

УДК 004.942(045)

**АЛГОРИТМ РОЗПІЗНАВАННЯ ФРАГМЕНТІВ ЗОРЯНОГО НЕБА****В. М. Опанасенко**, д-р техн. наук, проф.; **А. А. Каспаревич**, **К. І. Шевчук**

Національний авіаційний університет

vlopanas@ukr.net

*Запропоновано алгоритм розпізнавання фрагментів зоряного неба повзучим вікном (одиничним та широкоформатним) для визначення частини космічного простору, зображеного на знімку.*

**Ключові слова:** цифрова обробка зображення, алгоритм, розпізнавання фрагмента зоряного неба.

*Algorithm of digital image processing of the sky by sliding window of small and big size is proposed. This processing is used for pattern recognition of the sky.*

**Keywords:** digital image processing, algorithm, pattern recognition of the sky.

**Вступ**

Зоряні датчики (ЗД) усе більше застосовуються для високоточного визначення орієнтації космічних апаратів (КА). Сучасні системи орієнтації та навігації з використанням ЗД дають змогу здійснювати тривісну орієнтацію КА на основі алгоритмів розпізнавання сегментів зоряного неба [1].

Зоряні датчики використовують як фотоприймач ПЗЗ-матрицю та визначають орієнтацію шляхом порівняння зоряних величин зірок, які зображені на сегменті зоряного неба із зоряним каталогом, який зберігається в пам'яті бортового комп'ютера [2].

Зоряні каталоги містять таку інформацію, як: назва зірки; галактична довгота зірки; галактична широта зірки; видима зоряна величина; абсолютна зоряна величина; відстань у світлових роках [3; 4]. Кожний зоряний каталог описує зірки з певним діапазоном яскравості.

**Постановка проблеми**

Важливе місце у створенні переспективних систем орієнтації в космічному просторі відводиться розробці швидкодіючих алгоритмів визначення орієнтації КА з використанням методів ідентифікації сегментів зоряного неба.

Для розпізнавання фрагментів зоряного неба розміром  $m \times n$  пікселів необхідно розробити алгоритм, який дозволить виявляти зорі та отримувати їхнє значення яскравості для визначення зоряної величини.

Розпізнавання виконується двома видами вікон: розміром  $2 \times 2$  та  $20 \times 20$  пікселів.

Сканування фрагмента відбувається вікном розміром  $2 \times 2$  пікселів, яке виявляє зірки, та вікном розміром  $20 \times 20$  пікселів, яке виявляє зіркооб'єкти. Після того як буде знайдено зіркооб'єкт, можна буде визначити її яскравість, а разом з тим і зоряну величину.

Роботу алгоритму промодельовати за допомогою прикладної програми.

**Математична модель**

Аналіз зображення виконується повзучими вікнами: одиничним вікном (розміром  $2 \times 2$  пікселів), розміри якого повністю дозволяють проглянути знімок, та широкоформатним (розміром  $20 \times 20$  пікселів), під час сканування яким можуть траплятися випадки, коли його розміри не дозволяють повністю пройти зображення [5]. Це пов'язано з тим, що розмір фрагмента не завжди кратний 20. У цьому випадку необхідно використовувати вікна для крайового аналізу.

Для зображення розміром  $m \times n$  пікселів розмір крайових вікон буде таким:

- для правого краю зображення —  $X \times 20$  пікселів, де  $X$  — ширина крайового вікна;  $X = m \bmod 20$  пікселів;
- для нижнього краю зображення —  $20 \times Y$  пікселів, де  $Y$  — висота крайового вікна;  $Y = n \bmod 20$  пікселів.

Сітку розмітки зображення широкоформатними вікнами подано на рис. 1.

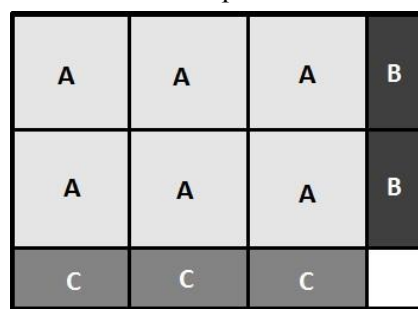


Рис. 1. Сітка сканування зображення розміром  $m \times n$  пікселів широкоформатними вікнами:

- A — вікно розміром  $20 \times 20$  пікселів;
- B — вікно розміром  $X \times 20$  пікселів;
- C — вікно розміром  $20 \times Y$  пікселів

Щоб виявити зірки, які належать до вмонтованого навігаційного каталогу ЗД, необхідно задати поріг яскравості зірок. Після цього відбувається сканування зображення одиничним вікном, під час якого виконується порівняння яскравості пікселів, які перебувають у полі зору вікна, із

заданим порогом яскравості. Координати тих вікон, в яких буде більше двох пікселів із заданого діапазону (три або чотири), записуються до пам'яті для вказівки на зірку.

Порівняння яскравості пікселя із пороговим значенням виконується за допомогою граничних структур, які можуть налаштовуватися на виконання будь-якої операції порівняння.

Формування зірок-об'єктів відбувається скануванням за допомогою широкоформатного вікна (як повнорозмірним вікном, так і крайовими вікнами). Якщо кількість зірок (сформованих одиничним вікном), які потрапили до широкоформатного вікна, буде більше чотирнадцяти, таке вікно слугуватиме вказівником на наявність шуканої зірки. Координати такого вікна дозволять сформувати зірку-об'єкт та визначити її яскравість, яка зберігається в ПЗЗ-матриці ЗД.

Знайшовши яскравість зірки, можна знайти її зоряну величину. Зоряна величина — це безрозмірна числова характеристика яскравості об'єкта, яка характеризує кількість квантів світла, які дійшли від світила до фотоприймача [6]. Вона визначається за формулою [7]:

$$m = -2,512 \lg I + C,$$

де  $I$  — яскравість зірки;  $C$  — константа, яка залежить від вибору базової точки шкали; 2,512 — коефіцієнт Погсона.

### Опис алгоритму

Більшість алгоритмів розпізнавання сегмента зоряного неба використовують прямі методи пошуку [9–16], в яких порівнюються зоряні величини із зоряним каталогом. Переважна більшість цих алгоритмів не мають достатньої швидкодії, що ускладнює їх практичне використання. Ідентифікація сегментів зоряного неба потребує високої швидкодії програмних процедур та економного використання оперативної пам'яті процесора. Для створення ефективного алгоритму необхідно використовувати найбільш інформативні ознаки. Як такі ознаки будуть виступати координати широкоформатного вікна, відносно яких можна визначити зірку та її атрибути.

Алгоритм організовано таким чином: спочатку зображення завантажується в оперативну пам'ять, потім отримуємо масив пікселів і скануємо його одиничним та широкоформатними вікнами. Масив пікселів зображення розміром  $m \times n$  пікселів (рис. 2) формується в такий спосіб:

$$Picture[] = \{p_0, p_1, p_2, \dots, p_{m-1}, p_m, p_{m+1}, \dots, p_{2m-1}, p_{2m}, p_{2m+1}, \dots, p_{3m-1}, p_{3m}, p_{3m+1}, \dots, p_{m(n-1)}, p_{m(n-1)+1}, \dots, p_{mn-1}\},$$

де  $Picture[]$  — масив пікселів зображення;  $p_i$  —  $i$ -й піксель зображення;  $i = 0$  —  $(mn - 1)$ .

$P_0$	$P_1$	$P_2$	...	...	...	$P_{m-1}$
$P_m$	$P_{m+1}$	...	...	...	...	$P_{2m-1}$
...	...	...	...	...	...	...
...	...	...	...	...	...	...
$P_{m(n-1)}$	$P_{m(n-1)+1}$	...	...	...	...	$P_{mn-1}$

Рис. 2. Піксельний набір зображення

Подальша робота алгоритму полягає у формуванні набору зірок за допомогою одиничного вікна, в кутах якого міститься піксель з масиву *Picture*. Індеси пікселів можна визначити у такий спосіб: припустимо, що маємо деяку змінну *index*, яка визначає індекс пікселя для аналізу одиничним вікном. Тоді індеси пікселів для кутів одиничного вікна будуть такими:

- для лівого верхнього кута — *index*;
- для правого верхнього кута — *index+1*;
- для лівого нижнього кута — *index+m*;
- для правого нижнього кута — *index+m+1*.

Сканування зображення вікном відбувається по горизонталі, тому під час ітеративного перебору масиву, для переміщення вікна на нову позицію необхідно значення змінної *index* збільшити на одиницю.

Перехід вікна на новий рядок зображення виконується тоді, коли координата значення правого кута по осі абсцис буде дорівнювати  $m$ . У такому разі необхідно виконати додатковий інкремент змінної *index*, щоб уникнути хибного аналізу неіснуючого квартету пікселів, таких як наприклад  $\{p_{m-1}, p_m, p_{2m-1}, p_{2m}\}$  (рис. 3).

$P_{m-1}$	$P_m$
$P_{2m-1}$	$P_{2m}$

Рис. 3. Неіснуючі пікселі

Фактично ці пікселі існують у масиві, але вони не розташовані на одній горизонтальній прямій зображення. Якщо було б виконано аналіз такого вікна, то було б отримано хибні дані.

Набір пікселів, які задовольняють умову формування зірки, зберігаються як початкові координати одиничного вікна зі статусом зірки.

Сформувавши набір координат таких вікон, переходимо до аналізу зображення широкоформатним вікном. Цей складається з трьох етапів:

I. Аналіз зображення вікном розміром  $20 \times 20$  пікселів;

II. Аналіз зображення вікном розміром  $X \times 20$  пікселів;

III. Аналіз зображення вікном розміром  $20 \times Y$  пікселів.

Аналіз широкоформатним вікном подібний до аналізу одиничним вікном. Відмінність лише в тому, що в цьому випадку будемо аналізувати не пікселі, а одиничні вікна.

Якщо кількість одиничних вікон, які мають статус зірки більше 14, координати розташування такого вікна записуються до пам'яті комп'ютера як вказівник на місце розташування зірки-об'єкта.

Аналіз широкоформатним вікном відбувається у такий спосіб (розглянемо випадок вікна розміром  $20 \times 20$  пікселів):

- визначаємо позицію вікна відносно зображення. Ця позиція буде задавати позицію, з якої відбувається аналіз тих одиничних вікон, які потрапили до широкоформатного вікна:

$$offset = j + i * (m - 1),$$

де  $offset$  — позиціонування вікна;  
 $j = 0 \div m - (m \bmod 20)$  — переміщення по горизонталі зображення;  
 $i = 0 \div n - (n \bmod 20)$  — переміщення по вертикалі зображення;  
 $m$  та  $n$  — ширина та висота зображення відповідно;

- знаходимо всі зірки, які потрапили у вікно з позицією  $offset$  та перевіряємо їх статус.

Індекс, який буде ідентифікувати зірку, визначається як

$$position = k + s * (m - 1) + offset,$$

де  $position$  — індекс одиничного вікна з масиву зірок,  $k = s = 0 \div 19$ .

Пошук зірки-об'єкта виконується таким чином: шукаємо перше широкоформатне вікно. Потім ітеративно виконуємо пошук найбільш віддаленого вікна по вертикалі вниз, шлях до якого можна прокласти через вікна, які слідує одне за одним нерозривно (рис. 4).



Рис. 4. Пошук найвіддаленішого вікна:

A — вікно, від якого виконується пошук;  
 B — найвіддаленіше вікно від вікна A

Як правило, на фрагменті зірки мають форму кола, тому необхідно побудувати квадрат, у який буде вписано коло. Розмір квадрата буде рівний кількості вікон від вікна A до вікна B (у нашому

випадку чотири). Тепер необхідно знайти всі існуючі широкоформатні вікна в околі вікна B, а саме, віддаляємося на чотири вікна праворуч та ліворуч від вікна B, щоб визначити наявність широкоформатних вікон нижньої грані квадрата (рис. 5).

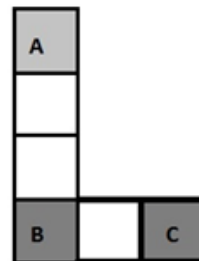


Рис. 5. Існуючі широкоформатні вікна нижньої грані в околі вікна B

З рис. 5 видно, що було знайдено три вікна нижньої грані, але квадрат має складатися з чотирьох широкоформатних вікон, тому для того, щоб не пропустити можливі наявні вікна в середині квадрату, додаємо по одному уявному вікну з кожного краю нижньої грані (рис. 6).

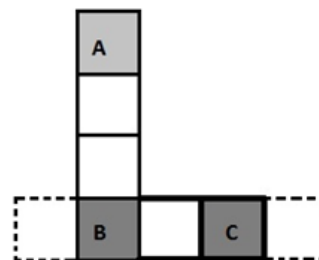


Рис. 6. Уявні вікна нижньої грані

Тепер лишилося пройти шлях від нижньої грані до верхньої грані та знайти всі існуючі широкоформатні вікна. В результаті аналізу отримуємо набір споріднених вікон, які будуть описувати зірку-об'єкт (рис. 7). Узагальнену схему алгоритму подано на рис. 8.

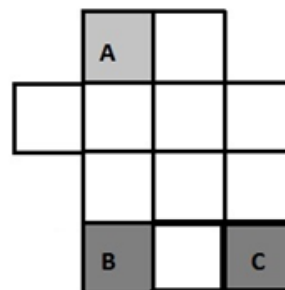


Рис. 7. Знайдені споріднені вікна, які описують об'єкт-зірку

Для моделювання роботи алгоритму було створено програму.

Ця програма дає змогу формувати звіти про аналіз зображення в табличному вигляді та надавати обробці засобами MS Excel.

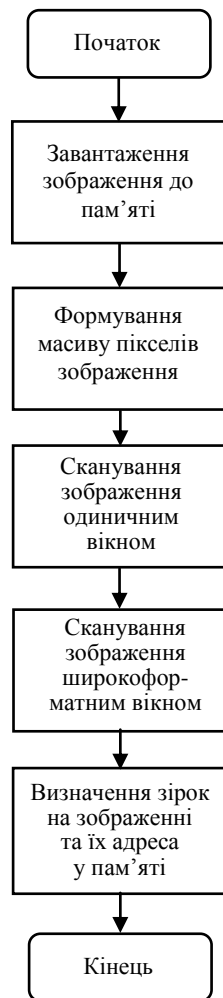


Рис. 8. Узагальнена схема алгоритму

Під час розробки програмної моделі алгоритму було враховано спосіб доступу до значення яскравості, яка зберігається в ПЗЗ-матриці зоряного датчика. Для того щоб відтворити умови доступу до ПЗЗ-матриці, в код програми було включено блок доступу до фізичної адреси.

#### Фізична адресація в Java

Відомо, що під час розробки мови Java головний акцент робився на «безпеку» коду (так званий «safe code») [8]. Це означало відмову від вказівників, роботи з пам'яттю і подібних низькорівневих засобів. Зовсім відмовитися не вдалося, оскільки треба було лишити таку можливість для власних класів.

Клас *sun.misc.Unsafe* входить до складу *Sun Java Runtime* і розміщується в пакеті *sun.\**. Тут наявні функції для роботи з пам'яттю (*allocateMemory*, *freeMemory* тощо), читання і запису за заданою адресою (*putLong*, *getLong* тощо) та деякі більш спеціалізовані (*throwException*, *monitorEnter* тощо) методи.

Створити об'єкт класу *Unsafe* неможливо, оскільки єдиний конструктор — приватний, а в *getUnsafe()* відбувається перевірка завантажувача

вище вказаного класу і об'єкт повертається якщо клас завантажено з використанням *Bootloader*. Інакше отримаємо помилку *SecurityException*. Щоб цього уникнути, використаємо механізм рефлексії. Наведемо приклад, в якому об'єкт класу *Unsafe* викликає два методи — *objectFieldOffset()* та *staticFieldOffset()*. Як результат отримаємо адресу розміщення об'єкта класу.

#### Приклад реалізації алгоритму

Для перевірки роботи алгоритму візьмемо зображення розміром 165×78 пікселів (рис. 9).

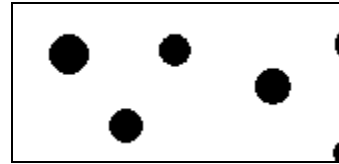


Рис. 9. Зображення для аналізу

Відповідно до вище згаданого алгоритму розпізнавання фрагмент буде скануватися двома типами вікон — одиничним та широкоформатним. У результаті буде отримано перелік координат широкоформатних вікон відносно зображення та відповідні фізичні адреси цих вікон.

Розміри зображення не дозволяють повністю просканувати зображення за допомогою широкоформатного вікна. Тому необхідно використовувати широкоформатні вікна для крайового аналізу. Розміри цих вікон визначаються автоматично в програмі.

У результаті роботи алгоритму отримаємо дані, наведені в табл. 1, 2 та 3.

Таблиця 1

Region [20×20]		
X	Y	Address
1	2	3
20	0	40
60	0	80
80	0	100
0	20	3300
20	20	3320
1	2	3
60	20	3360
80	20	3380
120	20	3420
40	40	6620
1	2	3
60	40	6640
120	40	6700

Таблиця 2

Region [5×20]		
X	Y	Address
1	2	3
160	0	180

Таблиця 3

Region [20×18]		
X	Y	Address
1	2	3
40	60	9900
60	60	9920

Як видно з таблиць, аналіз було проведено трьома широкоформатними вікнами: розміром 20×20 пікселів, 5×20 пікселів та 20×18 пікселів.

У стовпчиках X та Y вказано координати відносно зображення, а в стовпчику Address — адреса комірки пам'яті пристрою.

### Висновки

Запропоновано алгоритм для розпізнавання фрагментів зоряного неба, який виявляє наявні зірки та отримує їх значення яскравості.

Оскільки навігаційний каталог (зоряний каталог) зоряного датчика містить лише опис зірок з певного діапазону зоряних величин, доцільно виконувати граничне порівняння знайдених зірок за допомогою граничних структур. Ці структури можуть налаштовуватися на виконання будь-якої операції порівняння на базі логічних елементів та заданого порогового значення.

Розпізнавання та формування зірок, придатних для аналізу, відбувається за допомогою повзучих вікон різного формату: одиничним вікном (розміром 2×2 пікселів), широкоформатним (розміром 20×20 пікселів) та крайовими вікнами (розміром X×20 та 20×Y пікселів для фрагмента розміром m×n пікселів) для випадків, коли розмір зображення не кратний 20.

Для моделювання роботи алгоритму було створено програму, код якої написано мовою Java 2 SE 5.0 з використанням IDE Eclipse 3.2.

У коді міститься реалізація доступу до фізичної адреси, за допомогою класу *sun.misc.Unsafe*.

Звіт про аналіз зображення формується в табличному вигляді. Особливістю програми є її інтегрованість з MS Excel.

Алгоритм розпізнавання фрагментів зоряного неба підлягає реалізації на ПЛІС, що дозволяє перепрограмувати логіку роботи алгоритму та використовувати у складі процесора бортового комп'ютера космічних апаратів.

### ЛІТЕРАТУРА

1. Алгоритм определения ориентации космического аппарата с помощью звездного датчика / В. Е. Набивач, В. Н. Опанасенко, О. В. Семенив,

В. А. Яценко // Космична наука і техніка, 2011. — Т. 17. — № 1. — С. 84–89.

2. Дятлов С. А. Обзор звездных датчиков ориентации космических аппаратов / С. А. Дятлов, Р. В. Бессонов // Современные проблемы определения ориентации и навигации космических аппаратов. — Таруса, 2008. — С. 11–31.

3. Available at. — [Електронний ресурс]. — Режим доступу: <http://www.atlasoftheuniverse.com/stars.html>

4. Available at. — [Електронний ресурс]. — Режим доступу:

[https://wakata.nascom.nasa.gov/dist/generalProducts/attitude/ATT\\_SKYMAP.html](https://wakata.nascom.nasa.gov/dist/generalProducts/attitude/ATT_SKYMAP.html)

5. Бернд Яне. Цифровая обработка изображений / Яне Бернд; пер. с англ. А. М. Измайловой. — М. : Техносфера, 2007. — 584 с.

6. Available at. — [Електронний ресурс]. Режим доступу: <http://www.astronet.ru/db/msg/1174337>

7. Зоряна величина абсолютна // Астрономічний енциклопедичний словник: за заг. ред. І. А. Климишина та А. О. Корсунь. — Л. : ЛНУ-ГАО НАНУ, 2003. — С. 185. — ISBN 966-613-263-X, УДК 52(031).

8. Хорстманн К.С. Java 2. Библиотека профессионала / К. С. Хорстманн, Г. Корнелл. — Том 1. Основы, 8-е изд.: пер. с англ. — М. : ООО «И. Д. Вильямс», 2008. — 816 с.

9. Fossum E. R. Application of the active pixel sensor concept to guidance and navigation/ E. R. Fossum, R. K. Bartman, A. R. Eisenman // Proc. SPIE. — 1993. — P. 256–265.

10. Ceometric voting algorithm for star trackers / M. Kolomenckin, S. Pollak, I. Shimshoni, M. Lindbaum // IEEE Trans. Aerospace and Electron. Syst. — 2008. — P. 441–456.

11. Lee H. Star pattern identification technique by modified grid algorithm / H. Lee, H. Bang // IEEE Trans. Aerospace and Electron. Syst. — 2007. — P. 202–213.

12. Lin T. All-sky automated quaternary star pattern recognition / T. Lin, J. L. Zhou // J. Astronaut. — 2000. — P. 82–85.

13. Mortari D. Search-less algorithm for star pattern recognition / D. Mortari // J. Astronaut. Sci. — 1997 — P. 179 — 194.

14. Mortari D. The pyramid star identification technique / D. Mortari, M. A. Samaan, C. Bruccoleri // Navigation. — 2004. — P. 171–183.

15. Myers J. R. SKY2000 Catalog, Version 4 (Myers+2002) / J. R. Myers, C. B. Sande, A. C. Miller [et al.] // SAO/NASA ADS Astronomy. <http://vizier.cfa.harvard.edu/viz-bin/VizieR?-source=V/109>.

16. Rousseau G. L. A. Star recognition algorithm for APS star tracker: Oriented triangles / G. L. A. Rousseau, J. Bostel, B. Mazari // IEEE Aerospace and Electron. Syst. Mag. — 2005. — P. 27–31.

Стаття надійшла до редакції 07.03.12.