

## І Н Ф О Р М А Т И К А

УДК 004.8

### ВІДНОВЛЕННЯ ГРАФІЧНИХ ОБРАЗІВ ЗА ДОПОМОГОЮ КАРТ КОХОНЕНА

**В. Ленко, А. Караба**

*Львівський національний університет імені Івана Франка,  
вул. Університетська, 1, Львів, 79000, e-mail: [vs.lenko@gmail.com](mailto:vs.lenko@gmail.com)*

Розглянуто алгоритм відновлення графічних образів за допомогою карти Кохонена. Досліджено аспекти ефективної реалізації алгоритму, зокрема введено поняття “карти міри пошкодженості блоків”. Оцінено аспекти практичної реалізації алгоритму, що дало змогу виявити обмеження його застосовності. Проаналізовано залежність результату відновлення від вхідних параметрів алгоритму.

*Ключові слова:* відновлення графічного образу, штучні нейронні мережі, карта Кохонена.

#### 1. ВСТУП

Відновлення графічного образу – це процес відтворення втраченої інформації або певної її частини у графічному образі (зображенні, фотографії). Втраченою інформацією може бути відсутнє значення кольору та певна нечіткість даних. Мета відновлення – отримати більш якісний графічний образ, в якому немає втраченої інформації, або ж її кількість менша порівняно з вихідним зображенням.

Значний прогрес у сфері інформаційних технологій разом з великою кількістю наукових напрацювань у галузі штучного інтелекту допомогли автоматизувати процес відновлення графічного образу. Розвиток алгоритмів, які дають змогу якісно відновлювати графічну інформацію, є важливим кроком для підвищення продуктивності інтелектуальних систем, зокрема систем комп’ютерного зору.

Сьогодні задача відновлення зображення залишається серйозним викликом для реалізації систем штучного інтелекту. Дослідження шляхів підвищення швидкодії та поліпшення якості відновлення допоможе зменшити час прийняття рішення та покращити його ефективність.

Для вирішення проблеми відновлення графічного образу існує велика кількість різноманітних підходів та алгоритмів. У 2010 р. в [1] було подано застосування карт Кохонена для відновлення зображень. Отримані результати виявились набагато ліпшими, ніж при використанні методу інтерполяції бікубічними сплайнами. У 2011 р. в [2] було проаналізовано, як на якість відновлення впливають параметри карти, а саме розмір кожного нейрона, кількість нейронів на карті, кількість ітерацій навчання. Отримані результати дали змогу говорити про те, що карти Кохонена можна використовувати для відновлення втрачених мультимедійних даних, де якість відновлення досить висока порівняно з класичними алгоритмами.

#### 2. ФОРМУЛЮВАННЯ ЗАДАЧІ

Нехай задано довільний графічний образ, в якому певної частини інформації немає або пошкоджена. Щоб підвищити його інформативність, застосуємо один з

алгоритмів відновлення графічної інформації. Зауважимо, що алгоритм повинен швидко та якісно розв'язувати сформульовану задачу.

Зазвичай процес відновлення зображення характеризується значною кількістю операцій з великими об'ємами даних. Тому доцільно дослідити алгоритм стосовно використання оптимальних структур даних для забезпечення його швидкодії. Не менш важливим чинником є якість отриманих результатів, тому треба визначити вплив значень параметрів алгоритму на якість відновлення втраченої інформації. Останнім важливим елементом дослідження є виявлення обмежень застосовності алгоритму, які допоможуть означити його корисність для розв'язання реальних практичних задач.

Як підослідного обрали алгоритм відновлення графічного образу за допомогою карти Кохонена. В праці [1] описано метод відновлення інформації на підставі блоків пікселів, з використанням видозмінених формул для навчання карти Кохонена. Важливо визначити надійність методу та доцільність запропонованих модифікацій. Для цього потрібно проаналізувати структуру алгоритму та порівняти результати відновлення з тими, які отримали при навчанні карти класичними формулами.

### 3. САМООРГАНІЗАЦІЙНІ КАРТИ КОХОНЕНА

Самоорганізаційні карти Кохонена – це спеціальний клас штучних нейронних мереж, робота яких ґрунтується на конкурентному принципі навчання (competitive learning): виходи нейронів карти конкурують між собою за право перейти в стан збудження. Виходом мережі вважають нейрон-переможець (winner takes all).

Карта Кохонена складається з набору нейронів. Кількість нейронів задає розробник. Кожний нейрон зазвичай описується двома векторами:

- 1) вектор ваги нейрона,  $m$ . У загальному випадку розмірність вектора збігається з розмірністю векторів даних;
- 2) вектор координат,  $p$ . У загальному випадку вектор  $p$  описує точку в просторі, в якому розташований нейрон.

Зазвичай нейрони розташовують на площині у вершинах регулярної решітки з квадратними або шестикутними комірками. У процесі конкурентного навчання нейрони вибірково налаштовують на різні вхідні образи (стимули), а позиції нейронів-переможців впорядковуються стосовно інших.

#### 3.1. ІНІЦІАЛІЗАЦІЯ КАРТИ КОХОНЕНА

Ініціалізація карти, яка полягає у присвоєнні значень вектору ваг  $m$  в кожному нейроні, може бути виконана кількома способами:

- 1) задання всіх координат випадковими числами;
- 2) присвоєння вектору ваги значень випадкового спостереження з вхідних даних;
- 3) вибір векторів ваги з лінійного простору, натягнутого на головні компоненти набору вхідних даних.

Зауважимо, що кожний нейрон після ініціалізації стає нерухомим на карті, тобто вектор  $p$  не змінюється протягом усього навчання.

#### 3.2. АЛГОРИТМ НАВЧАННЯ КАРТИ КОХОНЕНА

Навчання карти Кохонена відбувається ітеративно. Нехай  $t$  – номер ітерації. Приймемо, що ініціалізація карти мала номер ітерації 0. Далі виконуються такі операції.

1. Вибираємо випадковий вектор  $x(t)$  з набору вхідних значень.

2. Знаходимо відстані до всіх векторів ваг усіх нейронів карти. Для цієї операції вибирають довільну міру, середньоквадратичне відхилення. Шукаємо нейрон, який найбільш близький до вхідного значення  $x(t)$ :  $d(x(t), m_c(t)) \leq d(x(t), m_i(t))$ , де  $m_c(t)$  – вектор ваги нейрона переможця (ВМУ, winner)  $M_c(t)$ ;  $m_i(t)$  – вектор ваги  $i$ -го вузла карти;  $d(x(t), m_i(t))$  – деяка міра відхилення. У випадку, якщо згадану умову задовольняють декілька нейронів, ВМУ вибирають випадково.

3. Визначення міри сусідства нейронів і зміна ваг нейронів у карті:

а) вибираємо міру сусідства – функцію  $h_{ci}(t)$ : ця функція визначає міру сусідства між нейронами  $M_i$  та  $M_c$ . Зазвичай як функції  $h_{ci}(t)$  використовують Гаусову функцію:

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\delta^2(t)}\right),$$

де  $0 < \alpha(t) < 1$  – крок навчання, який монотонно зменшується з кожною наступною ітерацією (тобто визначає коефіцієнт наближення значення векторів ваг ВМУ і його сусідів до спостереження);  $r_i, r_c$  – координати вузлів  $M_i(t)$  та  $M_c(t)$  на карті;  $\delta(t)$  – співмножник, який зменшує кількість сусідів з ітераціями, монотонно зменшується. Параметри  $\alpha, \delta$  та їхній характер задає аналітик. Простіший спосіб задання функції сусідства:  $h_{ci}(t) = \alpha(t)$ , якщо  $M_i(t)$  розташований в околі  $M_c(t)$  попередньо заданого аналітиком радіусу, і 0 в протилежному випадку. Функція  $h(t)$  дорівнює  $\alpha(t)$  для ВМУ (нейрона переможця) і зменшується з віддаленням від ВМУ;

б) обчислення помилки карти. Змінюємо вектор ваг згідно з класичною формулою:

$$m_i(t) = m_i(t-1) + h_{ci}(t)(x(t) - m_i(t-1)).$$

Отож, всі вузли, які є сусідами ВМУ, наближаються до спостереження, яке розглядається.

Вибір умови зупинки. Для визначення критерію зупинки здебільшого використовується помилка карти, наприклад, як середня арифметична відстань між спостереженнями та векторами ваг відповідних їм ВМУ:

$$\frac{1}{N} \sum_{i=1}^N \|x_i - m_c\|,$$

де  $N$  – кількість елементів набору вхідних даних.

#### 4. ПРИСТОСУВАННЯ КАРТ КОХОНЕНА ДЛЯ ВІДНОВЛЕННЯ ГРАФІЧНОГО ОБРАЗУ

Враховуючи вищеописаний алгоритм навчання карти Кохонена, можемо виділити такі елементи для означення алгоритму відновлення графічного образу:

1)  $x$  – вектор даних, за якими спостерігають. У застосуванні до відновлення графічного образу це може бути деякий набір пікселів, згрупованих за певним правилом;

2)  $m$  – вектор ваги нейрона. За означенням, має бути однієї розмірності з  $x$ ;

3)  $h_{ci}(t)$  – міра сусідства нейронів. Деяка функція, що повертає відстань між нейронами і залежить від номера ітерації  $t$  ;

4)  $d(x(t), m_i(t))$  – міра відхилення, яка виявляє, наскільки вектор  $x(t)$  не схожий на вектор  $m_i(t)$  .

Тепер опишемо міри та методи побудови векторів даних для графічних образів. Загалом зображення, представлене в цифровому вигляді, є набором пікселів. Зазвичай зображення є прямокутним і характеризується шириною та висотою в пікселях, а також кількістю значень, які може приймати кожен піксель. Палітру, як набір значень кольорів, потрібно розширити, додавши колір пошкодженого пікселя.

Для подальшого дослідження введемо такі поняття та позначення:

- піксель – номер кольору з палітри зображення, який відповідає певній точці на зображенні;
- пошкоджений піксель – брак номеру кольору з палітри зображення, який відповідає точці на зображенні, з огляду на пошкодження цієї точки;
- $G$  – палітра зображення. В загальному випадку  $G \subset Z$  ;
- $\#$  – колір, який відповідає пошкодженому пікселю;
- $B = G \cup \{\#\}$  – розширена палітра, яка зберігає в собі колір пошкодженого пікселя;
- $x[i]$  –  $i$ -та компонента вектора  $x$  ;
- $W_p \in Z$  – кількість стовпців у вхідному зображенні (ширина зображення в пікселях);
- $H_p \in Z$  – кількість рядків у вхідному зображенні (висота зображення в пікселях);
- $P[i, j] \in B$  – піксель, розташований в  $i$ -му рядку та  $j$ -му стовпчику на зображенні;
- $R[i, j] \in G$  – відновлене значення пікселя, розташованого в  $i$ -му рядку та  $j$ -му стовпчику на зображенні;
- $Q_x$  – ширина карти в нейронах;
- $Q_y$  – висота карти в нейронах;
- $Q$  – кількість нейронів у карті;
- $m\{x\} \in (0, \dots, Q_x - 1)$  – горизонтальна координата положення нейрона  $m$  ;
- $m\{y\} \in (0, \dots, Q_y - 1)$  – вертикальна координата положення нейрона  $m$  ;

Вибираючи вектор спостережень, обов'язково треба враховувати умови:

$$\lfloor S/2 \rfloor \leq x \leq W_p - \lfloor S/2 \rfloor, \lfloor S/2 \rfloor \leq y \leq H_p - \lfloor S/2 \rfloor,$$

$S$  повинно бути непарним. Перші дві умови вводяться на підставі того, що розмір блока, який розглядається, більше одного пікселя і кількість таких блоків буде меншою, ніж кількість пікселів у зображенні. Третя умова накладається для точнішого опису околу пікселя, а саме для його симетричності.

Міра сусідства нейронів  $h_{ci}(t)$  – це функція від індексів нейронів і номера ітерації. Її можемо зобразити в такому вигляді:

$$h_{ci}(t) = h(c, i, t),$$

де  $h$  – деяка задана аналітиком функція;  $c$  – індекс першого нейрона;  $i$  – індекс другого нейрона;  $t$  – номер ітерації.

Під час дослідження було вирішено використати видозмінену функцію Гауса

$$h(c, i, t) = \alpha(t) \cdot \exp\left(-\frac{d(c, i)^2}{2\delta(t)}\right). \quad (1)$$

Тут  $\alpha(t)$  набуває вигляду

$$\alpha(t) = \begin{cases} 1, & \text{якщо } t < 10, \\ \frac{1}{(t-9)^{0.2}}, & \text{якщо } t \geq 10, \end{cases} \quad (2)$$

$d(c, i)$  виглядає так:

$$d(c, i) = \min_{\substack{d_x \in \{-Q_x, 0, Q_x\} \\ d_y \in \{-Q_y, 0, Q_y\}}} \sqrt{(m_c\{x\} - m_i\{x\} + d_x)^2 + (m_c\{y\} - m_i\{y\} + d_y)^2}, \quad (3)$$

$\delta(t)$  набуває вигляду

$$\delta(t) = 5 \frac{\sqrt{Q}}{\sqrt{t}}. \quad (4)$$

Функція (2) на перших ітераціях має значення 1. Це зроблено для того, щоб максимально рівномірно ініціалізувати карту на початку навчання. При попаданні ВМУ в кут координатної сітки, кількість нейронів, які піддаються зміні при навчанні, зменшується в 4 рази порівняно з ситуацією, коли ВМУ потрапляє в центр карти. Саме тому функція (3) нівелює таку проблему і дає змогу максимізувати вплив навчання. Функція (4) зведена до такого вигляду у результаті експериментів при навчанні карти Кохонена.

У процесі навчання карти початково всі ваги нейронів виставлені в нульові значення (це відповідає чорному кольору в палітрі  $B$ ). Перші 10 ітерацій є ітераціями ініціалізації. Із зображення випадково вибирають 10 блоків, які не містять пошкоджених пікселів. У цьому разі положення ВМУ чітко задані, а саме – рівномірно розподілені по карті. Це робиться для того, щоб початково карта мала максимальну кількість інформації з зображення. На кожній наступній ітерації випадково вибирають пару координат  $(x, y)$  так, щоб виконувались умови:

$$\lfloor S/2 \rfloor \leq x \leq W_p - \lfloor S/2 \rfloor,$$

$$\lfloor S/2 \rfloor \leq y \leq H_p - \lfloor S/2 \rfloor,$$

$$P[i, j] \neq \#, \text{ при } i = x - S/2, \dots, S/2 + x, \quad j = y - S/2, \dots, S/2 + y.$$

Тобто, з зображення випадково вибирають блоки розміром  $S \times S$  пікселів, жоден з яких не є пошкодженим.

Відновлення зображення починається після процедури навчання. Воно, як і навчання, також відбувається ітераційно. При відновленні з набору векторів вхідних даних вибирають випадковий вектор, в якому пошкодженим є лише один піксель. Для цього вектора шукають ВМУ, але метрика в цьому випадку змінюється

$$d(x(t), m_i(t)) = \sum_{\substack{l \\ x(t)[l] \neq \#}} (m_i(t)[l] - x(t)[l])^2.$$

Отже, з всіх векторів ваг вибирають найбільш схожого, без врахування компоненти, що відповідає пошкодженому пікселю. Після того, як ВМУ знайшли, пошкоджений

піксель на зображенні замінюється на відповідний із вектора ваг нейрона. Цю процедуру повторюють доти, доки на вхідному зображенні не буде пошкоджених пікселів.

#### 5. ПРОГРАМНИЙ ЕКСПЕРИМЕНТ

Щоб дослідити властивості та ефективності алгоритму, реалізували об'єктно-орієнтоване програмне забезпечення мовою Java 1.7. Його функціональність полягає у можливості задання значень вхідних параметрів, відновленні графічного образу та візуалізації результату. Параметри алгоритму виглядають так

- графічний образ – матриця пікселів, в якій кожен піксель зображений RGB значенням;
- розмір блока,  $S$  – ціле непарне число, яке повинно задовольняти умову  $1 < S < W_p$  ;
- розмір карти,  $Q_x$  та  $Q_y$  – цілі числа, що є більшими за одиницю;
- колірний простір – простір, в якому подані значення пікселів графічного образу (RGB, CIELAB);
- метрика відстані між кольорами – для RGB простору передбачені Евклідова та Мангеттенська; для CIELAB – метрики CIE76, CIE94, CIEDE2000;
- параметри навчання карти – реалізована можливість вибору між типовими параметрами навчання карти Кохонена та видозміненими, які описані в [1].

Як тільки значення усіх параметрів задані, можна розпочинати відновлення пошкоджених пікселів. Процес роботи алгоритму виглядає так:

1. Привести графічний образ до заданого колірного простору.
2. Створити блоки для навчання карти.
3. Виконати навчання карти.
4. Відновити пошкоджені пікселі.
5. Подати графічний образ до простору RGB.
6. Візуалізувати результат відновлення.

#### *Подання графічного образу до заданого колірного простору.*

Цей крок доволі тривіальний, оскільки всі перетворення виконуються за традиційними формулами переходу між колірними просторами. Для зберігання інформації доцільно попередньо створити матрицю розміром  $W_p \times H_p \times 3$ , відповідно кожний піксель буде зображений масивом з 3 елементів – значеннями насиченості каналів кольору. Якщо піксель пошкоджений, то для усіх елементів масиву, які його зображають, приймемо значення Double.NaN.

#### *Створення блоків для навчання карти.*

Для навчання карти Кохонена треба створити вибірку блоків, причому жоден з них не повинен містити пошкоджених пікселів. Задля оптимізації створення блоків без пошкоджених пікселів та з одним пошкодженим пікселем (крок 4) доцільно створити карту міри пошкоженості блоків – матрицю розміром  $W_p \times H_p$ , де значення комірки  $(i, j)$  відповідає кількості пошкоджених нейронів у блоці з центром в точці  $(i, j)$ . Зауважимо, що ці карти безпосередньо залежать від розміру блоку. Тому для опису карти міри пошкоженості потрібно зазначати графічний образ і розмір блоку, якому вона відповідає.

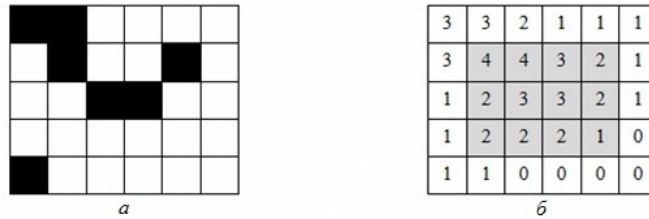


Рис. 1. (а) матричне подання графічного образу розміром  $6 \times 5$ ; чорні комірки відповідають пошкодженим пікселям; (б) карта міри пошкодженості для блоків розміром  $3 \times 3$

Алгоритм створення карти міри пошкодженості графічного образу з розміром  $W_p \times H_p$  для блоків розміром  $S \times S$  виглядає так.

- 1) Створити матрицю  $W_p \times H_p$ , кожний елемент якої початково дорівнює нулю.
- 2) Для кожного пошкодженого пікселя:
  - а) визначити його координати  $(i, j)$  у графічному образі;
  - б) збільшити значення на одиницю для тих комірок карти, координати яких  $(x, y)$  задовольняють умови

$$\min(i - \lfloor S/2 \rfloor, 0) \leq x \leq \max(i + \lfloor S/2 \rfloor, W_p - 1),$$

$$\min(j - \lfloor S/2 \rfloor, 0) \leq y \leq \max(j + \lfloor S/2 \rfloor, H_p - 1).$$

Підготувавши карту, можна переходити до створення блоків, які будуть використані для навчання нейромережі. Алгоритм створення блоків для навчання карти Кохонена запишемо так.

1. Для кожної комірки з карти міри пошкодженості з координатами  $(i, j)$  та значенням  $n$  перевірити виконання таких умов:

$$\lfloor S/2 \rfloor \leq i < W_p - \lfloor S/2 \rfloor,$$

$$\lfloor S/2 \rfloor \leq j < H_p - \lfloor S/2 \rfloor,$$

$$n = 0.$$

2. Якщо всі умови виконуються, то створити блок розміром  $S \times S$ , центром якого є піксель з координатами  $(i, j)$ .

З умов видно, що стовпці та рядки, які розташовані біля країв карти на відстані меншій, ніж  $\lfloor S/2 \rfloor$ , не використовують при створенні блоків, оскільки їхня позиція фізично не може відповідати позиції центру блока. Ця ідея проілюстрована на рис. 1, де сірим кольором позначено елементи, які використовують для створення блоків розміром  $3 \times 3$ . Отож, розмір карти міри пошкодженості може бути зменшеним до  $W_p - S + 1 \times H_p - S + 1$ , проте така оптимізація погіршить зручність роботи з картою, а також принесе додаткові обчислення для виявлення відповідності між комірками карти та пікселями графічного образу.

#### Навчання карти Кохонена.

Спершу треба створити двовимірну карту Кохонена розміром  $Q_x \times Q_y$ , в якій кожен нейрон характеризується позицією на карті  $(x, y)$  та ваговим вектором. Розмірність вагового вектора в усіх нейронів є однаковою і відповідає розмірності

вектора блока навчання. Загалом карту зручно подати у вигляді тривимірного масиву розміром  $Q_x \times Q_y \times (S * S * 3)$ .

Процес навчання розпочинається з ініціалізації значень вагових векторів нейронів карти. Початково усі елементи в кожному з вагових векторів повинні дорівнювати нулю. Задля того, щоб карта мала максимальну кількість інформації з графічного образу, з множини блоків для навчання вибирають 10 найінформативніших блоків і рівномірно розподіляють їх між нейронами карти. Інформативність блока з центром у пікселі  $(i, j)$  визначають за формулою

$$\sum_{\forall p[x,y] \in \text{блоку}} (P[i, j] - P[x, y])^2.$$

Процес вибірки  $k$  найінформативніших блоків доцільно реалізувати так.

1. Використати алгоритм пошуку  $k$ -ї порядкової статистики – BFPRT, для знаходження позиції  $k$ -го найінформативнішого блока. Часова складність:  $O(n)$ .
2. Застосувати процедуру Quicksort Partition з опорним елементом, позицію якого знайшли на попередньому кроці. Часова складність:  $O(n)$ .

Знайшовши 10 найінформативніших блоків можна розпочинати процес навчання карти Кохонена. Перші 10 ітерацій навчання полягають у рівномірному розподілі знайдених блоків на карті: кожному блоку ставлять у відповідність нейрон-переможець, координати якого визначають згідно з рівномірним розподілом. Отже, усі початкові ітерації складаються лише з двох етапів: кооперація та адаптація; етапу конкуренції немає.

Варто також зазначити, що в процесі дослідження постало питання доцільності вибору саме 10 блоків; ймовірно для карти, в якій кількість нейронів  $Q < 10$ , цілком достатньо знайти  $Q$  блоків. Зауважимо, що врахування цієї поправки потребує змін у формулі кроку навчання карти  $\alpha(t)$ . Також з вигляду формули  $\alpha(t)$  можна визначити, що початковим ітераціям відповідають порядкові числа  $\overline{0,9}$ , відповідно  $\delta(0) = \infty$ . Очевидно таке значення є небажаним, тому його можна уникнути, розпочинаючи відлік ітерацій з 1, що, зрештою, вплине на вигляд функції  $\alpha(t)$ . Підсумовуючи: нехай  $k = \min(10, Q)$  – кількість блоків ініціалізації, тоді рекомендованим виглядом функції  $\alpha(t)$  є такий:

$$\alpha(t) = \begin{cases} 1, & | t \leq k, \\ \frac{1}{(t-k)^{0.2}}, & | t > k. \end{cases}$$

Після ініціалізації, навчання карти продовжується за класичною схемою, де кожна ітерація складається з трьох етапів: конкуренція, кооперація та адаптація. Вибір блоків для навчання відбувається випадково. Критерієм зупинки доцільно вибрати факт, який у процесі адаптації жодна компонента у жодному з вагових векторів не змінила свого значення більше, ніж на 1, тобто  $\forall i, |\Delta P[i]| < 1$ .

#### Відновлення пошкоджених пікселів.

По закінченні навчання карти Кохонена можна приступати безпосередньо до відновлення пошкоджених пікселів. Процес відновлення виконується ітеративно,



причому на кожній ітерації відновлюється рівно один піксель. Алгоритм ітерації виглядає так.

1. З графічного образу вибирають блок з одним пошкодженим пікселем.
2. Для блоку шукають нейрон-переможець (процес конкуренції).
3. Пошкоджений елемент у блоці замінюється відповідним елементом вагового вектора нейрона-переможця.
4. Виконується оновлення графічного образу, шляхом заміни відновленим значенням відповідного пошкодженого пікселя.

З перелічених кроків детального опису заслуговує перший та останній – решта реалізуються згідно з формулами алгоритму.

Суть першого кроку полягає у виборі блока, в якому лише один піксель є пошкодженим. Пошук координат центра такого блока можна виконати двома способами:

- використовуючи тільки карту міри пошкодженості;
- використовуючи список пошкоджених пікселів і карту міри пошкодженості.

В обох випадках шукають комірку з координатами  $(i, j)$  та значенням  $n$ , яка задовольняє такі умови:

$$\begin{aligned} \lfloor S/2 \rfloor \leq i < W_p - \lfloor S/2 \rfloor, \\ \lfloor S/2 \rfloor \leq j < H_p - \lfloor S/2 \rfloor, \\ n = 1. \end{aligned}$$

Відмінність цих способів полягає у стратегії пошуку, тобто послідовності перебору комірок з карти міри пошкодженості блоків.

Стратегія пошуку для першого способу полягає у звичайному проходженні по карті міри пошкодженості блоків для пошуку комірки, яка задовольняє вищеописані умови. Як тільки комірка знайдена – пошук припиняється. Недоліком цього способу є той факт, що пошкоджений піксель зазвичай розташований на окраїні блоку (переважно в правому нижньому куті), що значною мірою погіршує якість відновлення. Перевага ж полягає у швидкості пошуку та відсутності потреби зберігання списку пошкоджених пікселів.

Якість відновлення пікселя значною мірою залежить від його позиції в блоці відновлення. Зазвичай можна створити декілька блоків відновлення для одного пошкодженого пікселя, в яких він займатиме різні позиції. Найоптимальнішим є той блок, у якому пошкоджений піксель є центром блока, проте зрозуміло, що не завжди такий блок задовольняє критерії алгоритму. Тому, як загальне правило, серед прийнятних блоків доцільно вибирати той, який мінімізує відстань пошкодженого пікселя до центра блока. Саме ця ідея стала основою стратегії пошуку для другого способу. Її реалізація виглядає так:

- 1) вибрати піксель зі списку пошкоджених;
- 2) комірку, позиція якої відповідає координатам пікселя, прийняти за центр блоку;
- 3) усі комірки, які входять до блоку, відсортувати за мангеттенською відстанню до центра блока у порядку зростання;
- 4) серед відсортованих комірок виконати пошук такої, що задовольняє критерій центра пошкодженого блоку;
- 5) якщо пошук не дав результатів, то повернутися до кроку 1.

Перевага другого способу – вища якість відновлення. Недолік полягає у зменшенні швидкості пошуку та необхідності зберігання списку пошкоджених пікселів.

Коли комірку знайдено, то створюється відповідний блок, зберігаються позиції пошкодженого пікселя. Далі виконується відновлення пошкодженого значення та оновлення відповідного пікселя в графічному образі. Також після кожної ітерації потрібно оновити карту міри пошкоженості блоків, а саме: зменшити на 1 значення в усіх комірках, які входять до блока, центром якого є відновлений піксель.

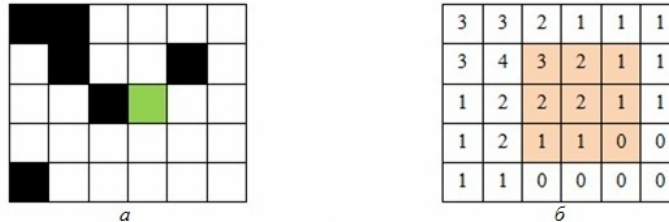


Рис. 2. (а) відновлений піксель; (б) оновлена карта міри пошкоженості для блоків розміром  $3 \times 3$

Варто зазначити, що в процесі дослідження алгоритму було виявлено обмеженість його застосовності, яка полягає в тому, що графічний образ може містити такі пошкодження, за яких на деякій ітерації відновлення не знайдеться блоку з одним пошкодженим пікселем (рис. 3). Один з варіантів обходу цього обмеження – наявність більше, ніж одного пікселя в пошкодженому блоці. Така модифікація породжує нову задачу, що полягає у пошуку послідовності пошкоджених блоків, яка максимізуватиме якість відновлення графічного образу. Розв'язання цієї задачі потребує додаткових досліджень.

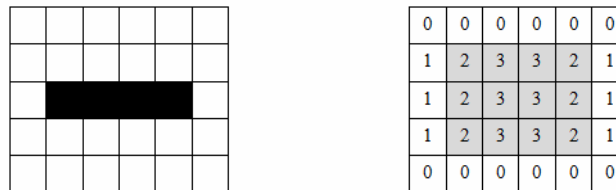


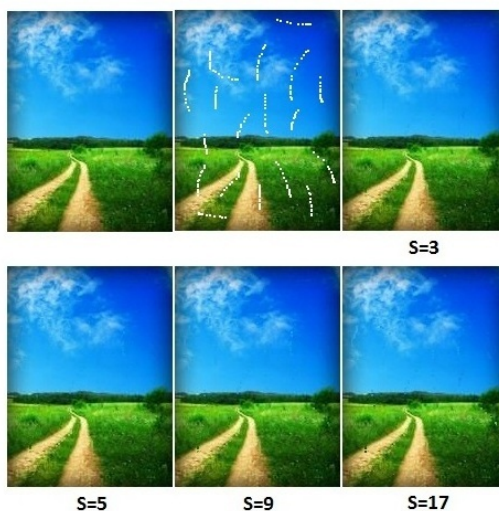
Рис. 3. Пошкодження, які неможливо відновити алгоритмом

#### Подання графічного образу до RGB простору.

Після відновлення пошкоджень, щоб їх візуалізувати, доцільно подати колірні дані до простору RGB. Це перетворення виконують згідно з формулами переходу між просторами кольорів.

#### 6. АНАЛІЗ РЕЗУЛЬТАТІВ ПРОГРАМНОГО ЕКСПЕРИМЕНТУ

Оцінимо вплив параметрів карти Кохонена на якість відновлення графічного образу. Першим розглянемо розмір блоку,  $S$ . Цей параметр має значний вплив на навчання карти. На рис. 4 зображено результати відновлення зображення за різних значень  $S$ . З рис. 4 видно, що якість відновлення при  $S = 17$  трішки гірша, ніж при інших значеннях. Це пояснюється тим, що блок з великим розміром містить багато пікселів, які значно різняться між собою, відповідно їхнє узагальнення – це компроміс, який зазвичай відрізняється від пікселів, що формують блок.

Рис. 4. Відновлення зображення за різних значень  $S$ 

Наступним важливим параметром є розмір карти Кохонена. Чим більший розмір карти, тим результат відновлення повинен бути якіснішим, адже в цьому випадку існує більша кількість нейронів, відповідно більша кількість класів спостережень. З рис. 5 видно, що при малих розмірах карти пошкоджені кольори замінюються на візуально невідповідні. Коли розмір карти є 10 на 10 нейронів, то навіть за значних початкових пошкоджень відновлене зображення майже не відрізняється від початкового.

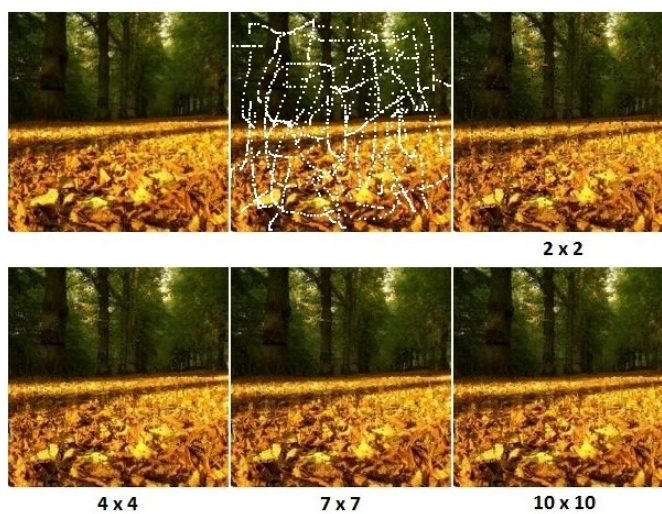


Рис. 5. Відновлення зображення при різних розмірах карти Кохонена

Також доцільно порівняти результати роботи алгоритму з використанням видозмінених і класичних формул для навчання карти. Як видно з рис. 6, використання модифікованих формул дає ліпший результат – пошкоджених пікселів

на зображенні майже не помітно. Це пояснюється тим, що формули були змінені для підвищення якості відновлення графічної інформації, тоді як класичні є своєрідним узагальненням, яке потребує вдосконалення у кожному окремому випадку застосування. Зауважимо, що значну роль також відіграє початкова ініціалізація нейронів і критерії зупинки навчання карти.



Рис. 6. Відновлення зображення з використанням типових і видозмінених параметрів карти Кохонена

Вибір метрики для визначення різниці між пікселями також має вплив на результат відновлення графічного образу. На рис. 7 зображено результат використання різних метрик. Зокрема, для простору RGB це Евклідова та Мангеттенська метрики, а для простору CIELAB – метрики CIE74, CIE94, CIEDE2000.



Рис. 7. Відновлення зображення з використанням різних метрик кольору

## 7. ВИСНОВКИ

Ми дослідили практичні аспекти реалізації відновлення графічного образу за допомогою карти Кохонена. Кожен крок алгоритму ретельно аналізували, внаслідок чого досягнули таких наукових результатів:

- 1) ввели поняття структури даних карти міри пошкодженості блоків, яка дає змогу ефективно оперувати блоками пікселів;
- 2) обґрунтована модифікація формули кроку навчання карти Кохонена  $\alpha(t)$ ;
- 3) запропоновано та досліджено два способи пошуку пошкодженого блока;
- 4) виявлено обмеженість застосування алгоритму до відновлення графічного образу, яка спричинена специфікою певних пошкоджень.

Розроблене ПЗ допомогло якісно оцінити залежність результатів відновлення від параметрів алгоритму. Виявлено, що класичні формули для навчання карти Кохонена дають гірший результат, ніж формули, які запропоновані в алгоритмі.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Восстановление изображений при помощи нейросетей. – 2011. – [Електронний ресурс]: <http://habrahabr.ru/post/120473/>.
2. Карты Кохонена как способ восстановления мультимедийной информации. – 2010. – [Електронний ресурс]: <http://jre.cplire.ru/koi/oct10/3/text.html>.
3. *Нікольський Ю. В.* Системи штучного інтелекту / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина. – Львів.: Магнолія-2006, 2010. – 279 с.
4. *Хайкен С.* Нейронные сети: полный курс / С. Хайкен, 2-е изд., испр.: пер с англ. – М.: ООО И. Д. Вильямс, 2006. – 1104 с.
5. Kohonen's Self Organizing Feature Maps. – 2013. – [Електронний ресурс]: <http://www.ai-junkie.com/ann/som/som1.html>.

*Стаття: надійшла до редколегії 05.02.2014*

*доопрацьована 19.03.2014*

*прийнята до друку 09.04.2014*

## ВОССТАНОВЛЕНИЕ ГРАФИЧЕСКИХ ОБРАЗОВ С ПОМОЩЬЮ КАРТ КОХОНЕНА

**В. Ленко, А. Караба**

*Львовский национальный университет имени Ивана Франко,  
ул. Университетская, 1, Львов, 79000, e-mail: [vs.lenko@gmail.com](mailto:vs.lenko@gmail.com)*

Рассмотрен алгоритм восстановления графических образов с помощью карты Кохонена. Исследованы аспекты эффективной реализации алгоритма, в частности введено понятие карты степени поврежденности блоков. Оценены аспекты практической реализации алгоритма, что позволило выявить ограничения его применимости. Проанализировано зависимость результата восстановления от входных параметров алгоритма.

*Ключевые слова:* восстановление графического образа, искусственные нейронные сети, карта Кохонена.

**AN IMAGE DATA RECOVERY USING KOHONEN'S MAPS****V. Lenko, A. Karaba***Ivan Franko National University of Lviv,  
Universytetska Str., 1, Lviv, 79000, e-mail: [vs.lenko@gmail.com](mailto:vs.lenko@gmail.com)*

An image data recovery algorithm based on Kohonen map is considered. The aspects of an efficient algorithm implementation are investigated; in particular the notion of the map of the measure of block damage is introduced. During the assessment of the aspects of practical implementation the restrictions of algorithm applicability were discovered. Correlation between the result and input parameters values is analyzed.

*Key words:* image recovery, artificial neural networks, Kohonen map.