий заповнений масив S3 записуємо в канал синього кольору контейнера. Виконуємо команду

«WRITERGB("Stego_Krest.bmp"):=augment(R,G,S3)».

В результаті виконання команди система MathCAD формує на фізичному носії новий файл з ім’ям «Stego_Krest.bmp».

Для графічного відображення вихідного (порожнього) і заповненого контейнера виконаємо вставку відповідних зображень:

"1.bmp"

"Stego_Krest.bmp"

Переконаємося у відсутності видимих спотворень.

3.4. Реалізуємо алгоритм вилучення даних з просторової області зображень методом «хреста». Для цього в тому ж вікні середовища MathCAD виконуємо команди читання растрових даних нерухомого зображення з заданого файла (файла заповненого контейнера) у вигляді двовимірного масиву цілих чисел. Для розглянутого прикладу виконуємо команди:

«C3:=READRGB(“Stego_Krest.bmp ”)»,

«R3:=READ_RED(“Stego_Krest.bmp”)»,
«G3:=READ_GREEN(“Stego_Krest.bmp”)»,
«B3:=READ_BLUE(“Stego_Krest.bmp”)».

Отримуємо такий результат:

B3 =

	0	1	2	3	4
0	135	128	123	122	119
1	155	142	141	134	133
2	174	159	151	152	141
3	162	160	151	151	147
4	172	161	159	164	164
5	184	181	190	184	183
6	198	197	200	203	206
7	225	222	226	222	...

B3

Далі, для кожного вбудованого біта інформації обчислюємо передбачене значення b синього кольору і порівнюємо зі стостережуваним значенням $B3_{i,i}$. Використовуємо таку процедуру:

$$M\_b3 := \left| \begin{array}{l} \text{for } i \in \sigma .. \text{rows}(B3) - \sigma - 1 \\ \quad b \leftarrow \dfrac{\left(\sum_{j=i-\sigma}^{i-1} B3_{i,j} + \sum_{j=i-\sigma}^{i-1} B3_{j,i} + \sum_{j=i+1}^{i+\sigma} B3_{i,j} + \sum_{j=i+1}^{i+\sigma} B3_{j,i} \right)}{4\sigma} \\ \quad M\_b3_{i-\sigma} \leftarrow 1 \text{ if } b < B3_{i,i} \\ \quad M\_b3_{i-\sigma} \leftarrow 0 \text{ if } b > B3_{i,i} \\ M\_b3 \end{array} \right.$$

M_b3 =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

В результаті порівняння b зі спостережуваним значенням $B3_{i,i}$ приймаємо рішення про значення вбудованого біта інформації. Виконання передбачення про значення яскравості синього кольору виконуємо для кожного пікселя, що підлягав модифікації. Позиція (координати) модифікованого пікселя є секретною ключовою інформацією.

Наведемо результат роботи цієї процедури вилучення даних для розглянутого прикладу.

Зрозуміло, що результат вилучення перших трьох бітів є неправильним. Настання такої події не виключено логікою алгоритму вилучення, ймовірність її ви-

никнення визначається статистичними властивостями контейнера. Підвищити ймовірність правильного вилучення інформаційних бітів даних можна за рахунок підвищення енергії вбудованих бітів даних, тобто за допомогою збільшення коефіцієнта γ. Однак подібна процедура неминуче призведе до збільшення внесених спотворень у контейнер-зображення.

3.5, 3.6. Формування масиву цілих чисел, що відповідають ASCII кодуванню вбудованих символів повідомлення і запис у текстовий файл отриманих інформаційних даних здійснюється аналогічно п. 1.5, 1.6.

Завдання 4. Дослідження ймовірнісних характеристик стеганографічного методу вбудовування даних Куттера–Джордана–Боссена (методу «хреста»)

4.1. Проведемо оцінку ймовірності правильного вилучення повідомлення і величини внесених спотворень від коефіцієнта γ. Для цього будемо послідовно збільшувати величину γ і для кожного значення розраховувати частоту v правильно отриманих інформаційних бітів. Одночасно будемо розраховувати усереднену величину w внесених спотворень, виражену у відсотковому співвідношенні до максимального значення яскравості. Використаємо для цього такі процедури:

$$v := \left| \begin{array}{l} v \leftarrow 0 \\ \text{for } i \in 0..\text{rows}(M\_b3) - 1 \\ \quad v \leftarrow v + 1 \text{ if } M\_b3_i = M\_b_i \\ v \leftarrow \dfrac{v}{\text{rows}(M\_b3)} \\ v \end{array} \right. \qquad w := \left| \begin{array}{l} w \leftarrow 0 \\ \text{for } i \in \sigma..\text{rows}(B3) - \sigma - 1 \\ \quad w \leftarrow w + \left| B3_{i,i} - B_{i,i} \right| \\ w \leftarrow \dfrac{w \cdot 100}{\text{rows}(M\_b3) \cdot 256} \\ w \end{array} \right.$$

Для розглянутого прикладу при γ = 0,45 маємо такі значення:

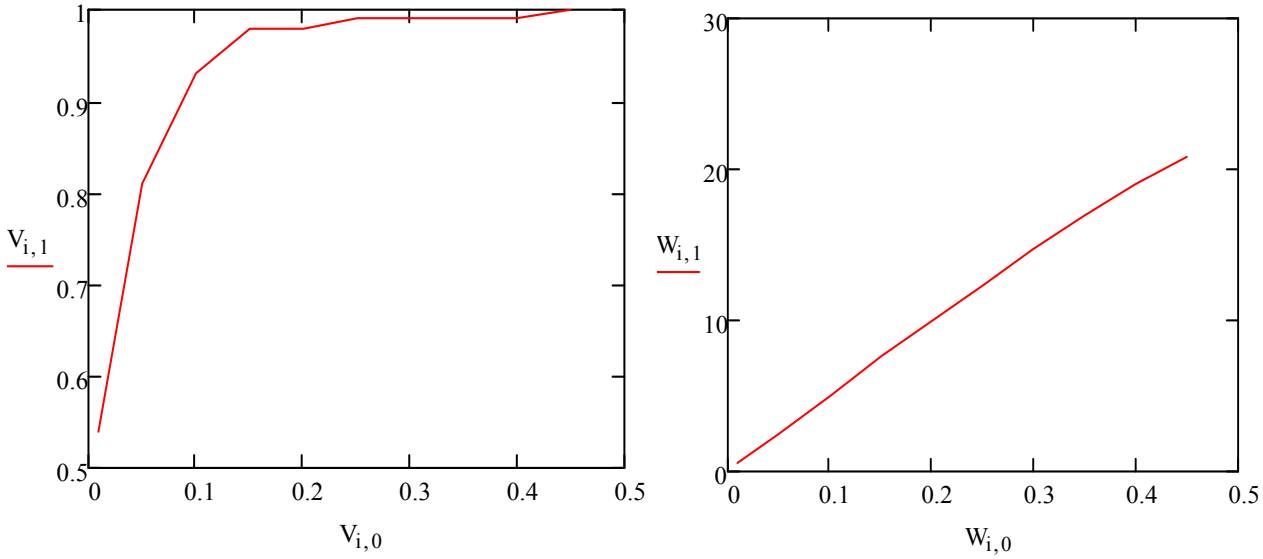
$$v = 1 \qquad w = 20.809$$

Отримані емпіричні дані занесемо до відповідних таблиць:

$$V := \begin{pmatrix} 0.01 & 0.54 \\ 0.05 & 0.81 \\ 0.1 & 0.93 \\ 0.15 & 0.98 \\ 0.2 & 0.98 \\ 0.25 & 0.99 \\ 0.3 & 0.99 \\ 0.35 & 0.99 \\ 0.4 & 0.99 \\ 0.45 & 1 \end{pmatrix} \qquad W := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix}$$

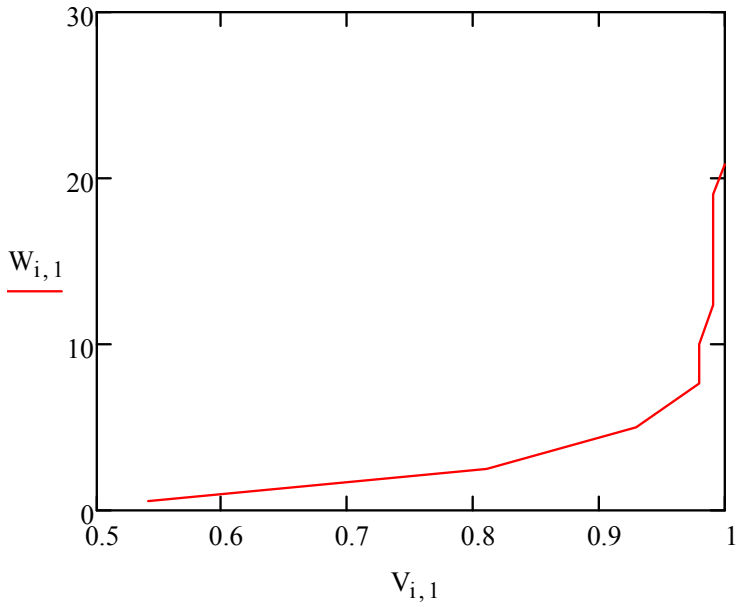
4.2. Побудуємо графіки отриманих емпіричних залежностей:

$$i := 0..9$$



Зрозуміло, що величина внесених спотворень зростає лінійно від коефіцієнта γ. Однак емпірична залежність ймовірності правильного вилучення інформаційних даних поводиться інакше. При малих значеннях коефіцієнта γ величина W зростає швидко, однак при γ> 0,2 подальше збільшення енергії вбудовування не призводить до суттєвого підвищення ймовірності правильного вилучення, підвищувати величину V в даному випадку недоцільно.

4.3. Побудуємо інтегральний графік залежності величини W внесених спотворень у контейнер-зображення при забезпеченні відповідної ймовірності V правильного вилучення інформаційних даних:



Зрозуміло, що ефективне приховування вбудованих інформаційних даних без внесення значних спотворень (W <5 %) в контейнер-зображення буде спостерігатися тільки при ймовірності правильного вилучення даних

V<0,8 ... 0,9, що відповідає енергії вбудовування γ = 0,05 ... 0,15. Підвищення достовірності отриманих даних за рахунок подальшого збільшення енергії вбудовування є недоцільним, оскільки це призводить до внесення невиправдано високих спотворень у контейнер-зображення. В даному випадку найбільш перспективною є реалізація завадостійкого кодування інформаційних даних і контроль помилок, що виникають при стеганографічних перетвореннях.

Завдання 5 (додаткове). Реалізація завадостійкого кодування інформаційних даних для підвищення ймовірнісних характеристик стеганографічного методу вбудовування даних Куттера–Джордана–Боссена (методу «хреста»)

5.1. Реалізуємо завадостійке кодування найпростішим лінійним блоковим кодом Хеммінга. Для цього введемо такі породжуючі та перевірочну матриці:

$$Gen := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \qquad H := \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

5.2. Реалізуємо алгоритми кодування і декодування окремих кодових слів. Для цього скористаємося такими функціями:

```
cod(inf) := | for i ∈ 0..6
            |     c_i ← 0
            |     for j ∈ 0..3
            |         c_i ← (c_i) ⊕ (inf_j · Gen_j,i)
            | c
```

```
decod(c) := | for i ∈ 0..2
            |     s_i ← 0
            |     for j ∈ 0..6
            |         s_i ← (s_i) ⊕ (c_j · H_i,j)
            | ss ← s_0 + s_1·2 + s_2·4
            | cc ← c
            | cc_4 ← (c_4) ⊕ 1  if ss = 1
            | cc_5 ← (c_5) ⊕ 1  if ss = 2
            | cc_2 ← (c_2) ⊕ 1  if ss = 3
            | cc_6 ← (c_6) ⊕ 1  if ss = 4
            | cc_0 ← (c_0) ⊕ 1  if ss = 5
            | cc_3 ← (c_3) ⊕ 1  if ss = 6
            | cc_1 ← (c_1) ⊕ 1  if ss = 7
            | cc
```

Як приклад розглянемо інформаційний вектор

$$\text{inf} := \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Після виконання функції кодування маємо наступне кодове слово

$$c := \text{cod}(\text{inf}) \qquad c = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Внесемо помилку в довільному кодовому символі, наприклад, $c_3 := 1$

Маємо наступне слово з помилкою, яке після декодування відновлюється в безпомилкову послідовність:

$$c = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \qquad \text{decod}(c) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

5.3. Реалізуємо алгоритм завадостійкого кодування масиву інформаційних даних:

$$\text{M\_b\_cod} := \left| \begin{array}{l} \text{for } i \in 0..\, \text{ceil}\left(\frac{\text{rows}(\text{M\_b})}{4}\right) - 1 \\ \quad \left| \begin{array}{l} \text{for } j \in 0..3 \\ \quad \text{inf}_j \leftarrow \text{M\_b}_{4\cdot i+j} \\ c \leftarrow \text{cod}(\text{inf}) \\ \text{for } l \in 0..6 \\ \quad \text{M\_b\_cod}_{7\cdot i+l} \leftarrow c_l \end{array} \right. \\ \text{M\_b\_cod} \end{array} \right.$$

Для розглянутого прикладу маємо:

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b_cod =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

Реалізована процедура зчитує по чотири біти з масиву M_b і кодує їх завадостійким кодом Хеммінга. Результат кодування блоками по сім бітів записується в масив M_b_cod.

5.4. Реалізуємо вбудовування сформованих даних у контейнер-зображення методом «хреста». Для цього скористаємося розглянутими в п. 3.3 процедурами:

$\gamma := 0.05$

$SV(x, y, b) := \text{round}\left[B_{x,y} + (2 \cdot b - 1) \cdot \gamma \cdot \lambda(x, y)\right]$

```
S4 := | for i ∈ 0.. cols(B) − 1
      |   for j ∈ 0.. rows(B) − 1
      |     S4[j,i] ← B[j,i]
      | for i ∈ σ.. rows(B) − σ − 1
      |   b ← SV(i, i, M_b_cod[i−σ])
      |   S4[i,i] ← b   if 0 ≤ b ≤ 255
      |   S4[i,i] ← 255 if b > 255
      |   S4[i,i] ← 0   if b < 0
      | S4
```

В результаті формуємо масив S4 синього кольору з вбудованими даними. Сформуємо заповнений контейнер і переглянемо результат:

WRITERGB("Stego_Krest_cod.bmp") := augment(R, G, S4)

S4

B

"1.bmp"

"Stego_Krest_cod.bmp"

5.5. Реалізуємо витяг сформованих даних у контейнер-зображення методом «хреста». Для цього скористаємося розглянутими в п. 3.4 процедурами:

B4 := READ_BLUE("Stego_Krest_cod.bmp")

B4 =

	0	1	2	3	4
0	135	128	123	122	119
1	155	142	141	134	133
2	174	159	151	152	141
3	162	160	151	151	147
4	172	161	159	164	164
5	184	181	190	184	183
6	198	197	200	203	206
7	225	222	226	222	...

B4

$$M\_b4 := \text{for } i \in \sigma .. \text{rows}(B4) - \sigma - 1$$

$$b \leftarrow \frac{\left(\sum_{j=i-\sigma}^{i-1} B4_{i,j} + \sum_{j=i-\sigma}^{i-1} B4_{j,i} + \sum_{j=i+1}^{i+\sigma} B4_{i,j} + \sum_{j=i+1}^{i+\sigma} B4_{j,i} \right)}{4\sigma}$$

$$M\_b4_{i-\sigma} \leftarrow 1 \text{ if } b < B4_{i,i}$$

$$M\_b4_{i-\sigma} \leftarrow 0 \text{ if } b > B4_{i,i}$$

M_b4

В результаті виконання розглянутих процедур формуємо масив витягнутих даних:

M_b4 =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

M_b_cod =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

5.6. Реалізуємо завадостійке декодування витягнутих даних:

$$
\begin{aligned}
&\text{M\_b\_decod} := \Big| \ \text{for } i \in 0..\text{floor}\left(\frac{\text{rows}(\text{M\_b4})}{7}\right) - 1 \\
&\quad \Big| \ \text{for } j \in 0..6 \\
&\quad\quad c_j \leftarrow \text{M\_b4}_{7\cdot i+j} \\
&\quad c \leftarrow \text{decod}(c) \\
&\quad \text{for } l \in 0..3 \\
&\quad\quad \text{M\_b\_decod}_{4\cdot i+l} \leftarrow c_l \\
&\text{M\_b\_decod}
\end{aligned}
$$

Наведена процедура зчитує витягнуті дані блоками по сім бітів і декодує їх розглянутою в п. 5.2 функцією. В результаті формуємо масив витягнутих даних з виправленими помилками.

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b_decod =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	1
13	1
14	0
15	...

5.7. Дослідимо ймовірнісні характеристики методу «хреста» з використанням завадостійкого кодування. Для цього скористаємося розглянутими в п. 4.1–4.3 процедурами:

$$v := \left| \begin{array}{l} v \leftarrow 0 \\ \text{for } i \in 0..\text{rows}(M\_b\_decod) - 1 \\ \quad v \leftarrow v + 1 \text{ if } M\_b\_decod_i = M\_b_i \\ v \leftarrow \dfrac{v}{\text{rows}(M\_b\_decod)} \\ v \end{array} \right.$$

$v = 0.804$

$$w := \left| \begin{array}{l} w \leftarrow 0 \\ \text{for } i \in \sigma..\text{rows}(B4) - \sigma - 1 \\ \quad w \leftarrow w + \left| B4_{i,i} - B_{i,i} \right| \\ w \leftarrow \dfrac{w \cdot 100}{\text{rows}(M\_b3) \cdot 256} \\ w \end{array} \right.$$

$w = 2.55$

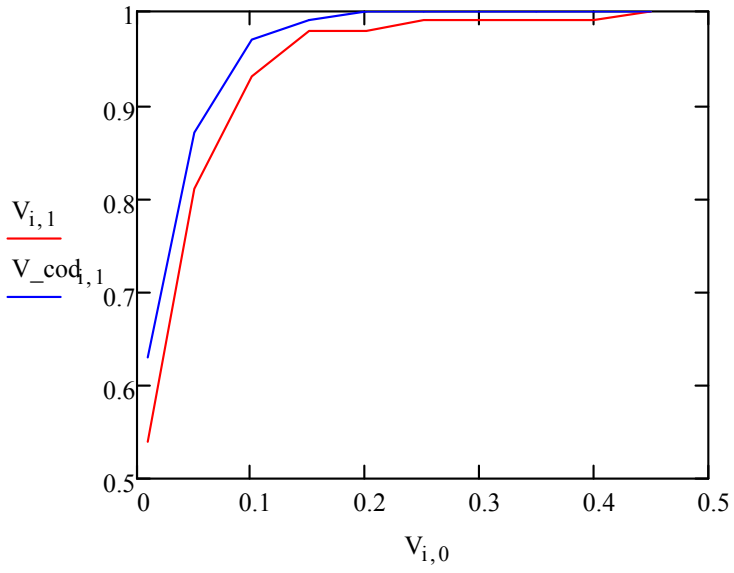
Отримані емпіричні дані занесемо до відповідних таблиць і порівняємо з вже наявними залежностями:

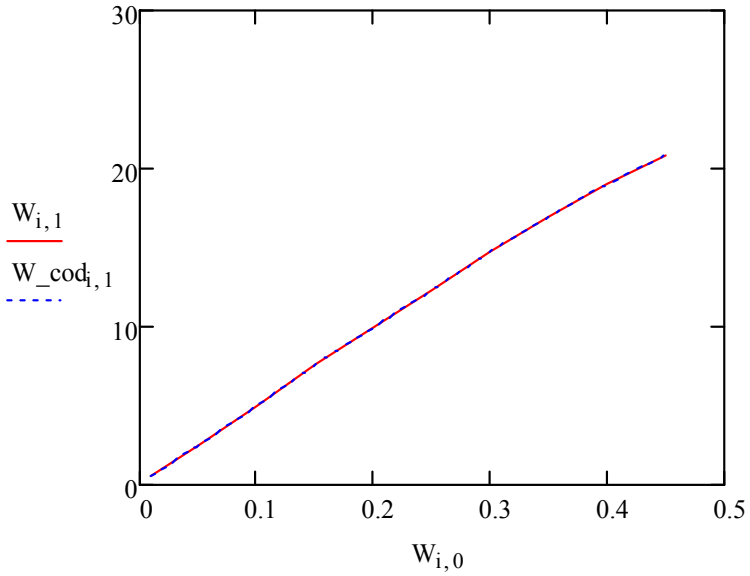
$$V := \begin{pmatrix} 0.01 & 0.54 \\ 0.05 & 0.81 \\ 0.1 & 0.93 \\ 0.15 & 0.98 \\ 0.2 & 0.98 \\ 0.25 & 0.99 \\ 0.3 & 0.99 \\ 0.35 & 0.99 \\ 0.4 & 0.99 \\ 0.45 & 1 \end{pmatrix} \quad V\_cod := \begin{pmatrix} 0.01 & 0.63 \\ 0.05 & 0.87 \\ 0.1 & 0.97 \\ 0.15 & 0.99 \\ 0.2 & 1 \\ 0.25 & 1 \\ 0.3 & 1 \\ 0.35 & 1 \\ 0.4 & 1 \\ 0.45 & 1 \end{pmatrix} \quad W := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix} \quad W\_cod := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix}$$

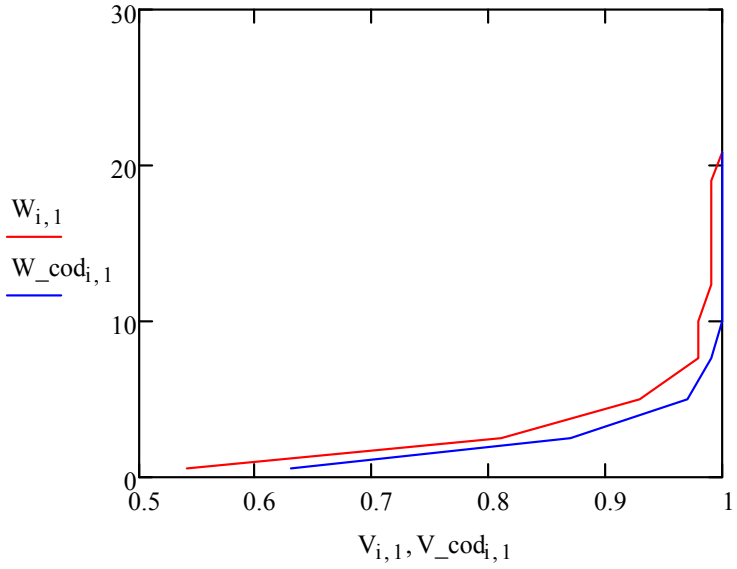
Зрозуміло, що використання завадостійкого кодування дозволило незначно підвищити ймовірність правильного вилучення вбудованих даних.

Побудуємо відповідні графіки:

$$i := 0 .. 9$$







Отримані залежності показують, що використання навіть найпростішого завадостійкого коду Хеммінга (синя крива) дозволяє поліпшити ймовірнісні характеристики методу «хреста». Рівень внесених спотворень при цьому не змінюється. У той же час використання коду Хеммінга призвело до зниження практично в два рази обсягу вбудованих інформаційних даних. Для практичного використання доцільно застосування потужних завадостійких кодів, що дозволять домогтися безпомилкового вилучення при незначному зниженні обсягу вбудованих інформаційних даних.

6. ПРИКЛАД ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНОЇ РОБОТИ

Лабораторна робота № 2

Приховування даних у просторовій області зображень методом блокового приховування, методом квантування, методом «хреста»

"1.bmp"

C :=READRGB("1.bmp")
R := READ_RED("1.bmp")
G := READ_GREEN("1.bmp")
B := READ_BLUE("1.bmp")
M := READBIN("2.txt", "byte")

R

R =

	0	1	2	3	4	5	6	7
0	86	79	72	72	72	69	71	74
1	110	97	90	86	78	71	70	70
2	132	120	112	105	96	88	81	83
3	122	116	105	104	103	102	101	99
4	131	122	117	118	118	116	105	107
5	147	147	148	148	150	153	145	135
6	169	164	167	170	173	175	164	155
7	189	195	193	189	183	172	173	173
8	191	192	194	199	194	187	182	182
9	186	188	194	198	192	187	187	180
10	195	196	199	200	201	190	192	186
11	185	189	202	203	203	199	203	199
12	192	196	198	199	204	202	206	199
13	177	185	187	186	180	178	179	177
14	173	176	174	166	165	165	163	161
15	160	162	158	153	156	158	157	...

$$B\_D(x) := \sum_{i=0}^{7} \left(x_i \cdot 2^i \right)$$

$$D\_B(x) := \left| \begin{array}{l} \text{for } i \in 0..7 \\ \quad \left| \begin{array}{l} V_i \leftarrow \text{mod}(x, 2) \\ x \leftarrow \text{floor}\left(\frac{x}{2}\right) \end{array} \right. \\ V \end{array} \right.$$

$$M\_b := \left| \begin{array}{l} \text{for } i \in 0..\text{rows}(M) - 1 \\ \quad \left| \begin{array}{l} V \leftarrow D\_B\left(M_i\right) \\ \text{for } j \in 0..7 \\ \quad M\_b_{i \cdot 8 + j} \leftarrow V_j \end{array} \right. \\ M\_b \end{array} \right.$$

```
S1 := | for i ∈ 0.. cols(R) − 1
      |   b ← mod( Σ_{j=0}^{rows(R)−1} R_{j,i}, 2 )
      |   if M_b_i ≠ b
      |     P ← D_B(R_{0,i})
      |     P_0 ← P_0 ⊕ 1
      |     S1_{0,i} ← B_D(P)
      |   S1_{0,i} ← R_{0,i}  if M_b_i = b
      |   for j ∈ 1.. rows(R) − 1
      |     S1_{j,i} ← R_{j,i}
      | S1
```

M =

	0
0	203
1	224
2	225
3	238
4	240
5	224
6	242
7	238
8	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

WRITERGB("Stego_Blok.bmp") := augment (S1, G, B)

S1 =

	0	1	2	3	4
0	87	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	183
8	191	192	194	199	194
9	186	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	177	185	187	186	180
14	173	176	174	166	165
15	160	162	158	153	...

R =

	0	1	2	3	4
0	86	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	183
8	191	192	194	199	194
9	186	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	177	185	187	186	180
14	173	176	174	166	165
15	160	162	158	153	...

S1

R

"Stego_Blok.bmp"

"1.bmp"

C1 := READRGB("Stego_Blok.bmp")

R1 := READ_RED ("Stego_Blok.bmp")

G1 := READ_GREEN ("Stego_Blok.bmp")

B1 := READ_BLUE("Stego_Blok.bmp")

R1 =

	0	1	2	3	4
0	87	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	131	122	117	118	118
5	147	147	148	148	150
6	169	164	167	170	173
7	189	195	193	189	...

R1

$$M\_b1 := \left| \begin{array}{l} \text{for } i \in 0..\text{cols}(R1) - 1 \\ \quad M\_b1_i \leftarrow \text{mod}\left(\sum_{j=0}^{\text{rows}(R)-1} R1_{j,i}, 2 \right) \\ M\_b1 \end{array} \right.$$

M_b1 =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	...

Метод квантування

```
d := for i ∈ 0..510
       d_{0,i} ← i − 255
       d_{1,i} ← ceil(rnd(2)) − 1
     d
```

d =	250	251	252	253	254	255	256	257	258
0	-5	-4	-3	-2	-1	0	1	2	...

```
S2 := for i ∈ 0.. rows(R) − 1
        b ← R_{i,0} − R_{i,1}
        S2_{i,0} ← R_{i,0}  if M_b_i = d_{1,b+255}
        if M_b_i ≠ d_{1,b+255}
          j ← 1
          while M_b_i ≠ d_{1,b+255+j} ∧ j < 509
            j ← j + 1
          S2_{i,0} ← R_{i,0} + d_{0,b+255+j} − b
        for j ∈ 1.. cols(R) − 1
          S2_{i,j} ← R_{i,j}
      S2
```

S2 =

	0	1	2	3	4
0	88	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	132	122	117	118	118
5	150	147	148	148	150
6	169	164	167	170	173
7	193	195	193	189	183
8	191	192	194	199	194
9	187	188	194	198	192
10	195	196	199	200	201
11	185	189	202	203	203
12	192	196	198	199	204
13	183	185	187	186	180
14	174	176	174	166	165
15	160	162	158	153	...

WRITERGB("Stego_Kvant.bmp") := augment (S2, G, B)

S2

R

"1.bmp"

"Stego_Kvant.bmp"

C2 := READRGB("Stego_Kvant.bmp")

R2 := READ_RED("Stego_Kvant.bmp")

G2 := READ_GREEN("Stego_Kvant.bmp")

B2 := READ_BLUE("Stego_Kvant.bmp")

R2

R2 =

	0	1	2	3	4
0	88	79	72	72	72
1	110	97	90	86	78
2	132	120	112	105	96
3	122	116	105	104	103
4	132	122	117	118	118
5	150	147	148	148	150
6	169	164	167	170	173
7	193	195	193	189	...

$$M\_b2 := \left| \begin{array}{l} \text{for } i \in 0..\text{rows}(R2) - 1 \\ \quad \left| \begin{array}{l} b \leftarrow R2_{i,0} - R2_{i,1} \\ M\_b2_i \leftarrow d_{1,b+255} \end{array} \right. \\ M\_b2 \end{array} \right.$$

M_b2 =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

Метод «хреста»

$\lambda(x,y) := 0.29890 \cdot R_{x,y} + 0.58662 \cdot G_{x,y} + 0.11448 \cdot B_{x,y} \qquad \gamma := 0.05 \qquad \sigma := 3$

$SV(x,y,b) := \text{round}\left[B_{x,y} + (2 \cdot b - 1) \cdot \gamma \cdot \lambda(x,y)\right]$

Приклад: $\lambda(7,7) = 176.42 \qquad B_{7,7} = 208 \qquad SV(7,7,0) = 199$

```
S3 := | for i ∈ 0.. cols(B) − 1
      |   for j ∈ 0.. rows(B) − 1
      |     S3[j,i] ← B[j,i]
      | for i ∈ σ.. rows(B) − σ − 1
      |   b ← SV(i, i, M_b[i−σ])
      |   S3[i,i] ← b   if 0 ≤ b ≤ 255
      |   S3[i,i] ← 255 if b > 255
      |   S3[i,i] ← 0   if b < 0
      | S3
```

S3 =

	0	1	2	3	4	5
0	135	128	123	122	119	124
1	155	142	141	134	133	122
2	174	159	151	152	141	136
3	162	160	151	151	147	147
4	172	161	159	164	164	160
5	184	181	190	184	183	183
6	198	197	200	203	206	207
7	225	222	226	222	218	206
8	223	226	222	224	219	215
9	220	221	230	229	224	221
10	224	228	231	225	229	221
11	221	217	227	231	232	233
12	222	224	228	230	231	231
13	215	211	218	217	211	208
14	209	207	204	198	197	197
15	200	196	192	190	189	...

WRITERGB("Stego_Krest.bmp") := augment (R, G, S3)

S3

B

"1.bmp"

"Stego_Krest.bmp"

C3 := READRGB("Stego_Krest.bmp")

R3 := READ_RED("Stego_Krest.bmp")

G3 := READ_GREEN("Stego_Krest.bmp")

B3 := READ_BLUE("Stego_Krest.bmp")

B3 =

	0	1	2	3	4
0	135	128	123	122	119
1	155	142	141	134	133
2	174	159	151	152	141
3	162	160	151	151	147
4	172	161	159	164	164
5	184	181	190	184	183
6	198	197	200	203	206
7	225	222	226	222	...

B3

M_b3 :=

for $i \in \sigma .. \text{rows}(B3) - \sigma - 1$

$$b \leftarrow \frac{\left(\sum_{j=i-\sigma}^{i-1} B3_{i,j} + \sum_{j=i-\sigma}^{i-1} B3_{j,i} + \sum_{j=i+1}^{i+\sigma} B3_{i,j} + \sum_{j=i+1}^{i+\sigma} B3_{j,i}\right)}{4\sigma}$$

$M\_b3_{i-\sigma} \leftarrow 1$ if $b < B3_{i,i}$

$M\_b3_{i-\sigma} \leftarrow 0$ if $b > B3_{i,i}$

M_b3

Дослідження ймовірнісних характеристик

v :=

$v \leftarrow 0$

for $i \in 0.. \text{rows}(M\_b3) - 1$

$v \leftarrow v + 1$ if $M\_b3_i = M\_b_i$

$$v \leftarrow \frac{v}{\text{rows}(M\_b3)}$$

v

w :=

$w \leftarrow 0$

for $i \in \sigma .. \text{rows}(B3) - \sigma - 1$

$w \leftarrow w + \left|B3_{i,i} - B_{i,i}\right|$

$$w \leftarrow \frac{w \cdot 100}{\text{rows}(M\_b3) \cdot 256}$$

w

M_b3 =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

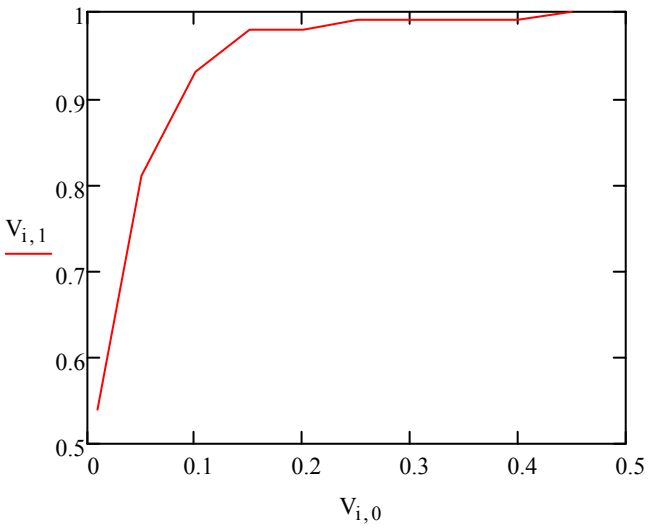
M_b =

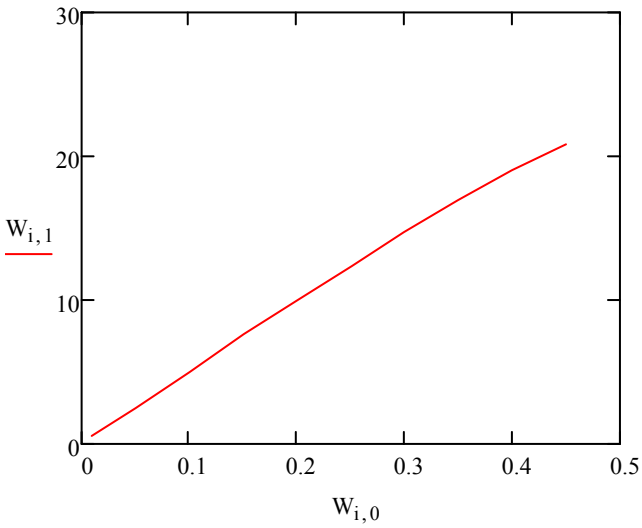
	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

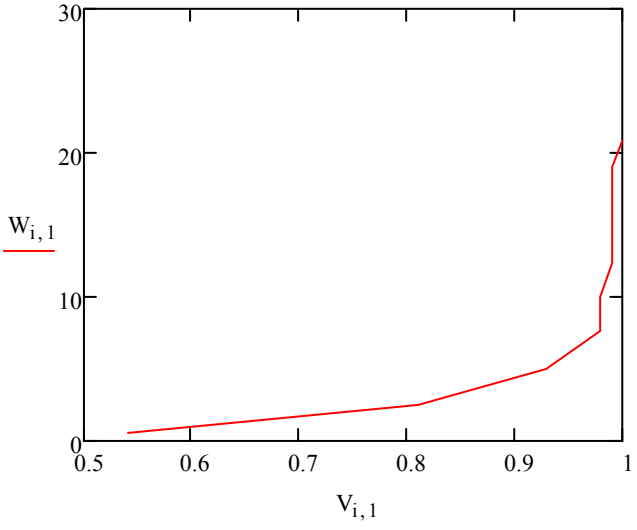
$$v = 0.81 \qquad w = 2.55$$

$$V := \begin{pmatrix} 0.01 & 0.54 \\ 0.05 & 0.81 \\ 0.1 & 0.93 \\ 0.15 & 0.98 \\ 0.2 & 0.98 \\ 0.25 & 0.99 \\ 0.3 & 0.99 \\ 0.35 & 0.99 \\ 0.4 & 0.99 \\ 0.45 & 1 \end{pmatrix} \qquad W := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix}$$

$$i := 0..9$$







Завадостійке кодування інформаційних даних

$$Gen := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \qquad H := \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$\text{cod}(\text{inf}) :=$
- for $i \in 0..6$
  - $c_i \leftarrow 0$
  - for $j \in 0..3$
    - $c_i \leftarrow (c_i) \oplus (\text{inf}_j \cdot \text{Gen}_{j,i})$
- c

$$\text{inf} := \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$c := \text{cod}(\text{inf})$

$$c = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$c_3 := 1$

$$c = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$\text{decod}(c) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$\text{decod}(c) :=$
- for $i \in 0..2$
  - $s_i \leftarrow 0$
  - for $j \in 0..6$
    - $s_i \leftarrow (s_i) \oplus (c_j \cdot H_{i,j})$
- $ss \leftarrow s_0 + s_1 \cdot 2 + s_2 \cdot 4$
- $cc \leftarrow c$
- $cc_4 \leftarrow (c_4) \oplus 1$ if $ss = 1$
- $cc_5 \leftarrow (c_5) \oplus 1$ if $ss = 2$
- $cc_2 \leftarrow (c_2) \oplus 1$ if $ss = 3$
- $cc_6 \leftarrow (c_6) \oplus 1$ if $ss = 4$
- $cc_0 \leftarrow (c_0) \oplus 1$ if $ss = 5$
- $cc_3 \leftarrow (c_3) \oplus 1$ if $ss = 6$
- $cc_1 \leftarrow (c_1) \oplus 1$ if $ss = 7$
- cc

```
M_b_cod := | for i ∈ 0.. ceil( rows(M_b) / 4 ) − 1
          |   for j ∈ 0..3
          |     inf_j ← M_b_{4·i+j}
          |   c ← cod(inf)
          |   for l ∈ 0..6
          |     M_b_cod_{7·i+l} ← c_l
          | M_b_cod
```

M_b=

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b_cod=

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

$\gamma := 0.05$

$SV(x,y,b) := \text{round}\left[B_{x,y} + (2\cdot b - 1)\cdot\gamma\cdot\lambda(x,y)\right]$

```
S4 := | for i ∈ 0.. cols(B) − 1
      |   for j ∈ 0.. rows(B) − 1
      |     S4_{j,i} ← B_{j,i}
      | for i ∈ σ.. rows(B) − σ − 1
      |   b ← SV(i, i, M_b_cod_{i−σ})
      |   S4_{i,i} ← b  if 0 ≤ b ≤ 255
      |   S4_{i,i} ← 255  if b > 255
      |   S4_{i,i} ← 0  if b < 0
      | S4
```

WRITERGB("Stego_Krest_cod.bmp") := augment(R, G, S4)

S4

B

"1.bmp"

"Stego_Krest_cod.bmp"

B4 := READ_BLUE("Stego_Krest_cod.bmp")

B4 =

	0	1	2	3	4
0	135	128	123	122	119
1	155	142	141	134	133
2	174	159	151	152	141
3	162	160	151	151	147
4	172	161	159	164	164
5	184	181	190	184	183
6	198	197	200	203	206
7	225	222	226	222	...

B4

M_b4 :=

for $i \in \sigma .. \text{rows}(B4) - \sigma - 1$

$$b \leftarrow \frac{\left(\sum_{j=i-\sigma}^{i-1} B4_{i,j} + \sum_{j=i-\sigma}^{i-1} B4_{j,i} + \sum_{j=i+1}^{i+\sigma} B4_{i,j} + \sum_{j=i+1}^{i+\sigma} B4_{j,i} \right)}{4\sigma}$$

$M\_b4_{i-\sigma} \leftarrow 1 \text{ if } b < B4_{i,i}$

$M\_b4_{i-\sigma} \leftarrow 0 \text{ if } b > B4_{i,i}$

M_b4

$$\text{M\_b\_decod} := \left| \begin{array}{l} \text{for } i \in 0..\text{floor}\left(\frac{\text{rows}(\text{M\_b4})}{7}\right) - 1 \\ \quad \left| \begin{array}{l} \text{for } j \in 0..6 \\ \quad c_j \leftarrow \text{M\_b4}_{7\cdot i+j} \\ c \leftarrow \text{decod}(c) \\ \text{for } l \in 0..3 \\ \quad \text{M\_b\_decod}_{4\cdot i+l} \leftarrow c_l \end{array} \right. \\ \text{M\_b\_decod} \end{array} \right.$$

M_b4 =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

M_b_cod =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	...

M_b =

	0
0	1
1	1
2	0
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	...

M_b_decod =

	0
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	1
8	0
9	0
10	0
11	0
12	1
13	1
14	0
15	...

Дослідження ймовірнісних характеристик

$$v := \left| \begin{array}{l} v \leftarrow 0 \\ \text{for } i \in 0..\text{rows}(M\_b\_decod) - 1 \\ \quad v \leftarrow v + 1 \text{ if } M\_b\_decod_i = M\_b_i \\ v \leftarrow \dfrac{v}{\text{rows}(M\_b\_decod)} \\ v \end{array} \right.$$

$$w := \left| \begin{array}{l} w \leftarrow 0 \\ \text{for } i \in \sigma..\text{rows}(B4) - \sigma - 1 \\ \quad w \leftarrow w + \left| B4_{i,i} - B_{i,i} \right| \\ w \leftarrow \dfrac{w \cdot 100}{\text{rows}(M\_b3) \cdot 256} \\ w \end{array} \right.$$

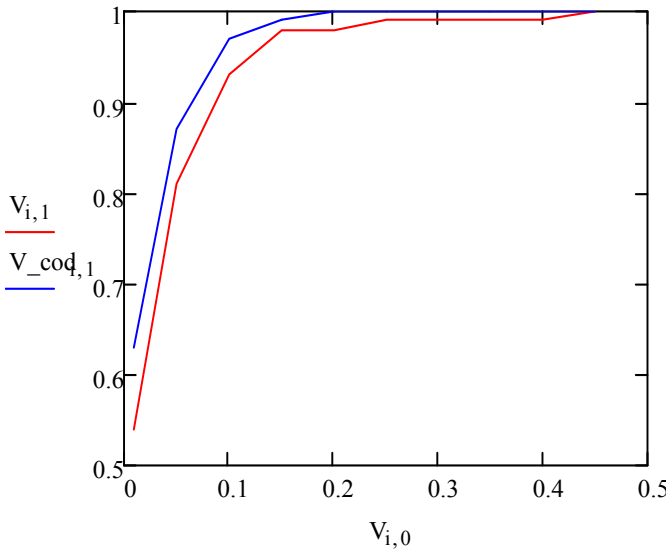
$v = 0.804$

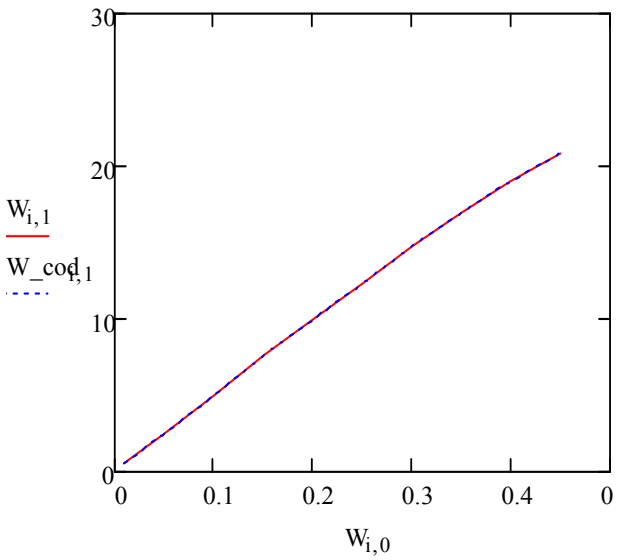
$$V := \begin{pmatrix} 0.01 & 0.54 \\ 0.05 & 0.81 \\ 0.1 & 0.93 \\ 0.15 & 0.98 \\ 0.2 & 0.98 \\ 0.25 & 0.99 \\ 0.3 & 0.99 \\ 0.35 & 0.99 \\ 0.4 & 0.99 \\ 0.45 & 1 \end{pmatrix} \quad V\_cod := \begin{pmatrix} 0.01 & 0.63 \\ 0.05 & 0.87 \\ 0.1 & 0.97 \\ 0.15 & 0.99 \\ 0.2 & 1 \\ 0.25 & 1 \\ 0.3 & 1 \\ 0.35 & 1 \\ 0.4 & 1 \\ 0.45 & 1 \end{pmatrix}$$

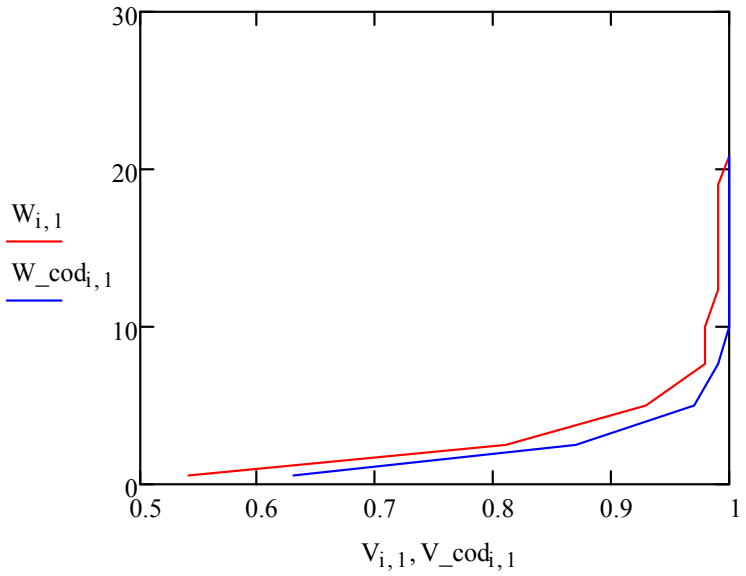
$w = 2.55$

$$W := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix} \quad W\_cod := \begin{pmatrix} 0.01 & 0.55 \\ 0.05 & 2.55 \\ 0.1 & 5.0 \\ 0.15 & 7.6 \\ 0.2 & 10 \\ 0.25 & 12.4 \\ 0.3 & 14.7 \\ 0.35 & 16.9 \\ 0.4 & 19 \\ 0.45 & 20.8 \end{pmatrix}$$

$i := 0..9$






30
20
10
0
0.5
0.6
0.7
0.8
0.9
1
W_cod i, 1
V_cod i, 1

Для нотаток

Навчальне видання

Кузнецов Олександр Олександрович
Полуяненко Микола Олександрович
Кузнецова Тетяна Юріївна

ПРИХОВУВАННЯ ДАНИХ У ПРОСТОРОВІЙ ОБЛАСТІ НЕРУХОМИХ ЗОБРАЖЕНЬ МЕТОДОМ БЛОКОВОГО ВБУДОВУВАННЯ, МЕТОДОМ КВАНТУВАННЯ ТА МЕТОДОМ «ХРЕСТА»

Методичні рекомендації
до лабораторної роботи з дисципліни «Стеганографія»
для студентів спеціальності 125 «Кібербезпека»

Коректор *Н. В. Мазепа*
Комп'ютерне верстання *Л. П. Зябченко*
Макет обкладинки *І. М. Дончик*

Формат 60×84/16. Ум. друк. арк. 2,86. Наклад 50 пр. Зам № 142/19.

Видавець і виготовлювач
Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2019

Видавництво ХНУ імені В. Н. Каразіна
Тел. 705-24-32