

УДК 005:37

Лізунов Петро Петрович

Доктор технічних наук, професор, завідувач кафедри основ інформатики, orcid.org/0000-0003-2924-3025
Київський національний університет будівництва і архітектури, Київ

Білощицький Андрій Олександрович

Доктор технічних наук, професор, заступник декана факультету інформаційних технологій, orcid.org/0000-0001-9548-1959
Київський національний університет ім. Т. Шевченка, Київ

Чала Лариса Ернестівна

Кандидат технічних наук, доцент, доцент кафедри штучного інтелекту, orcid.org/0000-0002-9890-4790
Харківський національний університет радіоелектроніки, Харків

Білощицька Світлана Василівна

Кандидат технічних наук, доцент кафедри інформаційних технологій проектування та прикладної математики, orcid.org/0000-0002-0856-5474
Київський національний університет будівництва і архітектури, Київ

Кучанський Олександр Юрійович

Кандидат технічних наук, доцент кафедри інформаційних технологій, orcid.org/0000-0003-1277-8031
Київський національний університет будівництва і архітектури, Київ

Удовенко Сергій Григорович

Доктор технічних наук, професор, завідувач кафедри інформатики та комп'ютерної техніки, orcid.org/0000-0001-5945-8647
Харківський національний економічний університет ім. С. Кузнеця, Харків

АВТОМАТИЧНИЙ АНАЛІЗ ПОДІБНОСТЕЙ СХЕМ ТА ДІАГРАМ В ЕЛЕКТРОННИХ ТЕКСТОВИХ ДОКУМЕНТАХ

***Анотація.** Поширеним типом графічних об'єктів в наукових текстах, що тематично пов'язані з проблемою розробки та використання технологій обробки даних в складних системах різного функціонального призначення, є схеми перетворення інформації (насамперед схематичні відображення алгоритмів системи) та спеціалізовані діаграми. Для аналізу подібностей схем та діаграм, які містяться в текстових документах, доцільно використовувати не лише оцінювання характеристик графічних зображень, застосовуючи стандартні методи їх порівняння, а й брати до уваги особливості побудови схемних зображень. Тому для знаходження повних або неповних дублікатів таких графічних об'єктів пропонується підхід, що передбачає використання результатів аналізу їх топологічної структури.*

***Ключові слова:** схеми та діаграми; сегментація; кодування; дублікат; графічні формати*

Вступ

Сьогодні існують ефективні засоби перевірки наукових текстових документів на можливу наявність подібностей з публікаціями, що вже знаходяться в електронних бібліотеках або в ресурсах глобальної мережі Інтернет [1 – 7]. В той же час системи пошуку повних чи неповних дублікатів зазвичай не мають можливості автоматично аналізувати подібності графічних об'єктів, які є частиною порівнюваних документів (статей, доповідей, дипломних проєктів, магістерських або дисертаційних робіт, звітів і т.ін.). До таких об'єктів належать усі ілюстративні елементи: рисунки, схеми, графіки, діаграми, фото,

графічні зображення тощо [8]. Поширеним різновидом графічних елементів сучасних наукових текстів, що тематично пов'язані з проблемою розробки та використання інформаційних технологій обробки даних в складних системах різного функціонального призначення, є схеми перетворення інформації (насамперед, схематичні відображення алгоритмів системи) та спеціалізовані діаграми. Діаграми та схеми є важливими інструментами в науці, бізнесі та індустрії. Вони широко застосовуються у процесі розробки бізнес-звітів, наукових статей та доповідей, технічної документації та інших типів документів. Крім того, побудова діаграм є невід'ємною складовою будь-яких UML-засобів. UML – уніфікована розповсюджена

мова моделювання. Для оптимізації дій проєктувальника були створені CASE-засоби спеціального виду, що дозволяють професійно будувати діаграми UML, які зараз є необхідним елементом системного та структурно-функціонального аналізу [9]. Крім того, графічне представлення діаграм UML, що інколи наводяться в електронних текстових документах, може здійснюватися за допомогою деяких графічних редакторів. Зазначимо, що і схеми алгоритмів, і діаграми UML є графічними зображеннями, що містять лінії, які відображають зв'язки між відповідними вузлами схеми або діаграми. При цьому топологія конкретних зображень разом з позначеннями, що зазвичай наявні на рисунках текстових документів, є певною мірою унікальною. Це означає, що порівняльний аналіз топології схем та діаграм в документах може доповнювати результати виявлення в них повних або неповних подібностей [10].

При цьому виникають проблеми, що не завжди можна вирішити за допомогою стандартних процедур. Ці проблеми виникають, насамперед, з використанням під час розробки графічних документів різноманітних графічних редакторів та форматів графічних даних. Слід зазначити, що дотепер відсутні універсальні вимоги до імпортування графічних об'єктів в тексти статей або доповідей. Зазвичай редакторські вимоги полягають у зазначенні мінімальної роздільної здатності (наприклад, не менше 300 dpi), переліку рекомендованих форматів представлення рисунків (наприклад, *bmp*, **.tif*, **.jpg*) та рекомендацій до бажаного використання деяких графічних редакторів (наприклад, *MS Visio*, *MS Paint*, *CorelDraw*, *AdobeIllustrator*). Інколи до вимог додаються умови імпортування графічного матеріалу в текст статті у вигляді об'єктів у градаціях сірого або чорно-білого, без використання обтінання та прив'язки об'єктів. Розміри рисунків та діаграм обмежуються при цьому заданими габаритами, а розміри шрифтів текстових об'єктів (підписи осей, шкала осей, легенди, підписи об'єктів тощо) у рисунках мають бути враховані так, щоб при масштабуванні цього рисунка остаточні розміри цих текстових об'єктів були сумірні з розміром основного тексту.

В даній роботі запропоновано підхід, який дозволяє аналізувати подібність схем та діаграм в наукових текстах, що порівнюються, з використанням аналізу структури вузлів та сегментів схемних графічних об'єктів.

Мета статті

Метою дослідження є аналіз існуючих засобів побудови схем та діаграм за допомогою сучасних редакторів та методів порівняння цих об'єктів в

наукових текстах, що дозволяє розробити підхід до виявлення повних або часткових дублікатів схем та діаграм в публікаціях.

Виклад основного матеріалу

1. Засоби побудови схем та діаграм

У тексті схеми та UML-діаграми можуть бути наведені як рисунки або як графічні об'єкти, створені за допомогою одного зі спеціальних графічних редакторів та збережені в одному із форматів графічних даних

1.1. Формати графічних даних

У комп'ютерній графіці застосовують чимало форматів файлів для збереження растрових або векторних зображень [11]. Розглянемо деякі з таких форматів, які використовуються для побудови схем та діаграм в текстових електронних документах.

TIFF (Tagged Image File Format). Формат призначений для збереження растрових зображень високої якості (розширення імені файлу – *.tif*). Відноситься до числа широко розповсюджених, відрізняється перенесенням між платформами (IBM PC і Apple Macintosh). Для зменшення розміру файлу застосовується алгоритм стиску LZW.

PSD (PhotoShop Document). Власний формат програми Adobe PhotoShop (розширення імені файлу – *.psd*), що дає можливість збереження растрової графічної інформації.

PCX. Формат збереження растрових даних програми PC PaintBrush фірми Z-Soft (розширення імені файлу – *.pcx*). Відсутність можливості зберігати кольоровідокремлені зображення і інші обмеження призвели до втрати популярності формату.

PhotoCD. Формат розроблений фірмою Kodak для збереження цифрових растрових зображень високої якості (розширення імені файлу – *.pcd*).

Windows Bitmap. Формат збереження растрових зображень в операційній системі Windows (розширення імені файлу – *.bmp*). Відповідно підтримується всіма додатками, що працюють у цьому середовищі.

JPEG (Joint Photographic Group). Формат призначений для створення та збереження растрових зображень (розширення імені файлу – *.jpg*). Дозволяє регулювати співвідношення між ступенем стиску файлу і якістю зображення. Формат найчастіше рекомендують використовувати для електронних публікацій.

GIF (Graphics Interchange Format). Стандартизований у 1987 році як засіб збереження стиснутих зображень з фіксованою (256) кількістю кольорів (розширення імені файлу – *.gif*). Набув популярності в Інтернеті завдяки високому ступеню стиску.

WMF (Windows MetaFile). Формат збереження векторних зображень операційної системи Windows (розширення імені файлу – .wmf). За визначенням підтримується всіма додатками цієї системи. Однак відсутність засобів для роботи зі стандартизованими колірними палітрами, прийнятими в поліграфії, і інші недоліки обмежують його застосування.

EPS (Encapsulated PostScript). Формат опису як векторних, так і растрових зображень мовою PostScript фірми Adobe, у фактичному стандарті в області дпечатних процесів і поліграфії (розширення імені файлу – .eps). Оскільки мова PostScript є універсальною, у файлі можуть одночасно зберігатися векторна і растрова графіка.

PDF (Portable Document Format). Формат опису документів, розроблений фірмою Adobe (розширення імені файлу – .pdf). Хоча цей формат в основному призначений для збереження документа цілком, його можливості дозволяють забезпечити ефективно представлення зображень в текстах.

1.2. Графічні редактори

Графічні редактори – це прикладні програми, призначені для створення й обробки графічних зображень (в тому числі, схем та діаграм) на комп'ютері в діалоговому режимі. Програми обробки графіки можна розділити на дві групи: растрові та векторні.

Розглянемо деякі з таких програм, які використовуються для побудови схем та діаграм в текстових електронних документах.

ABViewer. Редактор графічних файлів, що дозволяє працювати як з растровими, так і з векторними форматами файлів (зокрема з TIFF, JPEG, GIF).

b4Look. Редактор графічних файлів, що підтримує основні графічні формати: JPG, TIF, PNG, GIF, BMP. Програма використовує GDI + для максимально швидкої роботи з графічними файлами.

ACDSee. Редактор графічних файлів, що надає основні функції для обробки растрових зображень. До основних переваг програми слід віднести високу швидкість обробки графічних даних, багатопоточність, підтримку більшості відомих графічних форматів (зокрема, TIFF, JPEG, GIF, BMP), можливість конвертації зображень.

Adobe Photoshop. Цей пакет програм має потужні можливості для обробки зображень з використанням різноманітних фільтрів та ефектів. Пакет володіє засобами відновлення пошкоджених зображень, ретушування фотографій тощо. Підтримує як растрові (BMP, JPEG, GIF), так і векторні (AI, CDR) графічні формати.

CorelDraw. Професійний векторний графічний редактор, призначений для обробки і створення

векторної та растрової графіки. Програма Microsoft Draw, що входить в комплект MS Office, широко використовується в наукових статтях для створення схем та діаграм.

Gliffy. Графічний редактор, призначений для створення різноманітних схем, діаграм (зокрема BPMN, UML, UI Design, Venn diagrams, SWOT). Високий рівень функціональності досягається завдяки використанню технології Flash.

2. Метод аналізу та порівняння схем у наукових текстах

2.1. Аналіз просторової структури схем

Підхід, що пропонується, передбачає на першому етапі проведення аналізу схем алгоритмів та діаграм за їх просторовою топологією, тобто за особливістю структури зв'язків між вузлами схеми або діаграми [12].

Система автоматизованого аналізу просторової структури схем здійснює їх обробку, що передбачає реалізацію операцій фільтрації, аналізу структурних елементів схем, формування опису схем, візуалізацію, сегментацію та кодування отриманих даних щодо аналізованих об'єктів. Узагальнена структурна схема системи наведена на рис. 1.

На вхід системи надходить схемне зображення з текстового документу, що аналізується. Залежно від якості зображення та насиченості схеми здійснюється його попередня центроїдна фільтрація з метою видалення випадкових шумів (зайвих точок, що не є частиною інформації щодо структурних елементів зображення), після чого виділяються основні елементи. Далі реалізується центроїдна релаксація, що дозволяє виділити наявні на схемі криві лінії, кути, кола, інші замкнуті фігури, визначити їх геометричні характеристики для подальшого опису просторової структури схеми. Після релаксації оброблене зображення надходить до підсистеми лінійної сегментації, призначеної для обробки пересічних ліній, що утворюють вузли та сегменти. Сегментовані лінії та вузли піддаються кодуванню для подальшого кодованого представлення схеми сегментів. Водночас результати сегментації можуть використовуватися для візуалізації та графічного представлення опису схеми.

Розглянемо докладніше окремі блоки системи автоматизованого аналізу просторової структури схем. Центральним елементом цієї системи є підсистема лінійної сегментації, основною функцією якої є автоматизація процедури пошуку ліній та вузлів на растровому зображенні з подальшим кодованим представленням схеми сегментів. Структурна схема підсистеми лінійної сегментації наведена на рис. 2.

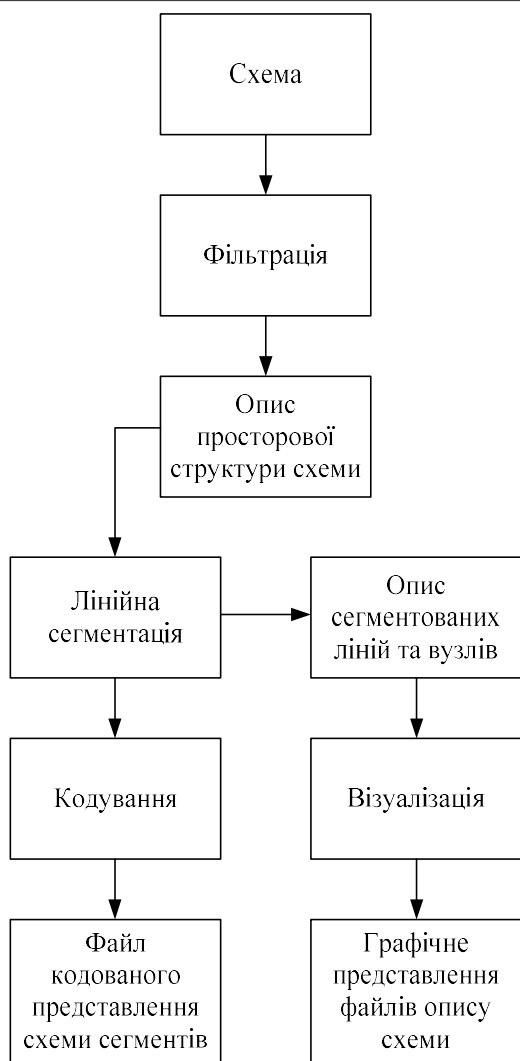


Рисунок 1 – Система аналізу просторової структури схем



Рисунок 2 – Структура підсистеми лінійної сегментації

В цій підсистемі використовуються такі структури даних:

а) формат вхідного масиву точок зображення:

$$\begin{matrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{matrix}, \quad (1)$$

де a_{ij} – елемент масиву точок, що відповідає точці зображення з координатами (i, j) та може приймати значення 0 або 1; $i = 1, \dots, m$; $j = 1, \dots, n$.

б) формат опису сегментів ліній:

$$\begin{matrix} n_1 : x_{11}, y_{11} [x_{12}, y_{12} [x_{13}, y_{13}]] \\ n_2 : x_{21}, y_{21} [x_{22}, y_{22} [x_{23}, y_{23}]] \\ \dots \\ n_k : x_{k1}, y_{k1} [x_{k2}, y_{k2} [x_{k3}, y_{k3}]] \end{matrix}, \quad (2)$$

де n_i – номер i -го сегмента лінії; x_{i1}, y_{i1} – координати першої точки i -го сегмента лінії; x_{i2}, y_{i2} – координати другої точки i -го сегмента лінії (указуються, якщо точка (x_{i2}, y_{i2}) є вузлом); x_{i3}, y_{i3} – координати напрямку кодування від точки (x_{i1}, y_{i1}) (указуються, якщо точки (x_{i1}, y_{i1}) та (x_{i2}, y_{i2}) є вузлами).

2.2. Алгоритми пошуку вузлів та сегментів

Процедура пошуку вузлів застосовується для визначення в межах схеми, що обробляється, елементів, які є областями перетину ліній (замкнутими фігурами). Ця процедура передбачає обхід масиву точок схеми з одночасним формуванням масиву елементів вузла, визначенням координат центра вузлів та загальної кількості вузлів. При цьому в масиві вузлів здійснюється нумерація елементів, що дозволяє визначити, до якого з вузлів належить конкретна точка. Отримані за результатами цієї процедури дані використовуються надалі для пошуку сегментів ліній, кодування ліній та визначення координат сегментів для їх графічного представлення. Вхідною інформацією процедури пошуку вузлів є масив точок з файлу, формат якого має вигляд (1). Розмір цього масиву відповідає розміру схемного зображення. Вихідною інформацією цієї процедури є масив вузлів схеми з зазначенням номера кожного вузла для їх подальшої ідентифікації.

На вхід процедури надходить схемне фільтроване зображення з текстового документу, що аналізується, у вигляді масиву точок, кожний елемент якого може приймати значення 0 або 1 (в разі відсутності чи наявності точки відповідно). Таким чином, структурні елементи схемного зображення

формується як сукупність точок зі значенням 1. Кожній одиниці зображення в масиві відповідає елемент масиву вузлів, значення якого формується таким чином:

- якщо значення елемента менше нуля, то цей елемент ще не є оброблений (це необхідно при обході масиву точок для виключення повторної обробки елементів);
- якщо значення елемента дорівнює нулю, то даному елементу не відповідає жоден з вузлів і відповідна точка в масиві точок не належить до вузла;
- якщо значення елемента більше нуля, то воно відповідає номеру вузла, до якого належить аналізована точка.

У загальному випадку зображення містить лінії одиничної товщини, тобто кожна точка окремої лінії може мати не більше двох сусідніх точок, що водночас визначають напрямок руху лінії. Лінію одиничної товщини можна описати таким чином:

$$S(x_k, y_k) = \{1, 2\}; k = \overline{1, N}, \quad (3)$$

де N – кількість точок в лінії; (x_k, y_k) – координати точки a_k ; $S(x_k, y_k)$ – кількість сусідніх з a_k точок.

Кількість сусідніх з a_k точок можна визначити так:

$$S(x_k, y_k) = \sum_{i=-1}^1 \sum_{j=-1}^1 C[x_k+i, y_k+j] - C[x_k, y_k], \quad (4)$$

де $C[x_k, y_k]$ – масив точок вхідного зображення.

Для розглянутого варіанта опису точки початку та кінця лінії мають лише одну сусідню точку, що однозначно визначає напрямок руху лінії. Всі інші точки лінії мають по дві сусідні точки (попередню та наступну). Це дозволяє уникнути надмірної інформації для опису лінії. Зображення, що містять лінії саме такого типу, можуть бути отримані на етапі попередньої фільтрації. За таких умов точки, що мають більше ніж дві сусідні, є вузловими та розглядаються як точки входу лінії до областей перетину ліній (замкнених фігур). Це можна відобразити таким чином:

$$B[x, y] = \begin{cases} n, & S(x, y) > 2, \\ 0, & S(x, y) \leq 2 \end{cases} \quad (5)$$

де $B[x, y]$ – масив вузлових точок для зображення розміру $M \times N$; n – номер вузла, що обробляється.

Пошук вузлових елементів передбачає послідовний розгляд елементів масиву точок. Однак при перетині ліній точки, що є сусідніми з вузлом, можуть бути більше ніж дві сусідні точки. Цієї неоднозначності можна запобігти за допомогою визначення центра вузла, координати якого мають бути отримані як середнє арифметичне усіх точок,

що належать до вузла. Зазначимо, що дві точки вважаються належними до одного вузла, якщо існує безперервний шлях з точок цього вузла, що їх з'єднує.

Алгоритм пошуку вузлів реалізовано у вигляді програмного модуля «SN (search nodes)»:

1. Початок;
2. Ініціалізація масиву вузлів;
3. $i=0; j=0; z=0$;
4. Якщо $j \geq N$, то перехід до п.11;
5. Якщо $i \geq M$, то перехід до п.10;
6. Якщо $(\text{apix}[i][j]=1)$ та $(\text{apix}2[i][j]<0)$ та $(\text{NC}(i,j)>2)$, то перехід до п.7, інакше до п.9;
7. $z=z+1$;
8. $\text{NS}(i,j,z)$;
9. $i=i+1$; перехід до п.5;
10. $i=0; j=j+1$; перехід до п.4;
11. Кінець.

У наведеному алгоритмі застосовано такі позначення: M, N – розміри вхідного зображення (схеми / діаграми); $(\text{apix}[i][j], \text{apix}2[i][j])$ – масиви точок та вузлів відповідно; $\text{NC}(\text{neig count})$ – функція обчислення кількості сусідніх точок; $\text{NS}(\text{node select})$ – функція пошуку вузлових точок; z – номер поточного вузла.

Процедура пошуку сегментів застосовується для визначення в межах схеми, що обробляється, сегментів ліній за результатами реалізації попередньої процедури пошуку вузлів та виділення структурних елементів зображення (схеми / діаграми). Це дозволяє отримати опис зображення схеми, що надалі кодується та обробляється в системі аналізу. Процедура лінійної сегментації передбачає виконання таких операцій:

- пошук окремих сегментів ліній, що не містять вузлів;
- пошук сегментів ліній, одна з початкових або кінцевих точок яких є вузлом;
- пошук сегментів ліній, обидві початкові та кінцеві точки яких є вузлами.

При цьому використовуються особливості, пов'язані з різними характеристиками структурних елементів графічного зображення. Формування результатів обробки визначається способом їх передавання до інших блоків системи (найчастіше, за допомогою ланцюгового кодування). Наприклад, у випадку опису сегментів ліній, повністю обмежених вузлами, виникає необхідність передавання додаткової координати для однозначного визначення напрямку руху при формуванні ланцюгового коду.

Вхідними даними для цієї процедури є: вхідний масив точок зображення (1); масив вузлів, що містить опис вузлових точок (їх області, центри та їх нумерацію для унікальної ідентифікації).

Пошук сегментів ліній здійснюється за припущенням їх одиничної товщини (лінії саме

такого типу можуть бути отримані на етапі попередньої фільтрації). Ця процедура передбачає обхід масиву точок схеми та аналіз їх можливої належності до масивів елементів вузлів, що дозволяє визначити, до якого із сегментів лінії належить конкретна точка. Кожній одиниці зображення в масиві відповідає елемент масиву сегментів, значення якого формується таким чином:

- якщо значення елемента менше нуля, то цей елемент ще не є оброблений (це необхідно при обході масиву точок для виключення повторної обробки елементів);

- якщо значення елемента дорівнює нулю, то даному елементу не відповідає жоден із сегментів ліній і відповідна точка в масиві точок не належить до цих сегментів;

- якщо значення елемента більше нуля, то воно відповідає номеру сегмента ліній, до якого належить аналізована точка.

Це можна описати таким чином:

$$D[x, y] = \begin{cases} n, & S(x, y) < 3, \\ 0, & S(x, y) \geq 3 \end{cases}, \quad x \in [1, M]; \quad y \in [1, N], \quad (6)$$

де (x, y) – координати аналізованої точки;

D – масив сегментів; n – номер сегмента, що обробляється.

Алгоритм пошуку сегментів реалізовано у вигляді програмного модуля «SS (search segs)»:

1. Початок;
2. Ініціалізація масиву сегментів;
3. $i=0$; $j=0$; ; $nsegs = 0$;
4. Якщо $j \geq N$, то перехід до п.11;
5. Якщо $i \geq M$, то перехід до п.10;
6. Якщо $(arix[i][j]=1)$ та $(arix1[i][j]<0)$ та $(NC(i,j)=1)$ або $(NC(i,j)=2)$, то перехід до п.7, інакше до п.9;
7. $nsegs = nsegs + 1$;
8. $NL(i,j, nsegs)$;
9. $i=i+1$; перехід до п.5;
10. $i=0$; $j=j+1$; перехід до п.4;
11. Кінець.

У наведеному алгоритмі застосовано такі позначення: M, N – розміри вхідного зображення (схеми / діаграми); $(arix[i][j], arix1[i][j])$ – масиви точок та сегментів відповідно; NC (neig count) – функція обчислення кількості сусідніх точок; NS (neig line) – функція обробки точок одного сегмента лінії; $nsegs$ – номер поточного сегмента.

2.3. Алгоритм кодування сегментів

Процедура кодування сегментів застосовується для формування кодів трьох типів сегментів ліній за результатами попереднього пошуку сегментів ліній та вузлів. При цьому обробляються такі типи сегментів ліній: сегменти, що не мають вузлів;

сегменти, одна з крайніх точок яких є вузлом; сегменти, обидві крайні точки яких є вузлами.

Вхідними даними для цієї процедури є: вхідний масив точок зображення (1); масив вузлів, що містить опис вузлових точок (їх області, центри та їх нумерацію для унікальної ідентифікації); масив сегментів, що містить опис сегментів ліній (точки, що належать лінії, координати їх перетину та їх нумерацію для унікальної ідентифікації).

При кодуванні сегментів ліній застосовується формат опису (2). Ця процедура передбачає пошук крайніх точок сегментів ліній, за якими визначається належність кожного з них до одного з трьох розглянутих вище типів. За результатами послідовного обходу координат зображення їх зі значеннями в масивах сегментів та вузлів однозначно визначається тип та координати сегмента, що обробляється, а також спосіб його кодування. Перевірка належності деякої точки зображення сегментам здійснюється таким чином: якщо цій точці відповідає додатне значення в масиві сегментів, то вона належить до сегмента з відповідним номером. Аналогічно перевіряється належність деякої точки зображення вузлам: якщо цій точці відповідає додатне значення в масиві вузлів, то вона належить до вузла з відповідним номером.

У свою чергу, порівняння масивів сегментів та вузлів дозволяє робити такі висновки:

- якщо точці зображення відповідає додатне значення в масиві сегментів та нульове значення в масиві вузлів, то ця точка належить до сегменту і не належить до вузлів;

- якщо ж ця точка має додатне значення в масиві сегментів та додатне значення в масиві вузлів, то вона належить більш ніж до одного сегмента.

При кодуванні сегментів ліній визначається тип сегмента (відповідно з визначеним типом формуються до шести координат точок опису сегмента).

Алгоритм кодування сегментів реалізовано у вигляді програмного модуля «CS (code segs)»:

1. Початок;
2. Ініціалізація масивів та змінних;
3. Якщо пошук вузлів завершено, то перехід до п.5, інакше до п.4;
4. Пошук вузлів (SN), перехід до п.3;
5. Якщо пошук сегментів завершено, то перехід до п.7, інакше до п.6;
6. Пошук сегментів (SS), перехід до п.5;
7. $i=0$; $j=0$; $lnum=0$; $liden=0$;
8. Якщо $j \geq N$, то перехід до п.15;
9. Якщо $j \geq M$, то перехід до п.14;
10. Якщо $(arix[i][j]=1)$ та $(arix1[i][j]<0)$ та $((NC(i,j)=1)$ або $(NC(i,j)=2))$, то перехід до п.11, інакше до п.13;
11. $lnum = lnum + 1$; $liden = liden + 1$;

12. LV (i,j, liden);
13. i=i+1; перехід до п.9;
14. i=0; j=j+1; перехід до п.8;
15. Кінець.

У наведеному алгоритмі застосовано такі позначення: M , N – розміри вхідного зображення (схеми / діаграми); $lnum$ – порядковий номер сегмента, що обробляється; $liden$ – ідентифікатор сегмента, що обробляється, в масиві сегментів; LV (line vect) – функція обробки точок сегмента та кодування його координат.

2.4. Підсистема порівняння схем та діаграм в текстових документах

Розглянутий метод аналізу та кодування схем та діаграм може бути застосований для порівняння цих графічних об'єктів і подальшого їх зберігання в базі даних системи виявлення дублікатів електронних документів.

У підсистемі порівняння схем та діаграм, що містяться в аналізованих текстових документах, можуть бути використані розглянуті вище алгоритми аналізу структури графічних об'єктів. Однією з функцій такої підсистеми є перевірка оригінальності графічного об'єкта, що аналізується (насамперед, схеми або діаграми UML), як за джерелами мережі Інтернет, так і за власними базами текстів з такими графічними об'єктами (статей, дисертацій, курсових, дипломних та магістерських робіт і т.ін.).

Припустимо, що згідно з використанням однієї з наявних систем пошуку текстів (без урахування можливої близькості наявних там схем або діаграм), що є близькими за сучасними критеріями близькості текстів або їх фрагментів, була сформована обмежена множина (A) об'єктів з високим ступенем подібності до текстового об'єкта X зі схемами або діаграмами, оригінальність якого аналізується. Як зазначалося вище, є різні варіанти представлення графічних об'єктів, що входять до документів з множини A. Втім слід зазначити, що лише частина з них застосовується в переважній більшості схем та діаграм текстових документів. Як правило, несумісні формати мають файли растрових та векторних зображень, хоча є формати, що дозволяють зберігати дані різних класів. Чимало додатків орієнтовані на власні «специфічні» формати, перенесення їхніх файлів в інші програми змушує використовувати спеціальні фільтри або експортувати зображення в «стандартний» формат. Зважаючи на особливості запропонованого вище методу аналізу та кодування схем та діаграм, зробимо наголос на варіанти представлення текстових об'єктів, що входять до множини A, коли вони містять графічні растрові зображення схем та діаграм (наприклад, у форматах GIF або JPEG). Таке представлення є достатньо

поширеним в текстових електронних документах, що мають науковий характер. У той же час є програмні засоби, які в деяких випадках дозволяють конвертувати схеми та діаграми інших форматів до стандартного растрового варіанта.

Розглянемо задачу порівняння L схем X_1, \dots, X_L з текстового об'єкта X зі схемами або діаграмами, оригінальність якого аналізується і який має високий ступінь подібності до текстових об'єктів множини A. Алгоритм такого порівняння полягає у такому:

- схеми X_1, \dots, X_L обробляються згідно з розглянутими алгоритмами структурного аналізу, що дозволяє сформувати її кодовані представлення X_{11}, \dots, X_{LL} ;
- аналогічно формуються кодовані представлення схем з текстових об'єктів, що входять до множини A (множина кодованих шаблонів Y);
- здійснюється перевірка повних збігів структури X_{11}, \dots, X_{LL} з кодованими шаблонами Y (процедура такої перевірки є тривіальною) та формується клас шаблонів Z, що відповідають випадкам повного збігу за результатами перевірки;
- у разі наявності однієї або кількох схем в множині Z, що мають повні структурні збіги зі схемними об'єктами у документі X, здійснюється перевірка збігу текстових написів і пояснень у схемах;
- у разі відсутності у документі X посилань на первинне джерело для випадків повного збігу (як за структурою схем, так і за відповідними текстовими поясненнями) документ X автоматично заноситься до електронної форми «Індикація можливості схемного плагіату», що створюється для кожного аналізованого об'єкта.

Для розглянутого алгоритму може бути передбачена також можливість виявлення часткового дублювання схемних зображень. Цей алгоритм може доповнювати загальну гіпотетичну процедуру повного автоматичного аналізу оригінальності документу, що перевіряється, з урахуванням подібностей тексту, формул, зображень, таблиць і т.ін.

Результатом реалізації запропонованої гібридної схеми може бути формування електронної форми «Індикація можливості плагіату схем та діаграм», що створюється для кожного аналізованого об'єкта, з визначенням загальних коефіцієнтів подібності графічних елементів, які перевіряються, та побудовою відповідної порівняльної гистограми.

Висновки

У статті наведено аналіз сучасних графічних форматів та редакторів, які використовуються для растрових схематичних зображень в наукових текстах, та особливості топологічної структури таких графічних об'єктів, як схеми та діаграми. Це дозволило випрацювати підхід до виявлення

повних або часткових дублікатів цих об'єктів у публікаціях, курсових, дипломних або магістерських проектах тощо.

Підхід, що пропонується, передбачає на першому етапі проведення аналізу схем алгоритмів та діаграм за їх просторовою топологією, тобто за особливістю структури зв'язків між вузлами схеми або діаграми. Система автоматизованого аналізу просторової структури схем здійснює їх обробку, що передбачає реалізацію операцій фільтрації, аналізу структурних елементів схем, формування опису схем, візуалізацію, сегментацію та кодування отриманих даних щодо аналізованих об'єктів.

Система передбачає можливість аналізу подібностей та виявлення повного або часткового дублювання схем та діаграм за результатами

кодованого представлення їх топологічної структури, що відображує зв'язки між вузлами, а також перевірки збігів текстових написів і пояснень в схемах та діаграмах. Розглянутий метод автоматичного аналізу та кодування схем та діаграм може бути застосований для порівняння цих графічних об'єктів та подальшого їх зберігання в базі даних системи виявлення дублікатів електронних текстових документів.

Отримані результати можуть бути, зокрема, використані для розширення функціональних можливостей наявних систем виявлення плагіату в наукових текстах, що містять такі поширені різновиди графічних об'єктів, як схеми та діаграми (насамперед, діаграми UML).

Список літератури

1. Білолицький А.О. Оптимізація системи пошуку збігів за допомогою використання алгоритмів локально чутливого хешування наборів текстових даних [Текст] / А.О. Білолицький, О.В. Діхтяренко // Управління розвитком складних систем. – 2014. – № 19. – С. 113 – 117.
2. Білолицький А.О. Метод вилучення помилкових збігів текстів в електронних документах [Текст] / А.О. Білолицький, С.Д. Криштоф, С.В. Білолицька, О.В. Діхтяренко // Управління розвитком складних систем. – 2015. – № 22(1). – С. 144 – 150.
3. Лізунов, П. П. Гібридний підхід до аналізу та розпізнавання математичних формул з метою виявлення в них подібностей [Текст] / П. П. Лізунов, А. О. Білолицький, Л. Е. Чала, С.В. Білолицька, О. Ю. Кучанський, С. Г. Удовенко // Управління розвитком складних систем. – 2016. – № 27. – С. 145 – 155.
4. Михайловський Ю.Б. Система Anti-Plagiarism як інструмент запобігання плагіату в навчальній та науковій діяльності [Текст] / Ю.Б. Михайловський, Н.А. Длугунович // Вісник Хмельницького національного університету. Технічні науки. – 2013. – № 3. – С. 162–168.
5. Лупаренко Л. А. Інструментарій виявлення плагіату в наукових роботах: аналіз програмних рішень [Текст] / Л.А. Лупаренко // Інформаційні технології і засоби навчання. – 2014. – Т. 40. – №2. – С. 151 – 169.
6. Шарапова Е.В. Исследование возможностей системы «Антиплагиат» для обнаружения заимствований [Текст] / Е.В. Шарапова // Перспективы науки и образования. – 2013 – №3. – С. 215 – 218.
7. Sheno M. Automatic Plagiarism Detection Using Similarity Analysis [online] / M. Sheno, K. C. Shet, U.D. Acharya // Advanced Computing: An International Journal. – 2012. – № 3 (3). – P. 59 – 62.
8. Чала Л. Э. Поиск неполных дубликатов в системах анализа цифровых изображений [Текст] / Л. Э. Чала, П. Ю. Попаденко // Вісник Кременчуцького національного університету імені Михайла Остроградського. – 2014. – Вип. 5. – С. 42-47.
9. Боровик В.М. Програмні компоненти проектування діаграм UML [Текст] / В.М. Боровик, О.І. Труш, О.М. Крива // Проблеми інформатизації та управління. – 2010. – № 3(31). – С. 14 – 19.
10. Гороховатский В.А. Исследование мер структурного соответствия компонентных объектов [Текст] / В.А. Гороховатский // Системы обработки информации. – 2009. – Вип. 2 (76). – С. 36 – 42.
11. Обробка графічної інформації [Текст] : навч. посібник / В. А. Романюк, О.М. Сальніков, В. Г. Малюк та ін.; під заг. ред. В. А. Романюка. – Х. : Акад.ВВ МВСУ, 2013. – 112 с.
12. Вдовин А.М. Исследование планарных элементов пространственной структуры изображений [Текст] / А.М. Вдовин, Б.С. Хаба, А.И. Мурынов, В.Е. Лялин // Химическая физика и мезоскопия. – 2001. – Т.3, №2. – С.134 – 147.

Стаття надійшла до редколегії 14.11.2016

Рецензент: д-р техн. наук, проф. С.Д. Бушуєв, Київський національний університет будівництва і архітектури, Київ.

Лизунов Петр Петрович

Доктор технических наук, профессор, заведующий кафедрой основ информатики, orcid.org/0000-0003-2924-3025

Киевский национальный университет строительства и архитектуры, Киев

Белошицкий Андрей Александрович

Доктор технических наук, профессор, заместитель декана факультета информационных технологий, orcid.org/0000-0001-9548-1959

Киевский национальный университет им. Т. Шевченко, Киев

Киевский национальный университет им. Т. Шевченко, Киев

Чалая Лариса Эрнестовна

Кандидат технических наук, доцент кафедры информационных технологий, orcid.org/0000-0002-9890-4790

Харьковский национальный университет радиоэлектроники, Харьков

Белошицкая Светлана Васильевна

Кандидат технических наук, доцент кафедры информационных технологий проектирования и прикладной математики, orcid.org/0000-0002-0856-5474

Киевский национальный университет строительства и архитектуры, Киев

Киевский национальный университет строительства и архитектуры, Киев

Кучанский Александр Юрьевич

Кандидат технических наук, доцент кафедры информационных технологий, orcid.org/0000-0003-1277-8031

Киевский национальный университет строительства и архитектуры, Киев

Удовенко Сергей Григорьевич

Доктор технических наук, заведующий кафедрой информатики и компьютерной техники, orcid.org/0000-0001-5945-8647

Харьковский национальный экономический университет им. С. Кузнеця, Харьков

АВТОМАТИЧЕСКИЙ АНАЛИЗ ПОДОБИЯ СХЕМ И ДИАГРАММ В ЭЛЕКТРОННЫХ ТЕКСТОВЫХ ДОКУМЕНТАХ

Аннотация. Распространенным типом графических объектов, тематически связанных с проблемой разработки и использования технологий обработки данных в сложных системах различного функционального назначения, являются схемы преобразования информации (в первую очередь, схемные отображения алгоритмов системы) и специализированные диаграммы. Для анализа подобия схем и диаграмм, содержащихся в текстовых документах, целесообразно использовать не только оценивание характеристик графических изображений, применяя стандартные методы их сравнения, но и учитывать особенности построения схемных изображений. Поэтому для нахождения полных или неполных дубликатов таких графических объектов предлагается подход, предусматривающий использование результатов анализа их топологической структуры.

Ключевые слова: **схемы и диаграммы; сегментация; кодирование; дубликат; графические форматы**

Lizunov Petro

DSc(Eng.), Professor, orcid.org/0000-0003-2924-3025

Kyiv National University of Construction and Architecture, Kyiv

Biloshchytskyi Andrii

DSc (Eng), Professor, Deputy Dean of the Faculty of Information Technology, orcid.org/0000-0001-9548-1959

Taras Shevchenko National University of Kyiv, Kyiv

Chala Larysa

PhD (Eng.), assistant professor of Artificial Intelligence Deptment, orcid.org/0000-0002-9890-4790

Kharkiv National University of Radioelectronics, Kharkiv

Biloshchytska Svitlana

Ph.D., assistant professor of information technology designing and applied mathematics, orcid.org/0000-0002-0856-5474

Kyiv National University of Construction and Architecture, Kiev

Kuchansky Alexander

PhD(Eng.), assistant professor of Information Technology Department, orcid.org/0000-0003-1277-8031

Kyiv National University of Construction and Architecture, Kyiv

Udovenko Serhii

DSc(Eng.), Head of Informatics and Computer Technique Deptment, orcid.org/0000-0001-5945-8647

Simon Kuznets Kharkiv National University of Economics, Kharkiv

AUTOMATIC ANALYSIS OF SIMILARITY OF SCHEMAS AND DIAGRAMS IN ELECTRONIC TEXT DOCUMENTS

Abstract. The widespread type of the graphic objects thematically related to the problem of development and use of technologies of processing of data in the difficult systems of the different functional setting are schemas of transformation of information (first of all, scheme reflections of algorithms of the system) and specialized diagrams. For the analysis of similarity of

the charts and diagrams, contained in text documents, it is expedient to use not only the evaluation of descriptions of graphic images, applying the standard methods of their comparison but also to take into account the features of construction of scheme images. Therefore for being of complete or incomplete duplicates of such graphic objects offered approach, envisaging drawing on the results of analysis their topological structure.

Keywords: *schemas and diagrams; segmentation; codage; duplicate; graphic formats*

References

1. Biloshchytskyi, A. & Dikhtiarenko, O. (2014). Optimization of Matching algorithms by using local-sensitive hash sets of text data. *Management of complex systems*, 19, 113 – 117.
2. Biloshchytskyi, A., Kristof, S., Biloshchytska, S. & Dikhtiarenko, O. (2015). The method of elimination of erroneous coincidences text in electronic documents. *Management of Development of Complex Systems*, Issue 22 (1), 144 – 150.
3. Lizunov, Petro, Biloshchytskyi, Andrii, Chala, Larysa, Biloshchytska, Svitlana, Kuchansky, Alexander, & Udovenko, Serhii. (2016). Hybrid approach to analysis and recognition of mathematical formulas to identify their similarity. *Management of Development of Complex Systems*, 27, 145–155.
4. Myhaylovskiy, Yu. & Dluhunovych, N. (2013). Anti-Plagiarism System as a Tool for Plagiarism Preventing in Educational and Research Activities. *Journal of Khmelnytskyi National University*, 3, 162–168.
5. Lupanenko, L.A. (2014). Plagiarism detection tools for research works: analysis of software solutions. *Information Technology and Learning Tools*, Vol 40, 2, 151–169.
6. Sharapova, E. (2013). Investigation of possibilities "Anti-plagiarism" system to detect borrowing. *Prospects for Science and Education*, 3, 215–218.
7. Shenoy, M., Shet, K., Acharya, U. (2012). Automatic Plagiarism Detection Using Similarity. *Advanced Computing: An International Journal*, 3 (3), 59–62.
8. Chala, L. & Popadenko, P. (2014). Search partial duplicates in digital image analysis systems. *Journal of Kremenchuk National University*, 5, 42–47.
9. Borovyk, V., Trush, O. & Kryva, O. (2010). Software components design UML diagrams. *Problems of information and management*, 3(31), 14–19.
10. Gorohovatskyi, V. (2009). Research measures the structural component matching objects. *Information processing systems*, 2(76), 36–42.
11. Romaniuk, V., Salnikov, O., Maliuk, V. etc. (2013). *Graphics processing*, 112.
12. Vdovin, A., Khaba, B., Murynov, A. & Lyalin, V. (2001). The study of planar elements of the spatial image structure. *Chemical physics and mezoscopy*, 3(2), 134–147.

Посилання на публікацію

- APA Lizunov, Petro, Biloshchytskyi, Andrii, Chala, Larysa, Biloshchytska, Svitlana, Kuchansky, Alexander, & Udovenko, Serhii. (2016). Automatic analysis of similarity of schemas and diagrams in electronic text documents. *Management of Development of Complex Systems*, 28, 160 – 169.
- ГОСТ Лізунов, П. П. Автоматичний аналіз подібностей схем та діаграм в електронних текстових документах [Текст] / П. П. Лізунов, А. О. Білощицький, Л. Е. Чала, С. В. Білощицька, О. Ю. Кучанський, С. Г. Удовенко // *Управління розвитком складних систем*. – 2016. – № 28. – С. 160 – 169.