УДК 004.67

О. М. Верес, Я. П. Кісь, В. А. Кугівчак, І. В. Рішняк
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

ВИБІР МЕТОДІВ ДЛЯ ПОШУКУ ОДНАКОВИХ АБО СХОЖИХ ЗОБРАЖЕНЬ

© Верес О. М., Кісь Я. П., Кугівчак В. А., Рішняк І. В., 2018

Досліджено методи аналізу зображень. Описано методи індексації зображень для пошуку дублікатів зображень, а також методи пошуку схожих зображень, які ґрунтуються на визначенні ключових точок. Створено прототип системи та виконано тестування описаних методів. Результат аналізу став основою для проекту інформаційної системи реверсного пошуку схожих або ідентичних зображень.

Ключові слова: аналіз, детектор, дескриптор, зображення, ключова точка, метод, піксель, хешування.

The article describes the research of image analysis methods. The methods of indexing images for the search of duplicate images, as well as methods for finding similar images based on the definition of key points are described. The prototype of the system was created, and testing of the described methods was carried out. The result of the analysis became the basis for the information system project of reverse search of similar or identical images.

Key words: analysis, detector, descriptor, image, key point, method, pixel, hashing.

Вступ. Загальна постановка проблеми

Графічні зображення є не менш важливою складовою, ніж текст, а інколи без них неможливо розкрити тему. Крім того, деякі типи зображень самі по собі є об'єктами авторського права і захищені законом України про захист авторських прав. Але це ніяк не заважає копіювати чи сканувати зображення і видавати їх за свої. Для того щоб знайти дублікати чи запозичення зображень у документах, необхідно визначити, які саме графічні елементи вважаються схожими. Очевидно, що такими є повні дублікати, які можуть, своєю чергою, бути зменшені чи розтягнуті. Копіюючи чужі зображення, плагіатор може вдатися до різних прийомів, але основне його завдання, як і за інших видів запозичення, – зробити зображення візуально не подібним до оригіналу, зберігши його інформативну цінність. До таких модифікацій можна зарахувати зміну яскравості, контрасту, зменшення гами кольорів (переведення зображення у відтінки сірого) тощо. Серед модифікацій, які впливають на інформативність зображення, але також можуть використовуватися в деяких випадках, – обрізання зображення або склеювання з кількох елементів у один.

З одного боку, такі зображення не є запозиченнями, хоча вони і повністю ідентичні, а з іншого – цінність зображення може полягати саме в контексті його використання, якщо автор серед можливої множини рішень використав саме цю ілюстрацію. Комп'ютерна програма не зможе оцінити вміст зображення та зробити висновок щодо ліцензії, під якою розповсюджується це зображення, тому остаточне рішення повинен приймати експерт, який перевіряє роботу за допомогою програми.

Аналіз останніх досліджень і публікацій

Серед підходів до обробки графічної інформації можна виділити два основних напрями: визначення ключових точок зображення та використання локально чутливого хешування. Ці методи можуть бути поєднані й загалом дають добрі результати у разі пошуку схожих зображень. Спочатку визначають ключові точки на зображенні, а потім ділять зображення на дрібні фрагменти. Виконуючи індексацію кожного фрагмента окремо, отримують масив сигнатур, який відповідає за зображення загалом. Використовуючи міру Хеммінга [1], знаходили однотипні зображення, навіть у разі 90 % обрізання зображення. Описаний спосіб охоплює максимальну кількість можливих

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

модифікацій, яких може зазнати зображення. Однак є одна проблема – велика ймовірність хибних результатів. Спосіб знаходить зображення, яке частково схоже на задане, а не є дублікатом, з максимально можливою точністю.

А. О. Білощицький і О. В. Діхтяренко [2] розробили власний спосіб визначення ключових особливостей зображення. На відміну від визначення ключових точок, у цьому випадку головні риси зображення описували за допомогою векторів. Отримані набори векторів були основою для створення сигнатури зображення, для хешування використано локально чутливу функцію minHash. Метод названо min-Hash and tf-idfWeighting. Основне його завдання – це швидке виявлення схожих зображень у великих масивах даних. Метод знаходить схожі зображення, навіть якщо це різні зображення одного предмета, але помилкових спрацювань також багато.

Найпопулярнішими є три методи індексації зображень для пошуку дублікатів зображень, а саме: *хешування за середнім значенням* (Average Hash); *хеш за різницею* (Difference hash); *перцептивний хеш* (Perceptual Hash).

Для пошуку схожих зображень використовують спосіб виділення ключових точок. *Ключова точка*, або *точкова особливість зображення*, – це точка, розміщення якої виділяється на фоні будь-якої іншої точки. Як особливість точки зображення для більшості сучасних алгоритмів беруть квадратне вікно, розмір якого становить 5 на 5 пікселів. Визначення цих точок на зображенні досягається методом використання детектора і дескриптора. *Детектор* – це метод визначення ключової точки, що виділяє її на фоні зображення, а дескриптори повинні забезпечувати інваріантність знаходження відповідності між ключовими точками щодо перетворень зображень. *Дескриптор* – це метод, котрий дає змогу вилучати ключові точки обидвох зображень та порівнювати їх між собою. У випадку модифікацій об’єктів дослідження детектор допомагає знайти ті самі ключові точки на обидвох об’єктах [3].

Ключові точки повинні мати низку особливостей [4]: *відмінність* – кожна точка має явно відрізнятися від інших та бути унікальною у своїй області; *інваріантність* – визначення ключової точки повинно бути незалежним від афінних перетворень; *стабільність* – виділення таких особливостей повинно бути стійким до шумів та модифікацій; *інтерпретація* – ключові точки повинні виділятися так, щоб їх можна було використовувати для аналізу відповідностей і видобування на їх основі необхідної інформації.

Отже, для того щоб знайти фрагменти зображення або ж схожі за змістом ілюстрації, необхідно експериментувати з методами визначення ключових точок, у кожного з яких також свій набір переваг та недоліків.

Основні методи, які використовують під час побудови детекторів та дескрипторів, – **FAST** (*Features from Accelerated Segment Test*) [5]; **SIFT** (*Scale Invariant Feature Transform*) [6]; **ORB** (*Oriented FAST and Rotated BRIEF*) [7, 8]; **AKAZE** (*Accelerated KAZE*) [9]; **BRIEF** (*Binary Robust Independent Elementary Features*) [10]; **BRISK** (*Binary Robust Invariant Scalable Keypoints*) [11].

Сьогодні є багато систем для розпізнавання зображень. Найпопулярніші з них: **TinEye**, **Google Similar Images**, **Яндекс. Картинки**, **AntiDupl. NET**. Загальним недоліком цих систем є неможливість завантаження галереї зображень та створення власної бази даних (БД) для роботи.

Не вирішені раніше частини загальної проблеми. Головні вимоги до способу пошуку однакових зображень, точніше, до результатів його роботи, – максимальна точність і мінімум помилок. Інформаційна система повинна не тільки знаходити всі явні дублікати (ті, в яких змінені лише кольори, розміри чи формат), але й “подібні” зображення, мінімізувавши обсяг роботи для оператора системи. Система також повинна знаходити зображення, які зазнали модифікацій, а саме: поворот; віддзеркалення; зміна кольорів; обрізання зображення.

Сьогодні немає такої інформаційної системи аналізу зображень, котра б могла ідеально визначити однакові або схожі зображення на основі власноруч створеної БД зображень. Тому актуальним завданням є проектування інформаційної системи реверсного пошуку зображень. Інформаційна система реверсного пошуку зображень призначена для порівняння об’єктів з наявними в базі даних ідентичними або схожими об’єктами для виявлення інформації про власників цих ілюстрацій або ж для пошуку їхніх модифікованих версій, щоб унеможливити плагіат у дослідницьких роботах користувачів.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Цілі (завдання) статті

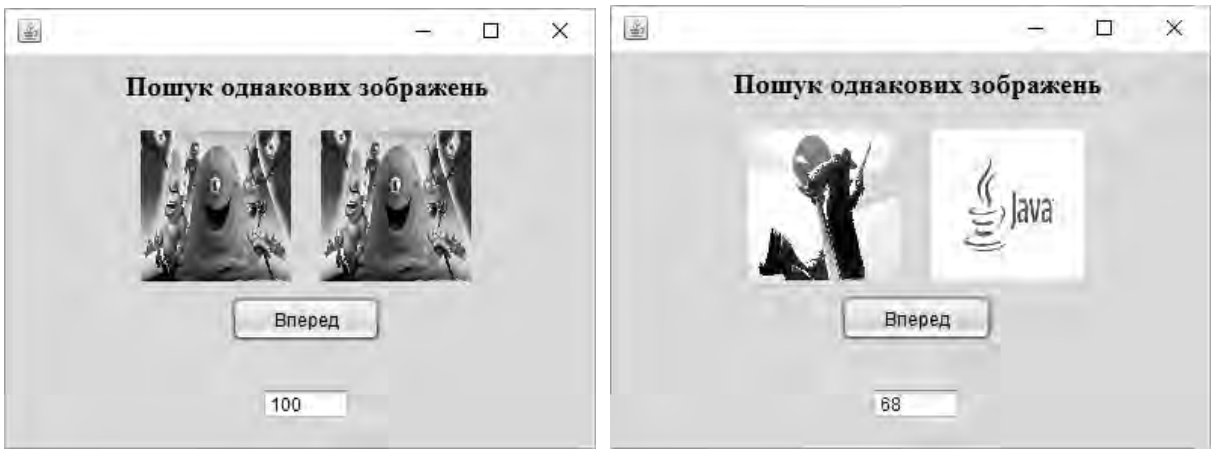
Для вибору методів аналізу та пошуку однакових зображень потрібно дослідити методи індексації зображень; визначення міри подібності; методи, які використовують для побудови детекторів та дескрипторів, а також проаналізувати ефективність методів для зображень, які зазнали модифікацій. Результат проведених досліджень є підґрунтям для проектування інформаційної системи реверсного пошуку зображень.

Аналіз методів, що працюють з контрольними точками

Для проведення дослідження створено тестові модулі програмної реалізації прототипу інформаційної системи реверсного пошуку зображень на основі власної вибірки даних [12–18].

Для отримання саме однакових, а не схожих зображень проаналізовано метод хешування за середнім значенням, а для визначення міри подібності застосовано відстань Хеммінга. Головна мета завдання дослідження полягає у визначенні порогової функції, котра дасть змогу стверджувати, що зображення є повними дублікатами.

Вибираємо картинку з наявних у базі даних і подивимося на результат (рис. 1).



а *б*

Рис. 1. Результат опрацювання зображень: а – однакові; б – різні

На рис. 1 праворуч – зображення користувача, а поряд – знайдене у базі даних зображення, нижче у вікні вказано відстань Хеммінга у відсотках. Для однакових зображень (рис. 1, *а*) відстань Хеммінга 100 %, це означає, що зображення знайдено і воно цілком ідентичне. У випадку різних зображень в результаті роботи програми знайдено найподібніше зображення, але відсоток подібності знизився аж на 32 %. Це свідчить про те, що зображення не можна розглядати як однакові.

Для того щоб знаходити схожі за змістом зображення, здійснимо порівняльний аналіз методів, котрі працюють із ключовими точками, а саме: ORB, BRISK, AKAZE, FAST відповідно на основі результатів класифікатора. Розмір вхідних зображень розглянуто у стисненому вигляді до 128, 256 та 512 пікселів з кожної сторони. Вхідні зображення поділено на три групи: 30 зображень із великою кількістю деталей (табл. 1); 30 зображень зі знімком монітора (табл. 2); 30 портретних фотографій людей (табл. 3). У колонці “Затрати” таблиць подано середній час на пошук однієї ключової точки та розрахунок її дескриптора.

Аналіз результатів першої групи. Усі зображення цієї групи містять численні деталі, розташовані у різних місцях. Відомості про оцінки методів для різних розширень ілюстрацій подано в табл. 1.

Найбільшу кількість ключових точок знайдено за допомогою методу BRISK, ця кількість зростає у геометричній прогресії. Відповідно, якщо зростає роздільна здатність досліджуваного зображення, необхідно значно більше часу для його опрацювання. Метод ORB виявився не надто чутливим до зміни розміру зображення у вибраних межах, його складність зростає у арифметичній прогресії. Найменший час виконання дескриптора у методі AKAZE. Метод FAST витрачає найменше часу на загальний пошук схожих зображень.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Таблиця 1

Оцінка методів для зображень першої групи

№ з/п	Розмірність вхідних зображень, пікселів	Метод	Загальна кількість знайдених ключових точок	Загальна кількість часу, витраченого на пошук ключових точок, мс	Час роботи дескриптора, мс	Витрати	Загальна тривалість роботи, с
1	128x128	ORB	10444	247	5199	0,5214	18
2	128x128	BRISK	11768	12496	12533	2,1269	20
3	128x128	AKAZE	5041	972	11128	0,4166	12
4	128x128	FAST	6568	144	4141	0,6524	12
5	256x256	ORB	12311	429	6129	0,5327	20
6	256x256	BRISK	26767	13096	12577	0,9591	44
7	256x256	AKAZE	7286	1872	1930	0,5218	18
8	256x256	FAST	15568	396	5643	0,3879	15
9	512x512	ORB	15719	602	7626	0,5234	22
10	512x512	BRISK	78395	14087	12683	0,3415	58
11	512x512	AKAZE	8688	2777	3541	0,7272	25
12	512x512	FAST	32210	801	8111	0,2767	17

Аналіз результатів другої групи

Візьмемо 30 ілюстрацій знімка монітора, кожна з яких подаватиме зображення у різних вікнах різних програм. Виконаємо аналіз цієї групи для різних розширень ілюстрацій (табл. 2).

Таблиця 2

Оцінка методів для зображення другої групи

№ з/п	Розмірність вхідних зображень, пікселів	Метод	Загальна кількість знайдених ключових точок	Загальна кількість часу на пошук ключових точок, мс	Час роботи дескриптора, мс	Витрати	Загальна тривалість роботи, с
1	128x128	ORB	1409	27	422	0,3187	12
2	128x128	BRISK	2178	2917	3014	2,7231	15
3	128x128	AKAZE	995	202	316	0,5206	8
4	128x128	FAST	1024	44	623	0,6514	9
5	256x256	ORB	1661	47	497	0,3278	13
6	256x256	BRISK	4954	3057	3025	1,2276	33
7	256x256	AKAZE	1438	389	541	0,6465	12
8	256x256	FAST	2427	121	849	0,3996	11
9	512x512	ORB	2121	66	619	0,3229	15
10	512x512	BRISK	14509	3288	3050	0,4369	44
11	512x512	AKAZE	1715	577	992	0,915	17
12	512x512	FAST	5022	245	1220	0,2917	13

Кількість ключових точок у сумі всіх зображень істотно зменшилася порівняно з першою групою. Це вплинуло на час роботи програми, дескриптора і на витрати. Відповідно, що менше ключових точок генерує будь-який алгоритм, то менше часу він витрачає на їхнє опрацювання. Всі витрати часу пропорційно залежать від кількості ключових точок. Результати алгоритмів практично не відрізняються від попередньої групи, це свідчить, що їхня робота не залежить від вхідних даних.

Аналіз результатів третьої групи. Для останньої групи відібрано портретні знімки 30 осіб (табл. 3).

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Таблиця 3

Оцінка методів для зображень третьої групи

№ з/п	Розмірність вхідних зображень, пікселів	Метод	Загальна кількість знайдених ключових точок	Загальна кількість часу на пошук ключових точок, мс	Час роботи дескриптора, мс	Витрати	Загальна тривалість роботи, с
1	128x128	ORB	5306	123	2516	0,3761	13
2	128x128	BRISK	5504	6082	6135	1,914	14
3	128x128	AKAZE	2382	463	570	0,3698	8
4	128x128	FAST	2996	74	1880	0,5145	8
5	256x256	ORB	6254	213	2966	0,5083	20
6	256x256	BRISK	12518	6375	6157	1,0011	44
7	256x256	AKAZE	3443	892	975	0,5424	18
8	256x256	FAST	7101	204	2562	0,3895	15
9	512x512	ORB	8864	332	4097	0,4996	22
10	512x512	BRISK	37030	6925	6271	0,3564	58
11	512x512	AKAZE	4557	1469	1986	0,7582	25
12	512x512	FAST	16309	458	4087	0,2787	17

Середня кількість згенерованих ключових точок більша за таку саму середню кількість точок у другій групі та менша за першу. Якщо брати кожен алгоритм окремо, то їхні результати відносно кожної групи пропорційні. Отже, час роботи програми залежить від вибраного методу та кількості виявлених ним ключових точок.

Виконаємо дослідження алгоритмів на виявлення схожості різних зображень, а також перевіримо на наявність помилок у роботі кожного з методів. Для цього створимо три групи зображень: однакові зображення, схожі зображення та різні зображення. Кожна з цієї групи містить по два тести для перевірки на достовірність роботи кожного з алгоритмів.

Однакові зображення: тест 1 – два повністю однакових зображення (рис. 2, *а*); тест 2 – два повністю однакових зображення (рис. 2, *а* і *б*), одне з яких повернуто на 90 градусів (рис. 2, *б*).

Схожі зображення: тест 1 – два схожих зображення (рис. 2, *а* і *в*), одне з яких сфотографоване з іншого ракурсу (основний об'єкт дослідження – книга, розташована ближче) (рис. 2, *в*); тест 2 – два схожих зображення (рис. 2, *в* і *г*), одне з яких сфотографовано з іншого ракурсу (рис. 2, *в*), основний об'єкт дослідження розміщений під кутом 45 градусів (рис. 2, *г*).

Різні зображення: тест 1 – два зовсім різні зображення, одне з яких – ілюстрація сайта радіо "Радій", а інше – головна сторінка сайта Львівської політехніки; тест 2 – два зовсім різні зображення, одне з яких – фото Національного університету "Львівська політехніка" (рис. 3, *а*), а інше – Львівського національного університету імені Івана Франка (рис. 3, *б*).

Для кожного із зображень визначено ключові точки та створено дескриптор, щоб вилучити ці точки та порівняти їх між собою на тотожність.

Результати проведених тестувань подано у табл. 4. Основні параметри результатів тестів: кількість ключових точок; кількість схожих ключових точок на двох зображеннях; відсоток спільних ключових точок – коефіцієнт подібності між двома зображеннями.

Метод ORB чудово справився із усіма тестами, адже відсоток спільних ключових точок зменшується відповідно для менш схожих між собою зображень. Кількість отриманих ключових точок практично однакова у кожному з експериментів, з огляду на це, можна стверджувати про хорошу стабільність цього методу. У відсотковому співвідношенні метод AKAZE показує результати на рівні ORB, але кількість згенерованих ним ключових точок значно менша та нерівномірна, тому можна сказати, що метод стабільний у результатах, але непередбачуваний щодо кількості створених основних точок зображення. Результати проходження тестів методом BRISK свідчать, що цей алгоритм також виконав своє завдання, але у співвідношенні з розглянутими вище

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

методами показав гірші результати у знаходженні схожих та однакових зображень, проте зміг чітко виділити різні ілюстрації у тестах. Кількість ключових точок нестабільна і зростає зі збільшенням кількості деталей на зображенні. Метод FAST – один з лідерів за швидкістю виявлення ключових точок та розрахування для них значень дескрипторів, але не впорався із тестами, і хоча кількість його ключових точок значно більша, ніж у попередників, це не дало йому змоги розпізнати однакові зображення, повернуті на 90 градусів, та схожі зображення у разі повороту на 45 градусів.

а *б* *в* *г*

Рис. 2. Однакові зображення: а – оригінальне; б – обернене на 90 градусів; в – змінений ракурс; г – обернене на 45 градусів

а *б*

Рис. 3. Різні зображення для тесту 2: а – Львівська політехніка; б – Львівський національний університет імені Івана Франка

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Таблиця 4

Результати тестувань

Метод	Ключові точки	Однакові зображення				Схожі зображення				Різні зображення			
		Тест 1		Тест 2		Тест 1		Тест 2		Тест 1		Тест 2	
		Фото 1	Фото 2	Фото 1	Фото 2	Фото 1	Фото 2	Фото 1	Фото 2	Фото 1	Фото 2	Фото 1	Фото 2
ORB	Кількість	416	416	416	415	416	419	416	396	423	416	455	454
	Кількість спільних	416		406		167		73		27		27	
	Подібність, %	100		98		40		19		6		6	
AKAZE	Кількість	33	33	33	33	33	36	33	35	114	129	326	376
	Кількість спільних	33		32		13		8		7		16	
	Подібність, %	100		97		36		23		5		4	
BRISK	Кількість	363	363	363	373	363	389	363	492	369	1009	1330	1444
	Кількість спільних	361		333		160		72		11		26	
	Подібність, %	99		89		41		15		1		2	
FAST	Кількість	566	566	566	566	566	560	566	658	411	742	1777	1863
	Кількість спільних	566		23		282		33		27		39	
	Подібність, %	100		4		50		5		4		2	

Аналіз результатів досліджень дає змогу сформулювати вимоги та виконати проектування інформаційної системи реверсного пошуку зображень. Для пошуку однакових зображень використовується метод хешування за середнім значенням. Для обчислення міри подібності – відстань Хеммінга. Щоб визначити ключові точки, застосовують детектор, для пошуку схожих ключових точок – дескриптор. Для цього використовують алгоритм ORB.

Проектування системи виконано засобами структурного моделювання. Для кращого розуміння взаємних зв'язків між системою та зовнішнім середовищем побудовано діаграми потоків даних [12]. Програмну компоненту системи реалізовано засобами мови програмування [13–18]: Java – основна мова для розроблення бізнес-логіки та користувацького інтерфейсу, а також SQL – для роботи з базою даних.

Розроблена інформаційна система надає користувачеві можливість здійснити вибірку зображень на основі вхідних даних, виконати їхній перегляд, знайти однакові та схожі зображення, додати нові зображення до бази даних, переглянути дані про власника ілюстрації, щоб визначити плагіат.

Висновки

Для розроблення проекту інформаційної системи реверсного пошуку зображень ми дослідили порогову функцію для пошуку дублікатів, використовуючи хешування за середнім значенням і міру Хеммінга. На основі результатів експериментів можна стверджувати, що порогову функцію необхідно вибирати між 68–88 %, відповідно чим менший цей показник, тим більше він визначатиме саме схожі (на основі середніх кольорів) зображення, а не повні дублікати. Протестовано також алгоритми пошуку схожих зображень на основі ключових точок. Головним елементом цього дослідження був час, витрачений на знаходження ключових точок та порівняння їх на схожість методами: ORB, BRISK, AKAZE та FAST. Найгіршим виявився алгоритм BRISK, адже кількість згенерованих ним точок дуже велика, що призвело до стрімкого збільшення часу опрацювання. Експериментально виявлено, що розмір зображення 256×256 пікселів найоптимальніший для його опрацювання.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

У другому дослідженні увагу зосереджено на визначенні того, у котрого із алгоритмів найменша кількість похибок спрацювання. Для цього було створено групи однакових зображень, схожих зображень та зовсім різних зображень. Не справився із цим завданням алгоритм FAST, тому, попри його найкращі результати в обробці зображень, цей метод не можна використовувати. Найкращі результати тестування за усіма показниками в алгоритмів ORB та AKAZE.

За результатами проведених тестувань для реалізації у інформаційній системі реверсного пошуку зображень вибрано метод ORB, адже він генерує значно більше ключових точок за одиницю часу, аніж метод AKAZE.

Вибраний метод реалізовано в прототипі інформаційної системи реверсного пошуку зображень. Система призначена для порівняння об'єктів з наявними в базі даних ідентичними або схожими об'єктами.

Подальші роботи будуть спрямовані на вдосконалення прототипу програмного забезпечення інформаційної системи та розроблення інтелектуальної складової для ефективного пошуку зображень.

1. Shapiro L. Computer vision / L. Shapiro, G. Stockman. – Видавництво Вашингтонського університету, 2006. – 752 с. 2. Білощицький А. О. Ефективність методів пошуку збігів у текстах / А. О. Білощицький, О. В. Діхтяренко // Управління розвитком складних систем. – 2013. – № 14. – С. 144–147. 3. Шозда Н. С. Поиск изображений по текстурным признакам в больших базах данных / Наукові праці Донецького державного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка". – Донецьк, 2002. – № 39. – С. 182–187. 4. Білощицький А. О. Оптимізація системи пошуку збігів за допомогою використання алгоритмів локально чутливого хешування наборів текстових даних / А. О. Білощицький, О. В. Діхтяренко // Управління розвитком складних систем. – 2014. – № 19.– С. 113–117. 5. Alcantarilla P. F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces / P. F. Alcantarilla, J. Nuevo, A. Bartoli // British Machine Vision Conference (BMVC). – 2013. 6. Grewenig S. Cyclic Schemes for PDEBased Image Analysis / S. Grewenig, J. Weickert, C. Schroers, A. Bruhn // International Journal of Computer Vision, 2013. 7. ORB: an efficient alternative to SIFT or SURF, Computer Vision / [E. Rublee, V. Rabaud, K. Konolige, G. Bradski]; (ICCV), IEEE International Conference. – 2011. – С. 2564–2571. 8. Rosten E. Machine learning for high-speed corner detection / E. Rosten, T. Drummond // 9th European Conference on Computer Vision (ECCV). – 2006. – С. 430–443. 9. Yang X. LDB: An ultra-fast feature for scalable augmented reality / X. Yang, K. T. Cheng // In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR). – 2012. – С. 49–57. 10. BRIEF: Binary Robust Independent Elementary Features / [M. Calonder, V. Lepetit, C. Strecha, P. Fua] // 11th European Conference on Computer Vision (ECCV). – 2010. – С. 778–792. 11. Leutenegger S. BRISK: Binary Robust Invariant Scalable Keypoints / S. Leutenegger, M. Chli, R. Siegwart. – Цюріх, 2011. – С. 2548–2555. 12. Литвин В. В. Проектування інформаційних систем: навч. посіб. / В. В. Литвин, Н. Б. Шаховська. – Львів: Новий світ – 2000, 2015. – 380 с. 13. Эккель Б. Философия Java: Библиотека программиста. – СПб.: Питер, 2001. – 980 с. 14. Машнин Т. JavaFX 2.0. Разработка RIA приложений. – БХВ-Петербург, 2012. – 320 с. 15. Дэвис М. Изучаем PHP и MySQL / М. Дэвис, Дж. Филлипс. – М.: Символ-Плюс, 2008. – 442 с. 16. Бернард В. Х. JDBC: Java и базы данных. – М.: Лори, 1999. – 324 с. 17. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2007. – 366 с. 18. Шилдт Г. Искусство программирования на JAVA / Г. Шилдт, Д. Холмс. – М.: Изд. дом "Вильямс", 2005. – 336 с.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua