

Д. т. н. В. М. БОРОВИЦЬКИЙ, В. А. АНТОНЕНКО

Україна, Національний технічний університет України
«Київський Політехнічний Інститут ім. Ігоря Сікорського»
E-mail: vborovytsky@yahoo.com

РОЗРАХУНОК ДИСКРЕТНОЇ КОРЕЛЯЦІЙНОЇ ФУНКЦІЇ У ФАСЕТНИХ СИСТЕМАХ ТЕХНІЧНОГО ЗОРУ

Запропоновано фасетну систему технічного зору, яка складається з однакових фасетних елементів. Кожний фасетний елемент містить оптичну систему, декілька фотоприймачів, попередній підсилювач та універсальний мікроконтролер. У такій системі всі фасетні елементи працюють паралельно та незалежно один від одного. Кожний фасетний елемент здійснює швидке вимірювання кутової швидкості об'єктів у своєму полі зору шляхом обчислення дискретних кореляційних функцій сигналів фотоприймачів. Досліджено можливість застосування у фасетних елементах недорогих універсальних мікроконтролерів.

Ключові слова: системи технічного зору, цифрова обробка сигналів, кутова швидкість, дискретна кореляційна функція, цифрові камери, безпілотні літальні апарати.

Зараз автономні транспортні засоби (АТЗ), у тому числі безпілотні літальні апарати (БПЛА) та самокеровані автомобілі, стають важливою частиною інфраструктури доставлення товарів і продукції [1, 2]. Щоб уникнути зіткнень з рухомими та нерухомими об'єктами, які їх оточують, їм потрібні швидкі системи технічного зору з широким полем зору. Такі системи мають забезпечувати блок керування АТЗ надійними даними про значення та напрямок швидкості оточуючих об'єктів. Зараз системи технічного зору застосовують цифрові або нейроморфні камери, зокрема камери, які використовують обробку подій — event-based cameras — саме вони вважаються одним з найкращих варіантів для АТЗ [3, 4]. Основними елементами цих камер є:

- оптична система, що формує зображення віддаленого об'єкта в площині його зображення;
- матричний фотоприймальний пристрій у вигляді матриці фоточутливих елементів і зчитувальної електроніки, розробленої спеціально для виявлення руху;
- цифровий сигнальний процесор для швидкої локалізації рухомих об'єктів;
- механічні, електричні та електронні компоненти для стабілізації та кутових переміщень камери.

Проте камери, що використовують обробку подій, мають низку недоліків. Наприклад, сучасні матричні фотоприймальні пристрої мають багатоканальні електронні блоки з послідовним зчитуванням сигналу. Кількість каналів зчитування та кількість ядер у процесорах для обробки сигналів дуже малі в порівнянні з кількістю фоточутливих елементів [5], що знижує продуктивність обробки сигналів. Також важко спроектувати систему бачення з широким або дуже

широким полем зору. Виконання таких камер з кутом огляду більше ніж $\pm 90^\circ$ є проблематичним, при цьому для АТЗ бажано, щоб датчик зображення міг «бачити» у полі зору близькому $\pm 180^\circ$ для виключення ймовірності зіткнення по всіх напрямках. Крім того, камери з матричними фотоприймальними пристроями потребують механічних компонентів для обертання в різних напрямках і стабілізації зображення, що робить їх достатньо дорогими, а також збільшує їх вагу та розміри.

Перераховані недоліки властиві й звичайним цифровим камерам, підключеним до цифрових сигнальних процесорів. Одним з найкращих альтернатив цифровим камерам є фасетні системи технічного зору (ФСТЗ), побудовані по аналогії з фасетними органами зору живих істот [6, 7]. Такі системи містять певну кількість однакових фасетних елементів і мають такі переваги:

- швидка обробка сигналів (велика кількість фасетних елементів здійснює паралельне зчитування та обробку сигналу; кожний фасетний елемент обробляє сигнали від своїх світлочутливих елементів, в результаті система зору може виявляти рухомі об'єкти за дуже короткий час);
- широке поле зору (фасетні елементи можуть покривати велику вигнуту поверхню, щоб забезпечити поле зору більше за півсферу) [7];
- достатня просторова роздільна здатність (такі системи мають майже постійну просторову роздільну здатність по всьому полю зору без будь-якого механічного сканування) [6];
- фасетні елементи можуть містити фотоприймачі, активовані поляризацією або певними спектральними смугами оптичного випромінювання, це

дає змогу фіксувати сигнали, які характеризують розподіл поляризації або оптичного спектру по всьому полю зору;

— висока надійність системи завдяки незалежній паралельній роботі численних фасетних елементів (система може працювати, навіть якщо деякі фасетки пошкоджені).

Ці переваги приваблюють дослідників та інженерів, які намагаються розробити ФСТЗ для керування автономними транспортними засобами [8—11]. ФСТЗ мають хороші перспективи застосування для навігації, безпечного наведення, відстеження об'єктів, усунення зіткнень з рухомими та нерухомими об'єктами шляхом вимірювання кутової швидкості переміщення об'єктів у полі зору кожного фасетного елемента [8—10]. Це дозволяє виявляти рухомі та нерухомі об'єкти та уникати зіткнень з ними [11]. Як показали дослідження, навіть один фасетний елемент, який вимірює кутову швидкість у малому полі зору, може забезпечити рух БПЛА з відслідковуванням профілю поверхні, а декілька фасетних елементів — рух з обминанням перешкод [11, 12].

Обробка сигналів у фасетних елементах може здійснюватися аналоговими або цифровими електричними схемами, зокрема мікроконтролерами (МК) та схемами програмованої логіки [12—14]. Відомо, що кутову швидкість можна вимірювати за допомогою обчислення дискретної кореляційної функції (ДКФ) сигналів, які зчитуються з різних фотоприймачів, з наступним розрахунком положення та значення максимуму цієї функції [14, 15]. Математично доведено, що кореляційно-екстремальні системи є оптимальними, бо саме вони забезпечують максимізацію співвідношення сигнал/шум системи у вихідному сигналі, а значить, і мінімальні похибки вимірювання [15].

Розрахунок ДКФ вимагає виконання великого обсягу обчислень, і тому виникає протиріччя: фасетні елементи у ФСТЗ мають бути компактними та дешевими, але це обмежує застосування спеціалізованих аналогових або цифрових інтегральних схем, а також потужних процесорів, зокрема багатоядерних, які мають високу вартість або недоступні для широкого застосування. З іншої сторони, для найкращого визначення кутової швидкості треба обчислювати ДКФ, що вимагає здійснення великого обсягу обчислень у кожному фасетному елементі за дуже короткий час — від 1 до 10 мс. У більшості ФСТЗ це протиріччя вирішується відмовою від обчислення ДКФ та застосуванням менш точних методів багатоканальної аналогової або цифрової фільтрації сигналів [8—14].

В цьому дослідженні автори поставили за мету знайти метод обчислення ДКФ за допомогою дешевого універсального мікроконтролера, який дозволить обчислювати не менше сотні значень цієї функції за 1—10 мс і тим самим зробить можливим створення вітчизняних недорогих та швидких фасетних елементів на елементній базі, яка доступна на ринку електронних компонентів.

Фасетна система технічного зору

Запропонована ФСТЗ (рис. 1) складається з однакових фасетних елементів 1, кожний з яких містить [16, 17]:

- однокомпонентну оптичну систему 2 (формує зображення у площині, де розташовуються фоточувливі елементи фотоприймачів 3);
- фотоприймачі 3 (фотодіоди або фототранзистори) у кількості від трьох до восьми;
- багатоканальний попередній програмований підсилювач 4 (перетворює струм фотоприймачів у

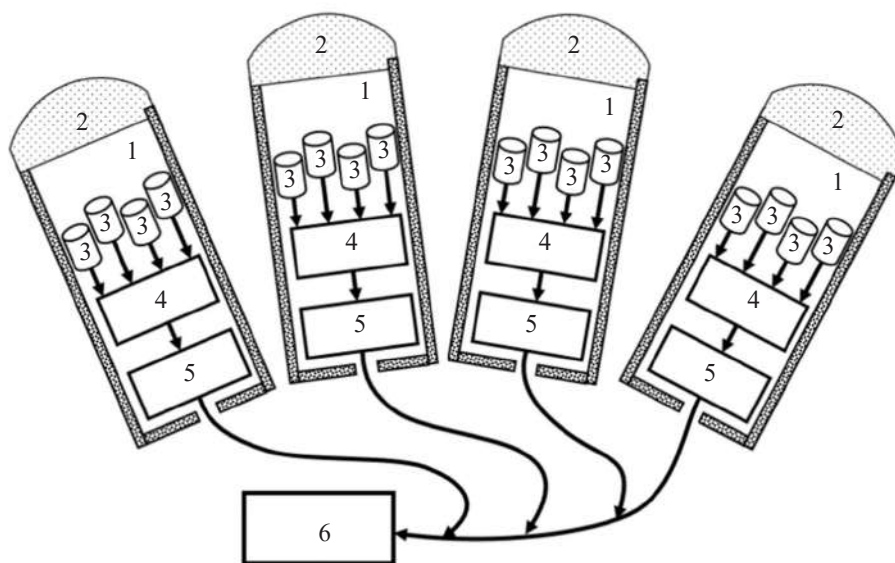


Рис. 1. Запропонована фасетна система технічного зору

напрягу та забезпечує потрібний рівень цієї напруги в вихідному сигналі);

— мікроконтролер 5 з аналого-цифровим перетворювачем (АЦП).

Після підсилення сигнал з фотоприймачів подається на вхід АЦП, де він переводиться у цифрову форму та у вигляді масиву цілих чисел записується в оперативну пам'ять мікроконтролера. Програма, що зберігається у постійній пам'яті МК, забезпечує зчитування та перетворення у цифрову форму сигналів усіх фотоприймачів шляхом подачі сигналів управління на програмований підсилювач та АЦП. Головним призначенням цієї програми є розрахунок кутової швидкості переміщення зображення шляхом обчислення положення максимального значення кореляційної функції зареєстрованих цифрових сигналів. Ці дані допомагають автономним транспортним засобам орієнтуватися в просторі під час руху [8, 9, 11, 12] — максимальне значення цієї функції та його положення у масиві показує відстань, на яку змістився сигнал на одному фотоприймачі відносно сусіднього, та напрям його переміщення. Якщо порахувати часову затримку максимального значення, то при відомих значеннях фокусної відстані оптичної системи 2 та відстані між фоточутливими елементами фотоприймачів 3 можна визначити кутову швидкість, з якою рухається об'єкт у полі зору фасетного елемента 1. Розраховані дані МК передає у центральний комп'ютер 6, де розраховується траєкторія руху з обминанням перешкод. Застосування масиву таких фасетних елементів дозволяє побудувати біологічно подібну ФСТЗ з широким полем зору та паралельною обробку сигналів. При цьому умовою створення придатних для практичного застосування фасетних елементів є швидкий розрахунок кореляційних функцій при застосуванні недорогих універсальних мікроконтролерів.

Розрахунок дискретної кореляційної функції

Сьогодні навіть невеликі БПЛА, які є у відкритому продажі, можуть розвивати швидкість в межах 50—130 км/год, і для орієнтування у просторі їм необхідно отримувати значення кутової швидкості кожні декілька мілісекунд [18]. Тому задача визначення часу розрахунку ДКФ за допомогою найдешевших МК стає дуже важливою в процесі розробки сучасних ФСТЗ.

Детальний опис методів розрахунку ДКФ та кореляційно-екстремальних систем наведено у [12].

Усі методи обчислення ДКФ можна розділити на дві групи: прямий розрахунок та розрахунок із застосуванням швидкого перетворення Фур'є (ШПФ). Перша група об'єднує методи, які полягають у виконанні операцій зчитування значення k -го елемента першого вхідного масиву та значення $(k+m)$ -го елемента другого вхідного масиву, їх перемноження та

подальшого додавання добутку до результату, що і є значенням ДКФ:

$$r_m = r_m(a, b) = \sum_{k=0}^{N-1} a_k b_{k+m}, \quad m = 0, 1, \dots, M-1, \quad (1)$$

де r_m — ДКФ вхідних масивів даних a та b ;

N — кількість елементів вхідних масивів, які використовуються для обчислення кореляційної функції;

M — кількість значень ДКФ;

k, m — індекси масивів вхідних даних та масиву ДКФ відповідно.

Далі проводиться пошук максимального значення ДКФ та його індексу, який використовується для визначення кутової швидкості. Звертаємо увагу на те, що для визначення напрямку кутової швидкості треба два рази обчислювати ДКФ — $r_m(a, b)$ та $r_m(b, a)$, тобто спочатку для випадку зсуву сигналу в одному напрямку, а потім у протилежному. Для таких обчислень розмір усіх масивів даних має дорівнювати $N + M$.

Методи з застосуванням ШПФ полягають у виконанні прямого перетворення Фур'є над двома вхідними масивами, перемноження двох отриманих комплексних спектрів та зворотного перетворення Фур'є [19]:

$$\begin{aligned} c_m &= c_m(a, b) = \text{IFFT} \left[\text{FFT}(a) \cdot \text{FFT}(b)^* \right] = \\ &= \text{IFFT}(Sa \cdot Sb^*) = \\ &= \frac{1}{N} \cdot \sum_{n=0}^{N-1} (Sa_n \cdot Sb_n^*) \cdot \exp\left(\frac{2\pi}{N} \cdot j \cdot n \cdot m\right), \end{aligned} \quad (2)$$

$m = 0, 1, \dots, N-1,$

$n = 0, 1, \dots, N-1,$

де c_m — циклічна ДКФ масивів даних a та b ;

FFT, IFFT — символи прямого та зворотного швидких перетворень Фур'є;

* — символ комплексно-узгодженого числа;

j, π — константи;

Sa, Sb — комплексні спектри вхідних масивів a та b відповідно:

$$Sa_m = \text{FFT}(a) = \sum_{k=0}^{N-1} a_k \exp\left(-\frac{2\pi}{N} \cdot j \cdot k \cdot m\right);$$

$$Sb_m = \text{FFT}(b) = \sum_{k=0}^{N-1} b_k \exp\left(-\frac{2\pi}{N} \cdot j \cdot k \cdot m\right).$$

Далі робиться пошук максимального значення ДКФ (2) та його індексу. При цьому треба вказати різницю між ДКФ (1) та (2): перший випадок — пряме обчислення, другий — обчислення циклічної ДКФ. Тобто при застосуванні ШПФ вхідні масиви розглядаються як фрагменти великої періодичної структури даних з періодом N , а ДКФ (2) є циклічною та дозволяє оцінити схожість сигналів фотоприймачів при їх зсувах в обох напрямках.

Час обчислення дискретної кореляційної функції мікроконтролером

Спробуємо оцінити час визначення максимального значення дискретної кореляційної функції, його положення та необхідний обсяг оперативної пам'яті при застосуванні двох найрозповсюдженіших універсальних та недорогих МК — ATmega 1284P та STM32F103C8T6 (табл. 1).

Спочатку зробимо порівняння двох методів. Як видно з табл. 2, пряме обчислення потребує виконання більшої кількості операцій, але меншого обсягу оперативної пам'яті. Застосування ШПФ — навпаки: потребує виконання формально меншої кількості операцій, але значно більшого обсягу оперативної пам'яті.

У фасетного елемента є обмеження за часом розрахунку — від 0,001 до 0,01 с, а МК має обмежений обсяг пам'яті. Тому виникає завдання знаходження оптимального методу розрахунку ДКФ, який би задовольняв вимогам як за часом обчислення, так і за обсягом оперативної пам'яті МК.

З метою знаходження такого методу виконаємо аналіз фрагментів підпрограм розрахунку ДКФ, представлених у додатках до статті, які записані на мові асемблер для МК ATmega 1284P та STM32F103C8T6 [22, 23]. Слід зазначити, що сучасні мікроконтролери можуть мати пристрої для швидкої цифрової обробки сигналів (DSP) та пристрої векторних обчислень, але це суттєво збільшує їх вартість. Недорогі

Таблиця 1

Технічні характеристики економічних універсальних мікроконтролерів [20, 21]

| Характеристика | Мікроконтролер | |
|----------------------------------|--------------------------------|-----------------------------------|
| | ATmega 1284P | STM32F103C8T6 |
| Компанія виробник | Microchip www.microchip.com | ST Microelectronics www.st.com |
| Архітектура | AVR, 8-біт | ARM, 32-біт |
| Максимальна робоча частота, МГц | 20 | 72 |
| Обсяг постійної пам'яті, Кбайт | 128 | 64 |
| Обсяг оперативної пам'яті, Кбайт | 16 | 20 |
| Розрядність АЦП, біт | 10 | 12 |
| Орієнтовна вартість, долл. США | 5 — 7 | |

Таблиця 2

Порівняння методів обчислення ДКФ

| Пряме обчислення | Обчислення з використанням ШПФ |
|--|---|
| <ul style="list-style-type: none"> не потребує великого обсягу оперативної пам'яті для зберігання результату (для зберігання значення та індексу максимуму кореляційної функції достатньо декілька байтів, усі значення кореляційної функції зберігати необов'язково); розмір масивів вхідних даних може приймати будь-які значення більші ніж 2; використовуються цілі числа без знаку; підпрограма знаходження максимального значення ДКФ є простою; кількість обчислень є великою, особливо при великих розмірах масивів, оскільки вона пропорційна добутку $N \cdot M$ | <ul style="list-style-type: none"> потребує великого обсягу оперативної пам'яті (для зберігання проміжних даних — комплексних спектрів — та результату масиву значень ДКФ); необхідність зберігання масиву з дискретними значеннями функцій \sin та \cos; розмір масивів вхідних даних повинен приймати певні значення — $N = 2^P$, де P — ціле число більше одиниці; використання операцій з цілими числами зі знаком є обов'язковим; програма для МК є достатньо складною й містить підпрограми ШПФ; кількість обчислень мала, що особливо відчутно при великих розмірах масивів, оскільки вона пропорційна добутку $N \cdot \log_2 N$ |

МК ATmega 1284P та STM32F103C8T6 таких пристроїв не мають [20, 21]. Тому наведені підпрограми розрахунку ДКФ використовують алгоритм прямого обчислення за формулою (1) та відомий алгоритм ШПФ зі збереженням дискретних значень функцій \sin та \cos , перемножених на певну константу, у вигляді масиву цілих чисел. У **табл. 3** наведено інформацію щодо формул для розрахунку часу обчислення ДКФ та обсягів оперативної пам'яті для цього.

Наведемо фактори, які були враховані в процесі розроблення підпрограм.

1. Для визначення напрямку та абсолютного значення кутової швидкості треба обчислити дві її ортогональні складові. Якщо фасетний елемент має 4 фотоприймачі, з яких зчитуються сигнали та перетворюються у 4 масиви вхідних значень, то при прямому обчисленні (формула (1)) ДКФ треба обчислювати 4 рази: $r_m(a, b)$ та $r_m(b, a)$ для кожного з двох ортогональних напрямків. При застосуванні ШПФ (формула (2)) ДКФ треба обчислювати лише 2 рази — по одному разу для кожного з двох ортогональних напрямків.

2. У 8-розрядному МК ATmega 1284P з 10-бітним АЦП вхідні масиви краще представити у вигляді масиву байтів, а у 32-розрядному МК STM32F103C8T6 з 12-бітним АЦП — як масиви 2-байтових цілих чисел. Відповідно, значення ДКФ треба буде представляти у вигляді 4-байтового числа у МК ATmega та 4- або 8-байтового числа у МК STM32F103C8T6. При розрахунку 4-байтових значень ДКФ є важливе обмеження: якщо використовуються 11 старших розрядів АЦП, то для уникнення переповнення 4-байтних

регістрів МК розмір масивів вхідних даних $N + M$ не повинен перевищувати $2^{10} = 1024$. Перевагою прямого обчислення ДКФ є те, що для отримання її максимального значення не треба зберігати весь масив ДКФ у оперативній пам'яті МК.

3. При обчисленні ДКФ з застосуванням ШПФ у оперативній пам'яті МК треба зберігати 4 вхідні масиви 2-байтових значень, 4 масиви 4-байтових значень для тимчасового зберігання двох комплексних спектрів сигналів та один 4-байтовий масив для зберігання ДКФ. Також для здійснення ШПФ необхідно, щоб допустимі розміри масивів вхідних даних дорівнювали ступеню числа 2, тобто 16, 32, 64 і т. д. У постійній пам'яті МК треба буде зберігати масив з дискретними значеннями функцій \sin та \cos , а сам МК має забезпечувати множення цілих чисел зі знаком. Оскільки 8-розрядний МК ATmega 1284P не містить відповідних для цього машинних команд, обчислення ДКФ з використанням ШПФ на цьому МК розглядатися не буде.

4. Знаходження максимального значення ДКФ полягає у виконанні численних порівнянь з поточним максимальним значенням i , залежно від результату порівняння, зміні цього значення. Тобто час обчислення ДКФ залежатиме від того, якою є послідовність вхідних даних: якщо значення ДКФ постійно зменшуються, тоді час знаходження її максимального значення буде мінімальним; якщо постійно збільшуються — максимальним. Тому для оцінювання часу обчислення був вибраний найгірший випадок — коли час обчислення максимального значення ДКФ буде максимальним.

Таблиця 3

Кількість операцій та обсяг оперативної пам'яті, необхідні для розрахунку ДКФ, при використанні мікроконтролерів ATmega 1284P та STM32F103C8T6

| Параметр* | Пряме обчислення ДКФ ($N_{CORR} = 4, N_D = 4$) | | | Обчислення ДКФ із застосуванням ШПФ ($N_{CORR} = 2, N_D = 4$) |
|--------------|---|--|--|---|
| | ATmega 1284P | STM32F103C8T6 8-байтові значення ДКФ | STM32F103C8T6 4-байтові значення ДКФ | STM32F103C8T6 4-байтові значення ДКФ |
| N_C | $N_{CORR}(27+27M+11NM)$ | $N_{CORR}(8+12M+5NM)$ | $N_{CORR}(7+7M+5NM)$ | $N_{CORR}(34+21 \cdot \log_2 N + (31+81 \cdot \log_2 N) \cdot N)$ |
| t_{CORR} | $N_{CORR}(97+66M+34NM)/f_T$ | $N_{CORR}(11+12M+5NM)/f_T$ | $N_{CORR}(10+7M+5NM)/f_T$ | $N_{CORR}(45+21 \cdot \log_2 N + (31+81 \cdot \log_2 N) \cdot N)/f_T$ |
| V_D , байт | $N_D(N+M)$ | $N_D \cdot 2(N+M)$ | $N_D \cdot 2(N+M)$ | $(18+2N_D) \cdot N$ |
| $\max(N+M)$ | 4096 | 2560 | 2560 | 512 |

* N_{CORR} — кількість обчислень ДКФ; N_D — кількість масивів даних; N_C — кількість машинних команд, які необхідно виконати; t_{CORR} — час обчислень за тактової частоти f_T ; V_D — обсяг пам'яті для зберігання даних; $\max(N+M)$ — максимальний розмір масивів вхідних даних.

5. Будемо розглядати МК, які не мають можливості підключення оперативної зовнішньої пам'яті, оскільки така опція ускладнює конструкцію фасетного елемента та збільшує його вартість.

6. Очевидно, що 32-розрядний мікроконтролер STM32F103C8T6 з архітектурою ARM має набагато більшу обчислювальну потужність ніж 8-розрядний МК ATmega 1284P. Своєю чергою, застосування МК ATmega 1284P з архітектурою AVR має свої переваги. По-перше, це простота підключення через меншу кількість виводів корпусу мікросхеми та широкий діапазон напруги живлення — від 1,8 до 5,5 В. По-друге — простота розроблення програм за допомогою інтегрованого середовища AVR Studio у порівнянні з програмами для мікроконтролерів з архітектурою ARM, яка вимагає знання інтерфейсу CMSIS та бібліотек функцій МК серії ST32F103 [22, 23]. Тому вибір МК ATmega 1284P для розроблення та до-

слідження макетів фасетних елементів можна вважати обґрунтованим.

7. Крім обчислення ДКФ МК має ще зчитувати дані з АЦП, задавати параметри програмованого підсилувача, щоб реалізувати автоматичне регулювання підсилення сигналів фотоприймачів, пересилати результати розрахунків у центральний комп'ютер АТЗ. При цьому зауважимо, що ці операції будуть займати набагато менше часу роботи процесора МК, ніж обчислення ДКФ.

У табл. 4 та на рис. 2 наведено інформацію щодо часу обчислення максимального значення ДКФ залежно від розміру масиву вхідних даних. Аналіз отриманих результатів показує наступне:

— недорогі універсальні МК забезпечують час прямого розрахунку максимального значення ДКФ 0,01 с для масивів вхідних даних розміром 512 елементів та 0,001 с для масивів зі 128 елементів. Це

Таблиця 4

Час обчислення максимального значення ДКФ та його положення в масиві ДКФ залежно від розміру масиву вхідних даних $N + M$ при використанні МК ATmega 1284P та STM32F103C8T6

| $N + M$ | Час прямого обчислення ДКФ, мс ($N_{CORR} = 4, N = M$) | | | Час обчислення ДКФ із застосуванням ШПФ, мс ($N_{CORR} = 2, N = M$) |
|---------|---|--|--|---|
| | ATmega 1284P | STM32F103C8T6 8-байтові значення ДКФ | STM32F103C8T6 4-байтові значення ДКФ | STM32F103C8T6 4-байтові значення ДКФ |
| 16 | 0,59 | 0,02 | 0,02 | 0,18 |
| 32 | 2,08 | 0,08 | 0,07 | 0,44 |
| 64 | 7,81 | 0,31 | 0,30 | 1,06 |
| 128 | 30,32 | 1,18 | 1,16 | 2,43 |
| 192 | 67,50 | 2,63 | 2,60 | — |
| 256 | 119,40 | 4,63 | 4,60 | 5,52 |
| 320 | 186,00 | 7,22 | 7,18 | — |
| 384 | 267,30 | 10,37 | 10,31 | — |
| 448 | 363,29 | 14,09 | 14,03 | — |
| 512 | 473,99 | 18,38 | 18,30 | 12,35 |
| 768 | 1063,78 | 41,22 | 41,11 | — |
| 1024 | 1888,77 | 73,16 | 73,01 | 27,34 |
| 1280 | 2948,96 | 114,20 | 114,03 | — |
| 1536 | 4244,35 | 164,35 | 164,14 | — |
| 1792 | 5774,93 | 223,60 | 223,36 | — |
| 2048 | 7540,73 | 291,96 | 291,67 | 59,97 |
| 2560 | 11777,92 | 455,97 | 455,61 | — |
| 3072 | 16955,93 | 656,39 | 655,95 | — |
| 3584 | 23074,73 | 893,21 | 892,72 | — |
| 4096 | 30134,34 | 1166,45 | 1165,88 | 130,51 |

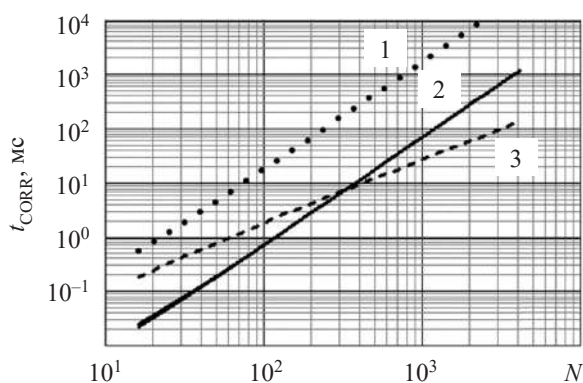


Рис. 2. Залежність часу обчислення ДКФ від розміру масиву даних у разі прямого розрахунку на МК ATmega 1284P (1) та на МК STM32F103C8T6 (2), а також для розрахунку зі застосуванням ШПФ на МК STM32F103C8T6 (3)

означає, що фасетний елемент з універсальним МК задовольняє вимогам АТЗ, у тому числі БПЛА. Слід зазначити, що $N + M = 128$ гарантує динамічний діапазон вимірювача кутової швидкості 64 — 100, що є достатнім для розв’язання задач навігації та орієнтації. Якщо масив має розмір 512 значень, динамічний діапазон може досягати 250 — 400, що робить фасетний елемент точним вимірювачем кутової швидкості;

— МК STM32F103C8T6 забезпечує необхідну швидкість обчислення ДКФ при достатньо низькій ціні. Якщо розмір масиву даних менший за 512 значень, пряме обчислення ДКФ виконується швидше, але у разі 512 значень та більше слід застосовувати алгоритми з ШПФ. Це зумовлено такими факторами, як висока тактова частота МК, 32-розрядна архітектура, виконання усіх команд за один машинний такт, ефективна система команд, особливо — наявність команди множення з додаванням та команд множення чисел зі знаком та без знаку;

— МК ATmega 1284P слід використовувати тільки для макетування фасетних елементів, оскільки він має низьку обчислювальну потужність для забезпечення обчислень ДКФ у реальному часі. Крім того, розробка ФСТЗ може зацікавити не тільки досвідчених інженерів та науковців, а й школярів та студентів, які вже опанували МК з архітектурою AVR та мають бажання створювати оптичні сенсори.

Запропонований фасетний елемент суттєво відрізняється від аналогів [12, 24, 25]. Так, у аналогічних компактних фасетних елементах, які містять МК, ДКФ не обчислюється [12, 24]. Замість цього здійснюється фільтрація низьких частот, порогова обробка та обчислення часу за допомогою таймера між імпульсами — результатами порогової обробки [8—12, 24]. Такий метод вимірювання кутової швидкості не потребує великої кількості обчислень, але його не можна застосовувати, коли співвідношення сигнал/шум є малим і порогова обробка не дозволяє достовірно розділяти корисний сигнал та шумову складову. У [25] приведено цікаву систему для знаходження відстані до перешкод за допомогою оптичних сенсорів у вигляді фасетних елементів з 8 фотодіодами, прикріплених до гвинтів гвинтокрилу. Для знаходження відстані обчислюється нормована ДКФ, але ці розрахунки виконуються за допомогою зовнішнього потужного комп’ютера. Цей розрахунок ДКФ ускладнює необхідність зміни частоти обертів під час польоту, що потребує введення допоміжних часових затримок. Треба зазначити, що запропонований фасетний елемент може забезпечувати обчислення ДКФ у цій системі без використання зовнішнього потужного комп’ютера.

Розроблені авторами варіанти фасетних елементів з МК ATmega 1284P та STM32F103C8T6 представлені на **рис. 3**.

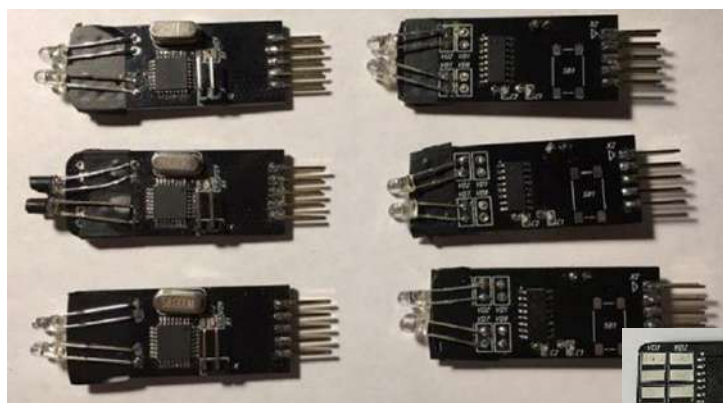
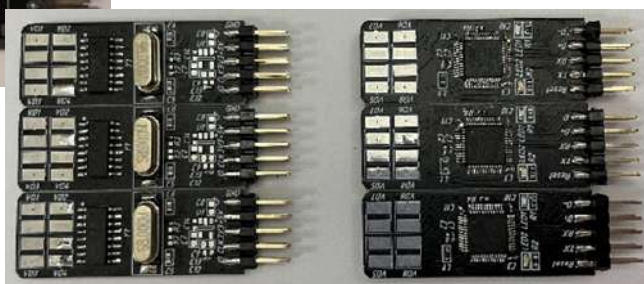


Рис. 3. Фото електронного блоку фасетного елемента з МК ATmega 1284P (ліворуч) та з МК STM32F103C8T6 (праворуч)



Висновки

Отримані результати підтверджують можливість створення недорогих фасетних елементів з використанням універсальних МК з архітектурою ARM, які забезпечують визначення напрямку та значення кутової швидкості об'єктів, що знаходяться у полі зору АТЗ, шляхом обчислення ДКФ сигналів з фотоприймачів за час від 1 до 10 мс. Який метод обчислення ДКФ слід обирати, залежить від розміру масиву даних: якщо менше 512, то швидшим буде пряме обчислення; якщо 512 та більше — суттєво менший час забезпечить обчислення ДКФ з застосуванням ШПФ, але слід пам'ятати, що його можна використовувати тільки у разі виконанні умови $N = 2^p$. При цьому необхідно зазначити, що у більшості випадків можна використовувати пряме обчислення ДКФ, оскільки розміри масиву даних в основному не перевищують 200 — 400 елементів

Усі компоненти пропонованого фасетного елемента є доступними на вітчизняному ринку напівпровідникових компонентів, а це робить можливим їх ма-

сове виробництво. Орієнтована вартість такого фасетного елемента становитиме 400 — 500 грн, а вартість ФСТЗ з 20 — 30 фасетними елементами не перевищуватиме 8 000 — 15 000 гривень.

Дослідження показали, що фасетні елементи на недорогих універсальних мікроконтролерах дозволяють виконати вимірювання шляхом обчислення максимального значення ДКФ сигналів з 4 фотоприймачів за 10,3 мс при розмірі масиву вхідних даних 384 елементи та за 1,2 мс при розмірі до 128 елементів. Тобто запропоновані фасетні системи технічного зору здатні забезпечувати вимірювання розподілу кутових швидкостей рухомих та нерухомих об'єктів, що знаходяться у полі зору автономного транспортного засобу, майже 1000 разів за секунду, що є недосяжним показником для більшості цифрових камер для АТЗ. Це робить такі фасетні системи перспективними сенсорами для навігації, орієнтації та уникнення зіткнень з рухомими та нерухомими об'єктами у майбутніх АТЗ, включно з безпілотними літальними апаратами.

ДОДАТОК А

```
// -----
//          Calculation of discrete correlation function (ARM assembler)
//          64 bits correlation function values, (C) V. Borovytsky, V. Antonenko, 2022
// -----
// Name specifications using                               // Correlation calculation
// assembler directives:                                  M2      ldrh UA, [A1], #2
// A1, A2, B1, NUM_B,                                     ldrh UB, [B1], #2
// CORR_MAX_1, CORR_MAX_0
// CORR_MAX_POS, POS_B
// CORR_1, CORR_0, UA, UB                               umlal CORR_0, CORR_1, UA, UB
// -----                                               cmp A1, A2
// Data addresses initialization                           bne M2
//          ldr A1, DATA_1_Address                       // Calculation of maximum value
//          ldr A2, DATA_1_Address + 2 * N               cmp CORR_MAX_1, CORR_1
//          ldr B1, DATA_2_Address                       bhi M4
//
//          ldr POS_B, #0                                  bls M3
//          ldr NUM_B, M                                   cmp CORR_MAX_0, CORR_0
//
//          ldr CORR_MAX_1, #0                             bhi M4
//          ldr CORR_MAX_0, #0
//          ldr CORR_MAX_POS, POS_B
//
// Data initialization
M1      add B1, POS_B lsl 2
M2      add POS_B, #1
M3      mov CORR_MAX_1, CORR_1
M4      mov CORR_MAX_0, CORR_0
        mov CORR_MAX_POS, POS_B
        cmp POS_B, NUM_B
        bne M1
```


ДОДАТОК А (продовження)

| | |
|---|--|
| Кількість масивів даних | $N_D = 4$ |
| Обсяг пам'яті для зберігання даних | $V_D = N_D \cdot 2 \cdot (N + M)$ |
| Кількість обчислень ДКФ | $N_{CORR} = 4$ |
| Кількість машинних команд, які потрібно виконати: | $N_C = N_{CORR} \cdot (8 + M \cdot (12 + 5 \cdot N))$ |
| Кількість тактів з урахуванням псевдокоманд завантаження | $N_T = N_{CORR} \cdot (11 + M \cdot (12 + 5 \cdot N))$ |
| Час виконання обчислень з урахуванням тактової частоти МК f_T | $t_{CORR} = N_T / f_T$ |

ДОДАТОК Б

```
// -----
//      Calculation of discrete correlation function (ARM assembler)
//      32 bits correlation function values, (C) V. Borovytsky, V. Antonenko, 2022
// -----
// Name specifications using
// assembler directives:
// A1, A2, B1, NUM_B,
// CORR_MAX, CORR_MAX_POS, POS_B
// CORR, UA, UB
// -----
// Data addresses initialization
    ldr A1, DATA_1_Address
    ldr A2, DATA_1_Address + 2 * N
    ldr B1, DATA_2_Address

    ldr POS_B, #0
    ldr NUM_B, M

    ldr CORR_MAX, #0
    ldr CORR_MAX_POS, POS_B

// Data initialization
M1    add B1, POS_B lsl 2

// Correlation calculation
M2    ldrh UA, [A1], #2
        ldrh UB, [B1], #2
        mla CORR, UA, UB
        cmp A1, A2
        bne M2

// Calculation of maximum value
        cmp CORR_MAX, CORR

// Assigning the new maximum value
        movls CORR_MAX, CORR
        movls CORR_MAX_POS, POS_B

        add POS_B, #1
        cmp POS_B, NUM_B
        bne M1
```

| | |
|---|---|
| Кількість масивів даних | $N_D = 4$ |
| Обсяг пам'яті для зберігання даних | $V_D = N_D \cdot 2 \cdot (N + M)$ |
| Кількість обчислень ДКФ | $N_{CORR} = 4$ |
| Кількість машинних команд, які потрібно виконати: | $N_C = N_{CORR} \cdot (7 + M \cdot (7 + 5 \cdot N))$ |
| Кількість тактів з урахуванням псевдокоманд завантаження | $N_T = N_{CORR} \cdot (10 + M \cdot (7 + 5 \cdot N))$ |
| Час виконання обчислень з урахуванням тактової частоти МК f_T | $t_{CORR} = N_T / f_T$ |

ДОДАТОК В

```
// -----
//      Calculation of discrete correlation function (AVR assembler)
//      32 bits correlation function values, (C) V. Borovytsky, V. Antonenko, 2022
// -----
// Name specifications using                               // Correlation calculation
// assembler directives:                                  M2:  ldr UA, X+
// A1_H (XH), A1_L (XL),                                  ldr UB, Z+
// A2_H (YH), A2_L (YL),                                  mul UA, UB
// B2_H (ZH), B2_L (ZL),                                  add CORR_0, UB
// POS_B_H, POS_B_L                                       adc CORR_1, UA
// NUM_B_H, NUM_B_L                                       adc CORR_2, R_ZERO
// CORR_MAX_3, CORR_MAX_2                                  adc CORR_3, R_ZERO
// CORR_MAX_1, CORR_MAX_0                                  cmp A1_L, A2_L
// CORR_MAX_POS_1,                                         brne M2
// CORR_MAX_POS_0                                         cmp A1_H, A2_H
// CORR_3, CORR_2, CORR_1, CORR_0                         brne M2
// UA, UB, R_ZERO
// -----
// Data addresses initialization                            // Calculation of maximum value
ldi A1_H, High( DATA_1_Address )                        cmp CORR_MAX_3, CORR_3
ldi A1_L, Low( DATA_1_Address )                          brgt M4
ldi A1_H, High(DATA_1_Address + N)                        brlt M3
ldi A1_L, Low( DATA_1_Address + N)
ldi B1_H, High( DATA_2_Address )
ldi B1_L, Low( DATA_1_Address )
ldi NUM_B_H, High(DATA_1_size)
ldi NUM_B_L, Low( DATA_1_size)

clr POS_B_H
clr POS_B_L
clr R_ZERO

clr CORR_MAX_3
clr CORR_MAX_2
clr CORR_MAX_1
clr CORR_MAX_0

clr CORR_MAX_POS_1
clr CORR_MAX_POS_0

// Data initialization
M1:  ldi B1_H, High( DATA_B )
     ldi B1_L, Low( DATA_B )

     add B1_L, POS_B_L
     adc B1_H, POS_B_H

M2:  ldr UA, X+
     ldr UB, Z+
     mul UA, UB
     add CORR_0, UB
     adc CORR_1, UA
     adc CORR_2, R_ZERO
     adc CORR_3, R_ZERO
     cmp A1_L, A2_L
     brne M2
     cmp A1_H, A2_H
     brne M2

// Calculation of maximum value
cmp CORR_MAX_3, CORR_3
brgt M4
brlt M3

cmp CORR_MAX_2, CORR_2
brgt M4
brlt M3

cmp CORR_MAX_1, CORR_1
brgt M4
brlt M3

cmp CORR_MAX_0, CORR_0
brgt M4

// Assigning the new maximum value
M3:  mov CORR_MAX_3, CORR_3
     mov CORR_MAX_2, CORR_2
     mov CORR_MAX_1, CORR_1
     mov CORR_MAX_0, CORR_0
     mov CORR_MAX_POS_1, B_POS_1
     mov CORR_MAX_POS_0, B_POS_0

M4:  inc POS_B_L
     adc POS_B_H, R_ZERO

     cmp POS_B_L, NUM_B_L
     brne M1
     cmp POS_B_H, NUM_B_H

     brne M1
```

Кількість масивів даних $N_D = 4$
 Обсяг пам'яті для зберігання даних $V_D = N_D \cdot (N + M)$
 Кількість обчислень ДКФ $N_{CORR} = 4$
 Кількість машинних команд, які потрібно виконати: $N_C = N_{CORR} \cdot (27 + M \cdot (27 + 11 \cdot N))$
 Кількість тактів з урахуванням псевдокоманд завантаження $N_T = N_{CORR} \cdot 2 \cdot (47 + M \cdot (33 + 17 \cdot N))$
 Час виконання обчислень з урахуванням тактової частоти МК f_T : $t_{CORR} = N_T / f_T$

ДОДАТОК Г

```

// -----
//      Calculation of discrete correlation function using FFT (ARM assembler)
//      (C) V. Borovytsky, V. Antonenko, 2022
// -----
// Copy data Re(), Im() parts                                // -----
//   to spectrum array
// -----
// Name specifications using                                // Search correlation
// assembler directives:                                  //   maximum and
// A1, A2, B1, U, R_zero                                   //   its position
// -----
// Name specifications using                                // -----
// assembler directives:                                  // Name specifications using
// A1, A2, NUM                                            // assembler directives:
// U1_Re, U1_Im, U2_Re, U2_Im                            // A1, A2, NUM
// -----
// U1_Re, U1_Im, U2_Re, U2_Im                            // U1_Re, U1_Im, U2_Re, U2_Im
// -----
INIT_SPECTR PROC
    ldr A1, Data_Address
    ldr A2, Data_Address + 2*N
    ldr B1, Spectrum_Address
    ldr R_zero, #0
    M1   ldrh U,    [A1], #2
        str  U,    [B1], #4
        str  R_zero, [B1], #4
        cmp  A1, A2
        bne  M1
        ENDP
// -----
// Spectrum multiplication
//   S1 · S2*
// -----
// Name specifications using
// assembler directives:
// A1, A2, B1,
// U1_Re, U1_Im, U2_Re, U2_Im
// -----
SPECTR_MULT PROC
    ldr A1, Spectrum_1_Address
    ldr A2, Spectrum_1_Address+4*N
    ldr B1, Spectrum_2_Address
    M1   ldr  U1_Re, [A1], #4
        ldr  U1_Im, [A1], #4
        ldr  U2_Re, [B1], #4
        ldr  U2_Im, [B1], #4
        smull U0, U1, U1_Re, U2_Re
        smull U2, U3, U1_Im, U2_Im
        add  U1, U3
        str  U1, [ A1, #-8 ]
        smull U0, U1, U1_Re, U2_Im
        smull U2, U3, U1_Im, U2_Re
        sub  U3, U1
        str  U1, [ A1, #-4 ]
        cmp  A1, A2
        bne  M1
        ENDP
// -----
// Fast Fourier Transform
// -----
// Name specifications using
// assembler directives:
// A1, B1, REP_NUM
// bnum, bnum_max,
// bpos, bstep, U
// X1_Re, X1_Im, Y1_Re, Y1_Im
// -----
FFT PROC
    ldr bnum_max, N lsr 1
    ldr bstep, #1
    ldr rep_num, #0
    M1   ldr bnum, #0
        mul bbase, bnum, bstep
    M2   ldr bpos, #0
    M3   ldr A1, Spect_Address
        add B1, A1, bstep
    
```

ДОДАТОК Г (продовження)

```

ldr X1_Re, [A1], #4
ldr X1_Im, [A1]
ldr Y1_Re, [B1], #4
ldr Y1_Im, [B1]

// -----
// "Butterfly" calculation
// X2 = X1 + Y1
// Y2 = ( X1 - Y1 ) * W(k,N)
// W(k,N)=exp(- 2 * Pi * j * k / N )
// -----

add U, X1_Re, Y1_Re
str U, [A1, #-4]

add U, X1_Im, Y1_Im
str U, [A1]

sub X1_Re, Y1_Re
sub X1_Im, Y1_Im

ldr A1, Cos_Sin_Table_Address
mla A1, bnum_max lsl 2, bpos, A1

ldr U, [A1], #4 // load cos-value

smull Y1_Re, U, X1_Re, U
str U, [B1, #-4]

ldr U, [A1] // load sin-value
smull Y1_Im, U, X1_Im, U
str U, [B1]

add bpos, #1
cmp bpos, bstep
bne M3

add bnum, #1
ldr U, NUM_MAX
cmp bnum, U
bne M2

mov bnum_max, bnum_max lsr 1
mov bstep, bstep lsl 1

add rep_num, #1
ldr U, LOG2N
cmp rep_num, U
bne M1
ENDP
    
```

Кількість масивів даних $N_D = 4$

Обсяг пам'яті для зберігання даних $V_D = (4 \cdot N_D + 4 \cdot 4 + 4 \cdot 1) \cdot N$

Кількість обчислень ДКФ $N_{CORR} = 2$

Кількість машинних команд, як треба виконати:

- 1) підготовка двох масивів комплексних даних:
 $N_{C1} = N_{CORR} \cdot (4 + 5 \cdot N)$
- 2) два прямих швидких перетворення Фур'є:
 $N_{C2} = N_{CORR} \cdot (3 + \log_2(N) \cdot (8 + 28 \cdot N)) = N_{CORR} \cdot (6 + 8 \cdot \log_2(N) + 28 \cdot \log_2(N) \cdot N)$
- 3) перемноження двох комплексних спектрів:
 $N_{C3} = N_{CORR} \cdot (3 + 14 \cdot N)$
- 4) одне зворотне швидке перетворення Фур'є:
 $N_{C4} = N_{CORR} \cdot (3 + \log_2(N) \cdot (8 + 28 \cdot N)) = N_{CORR} \cdot (6 + 8 \cdot \log_2(N) + 28 \cdot \log_2(N) \cdot N)$
- 5) розрахунок максимального значення ДКФ та його індексу:
 $N_{C5} = N_{CORR} \cdot (5 + 7 \cdot N)$

Загальна кількість машинних команд, як треба виконати:

$$N_C = 2 \cdot N_{C1} + 2 \cdot N_{C2} + N_{C3} + N_{C4} + N_{C5} =$$

$$= N_{CORR} \cdot (25 + 24 \cdot \log_2(N) + (31 + 84 \cdot \log_2(N)) \cdot N)$$

Кількість тактів з урахуванням псевдокоманд завантаження:

- 6) підготовка двох масивів комплексних даних:
 $N_{T1} = N_{CORR} \cdot (7 + 5 \cdot N)$
- 7) два прямих швидких перетворення Фур'є:
 $N_{T2} = N_{CORR} \cdot (4 + \log_2(N) \cdot (9 + 31 \cdot N)) = N_{CORR} \cdot (4 + 9 \cdot \log_2(N) + 31 \cdot \log_2(N) \cdot N)$
- 8) перемноження двох комплексних спектрів:
 $N_{T3} = N_{CORR} \cdot (6 + 14 \cdot N)$
- 9) одне зворотне швидке перетворення Фур'є:
 $N_{T4} = N_{CORR} \cdot (4 + \log_2(N) \cdot (9 + 31 \cdot N)) = N_{CORR} \cdot (4 + 9 \cdot \log_2(N) + 31 \cdot \log_2(N) \cdot N)$

ДОДАТОК Г (продовження)

10) розрахунок максимального значення ДКФ та його індексу:

$$N_{T5} = N_{CORR} \cdot (7 + 7 \cdot N)$$

Загальна кількість машинних тактів, які треба виконати для обчислення ДКФ:

$$N_T = 2 \cdot N_{C1} + 2 \cdot N_{C2} + N_{C3} + N_{C4} + N_{C5} =$$

$$= N_{CORR} \cdot (39 + 27 \cdot \log_2(N)) + (31 + 93 \cdot \log_2(N)) \cdot N$$

Час виконання обчислень t_{CORR} з урахуванням тактової частоти МК f_T : $t_{CORR} = N_T / f_T$

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Custers B. *The Future of Drone Use*. Hague: TMC Asser Press, 2016. <https://doi.org/10.1007/978-94-6265-132-6>
2. D'Andrea R. Can drones deliver? *IEEE Trans. on Automation Science and Engineering*, 2014, no. 11(3), pp. 647–648. <https://doi.org/10.1109/TASE.2014.2326952>
3. Gallego G., Delbruck T., Orchard G. et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, vol. 44, iss. 1, pp. 154–180. <https://doi.org/10.1109/TPAMI.2020.3008413>
4. Barrios-Avilés J., Iakymchuk T., Samaniego J. et al. Movement detection with event-based cameras: Comparison with frame-based cameras in robot object tracking using powerlink communication. *Electronics*, 2018, no. 7(11), 304, pp. 1–19. <https://doi.org/10.3390/electronics7110304>
5. Döge J., Hoppe C., Reichel P. et al. A 1 megapixel HDR image sensor soc with highly parallel mixed-signal processing. *Proc. of 2015 IISW*, 2015, Vaals, Netherlands. <https://bit.ly/3xLbDHI>
6. Franceschini N. Small brains, smart machines: from fly vision to robot vision and back again. *Proc. of IEEE*, 2014, no. 102, pp. 1–31. <https://doi.org/10.1109/JPROC.2014.2312916>
7. Leitel R., Brückner A., Buss W. et al. Curved artificial compound-eyes for autonomous navigation. *Proc. of SPIE*, 2014, vol. 9130, pp. 91300H. <https://doi.org/10.1117/12.2052710>
8. Colonnier F., Ramirez-Martinez S., Viollet S. et al. A bio-inspired sighted robot chases like a hoverfly. *Bioinspiration&Biomimetics*, 2019, vol. 14, no. 3, 036002. <https://doi.org/10.1088/1748-3190/aaffa4>
9. Juston R., Viollet S. A miniature bio-inspired position sensing device for the control of micro-aerial robots. *Proc. of IEEE/RSJ*, 2012, Vilamoura, pp. 1118–1124. <http://dx.doi.org/10.1109/IROS.2012.6385937>
10. Colonnier F., Manecy A., Juston R. et al. A small-scale hyperacute compound eye featuring active eye tremor: Application to visual stabilization, target tracking, and short-range odometry. *Bioinspiration&biomimetics*, 2015, no. 10(2), 026002. <https://doi.org/10.1088/1748-3190/10/2/026002>
11. Zhao J., Hu C., Zhang C. et al. A bio-inspired collision detector for small quadcopter. *Proc. of IJCNN 2018*, 2018, pp. 1–7. <https://doi.org/10.1109/IJCNN.2018.8489298>
12. Ruffier F., Viollet S., Amic S. et al. Bio-inspired optical flow circuits for the visual guidance of micro air vehicles. *Proc. of ISCAS 2003*, 2003, pp. III-III. <https://doi.org/10.1109/ISCAS.2003.1205152>
13. Aubépart F., Serres J., Dilly A. et al. Field programmable gate array (FPGA) for bio-inspired visio-motor control systems applied to micro-air vehicles. *Aerial Vehicles*. Delph, InTech, 2009. <https://doi.org/10.5772/6466>
14. Mockel R. *Bio-Inspired Optical Flow Vision Sensors for Visual Guidance of Autonomous Robots*. Dissertation ... Doctor of Sciences – DISS. ETH no. 20736, 1981, 270 p. <https://www.research-collection.ethz.ch/handle/20.500.11850/153958>
15. Баглицкий В.К. *Корреляционно-экстремальные методы навигации и наведения*, Тверь, Книжный клуб, 2009, 360 с.
16. Боровицький В., Антоненко В. *Датчик швидкості для безпілотного літального апарата*. Патент України № 143 242, 27.07.2020, бюл. № 14.
17. Borovytsky V., Antonenko V. Biologically inspired compound eye. *Proc. SPIE*, 2020, vol. 11369, pp. 113691T. <https://doi.org/10.1117/12.2553678>
18. Valavanis K.P., Vachtsevanos G. J. *Handbook of Unmanned Aerial Vehicles*. Heidelberg, Springer Publishing Company Inc., 2014, 3022 p. <https://doi.org/10.5555/2692452>
19. Нуссбаумер Г. *Быстрое преобразование Фурье и алгоритмы вычисления свёрток*. Москва: Радио и связь, 1985, 248 с.
20. Microchip Technology Inc. 8-bits MCUs. AVR® MCUs, <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus>
21. *STMicroelectronics*. STM32F103, <https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html>
22. Ревич Ю. В. *Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера*. Санкт-Петербург, БХВ-Петербург, 2011, 352 с.
23. Козаченко В. Ф. *Практический курс микропроцессорной техники на базе процессорных ядер ARM-Cortex-M3/M4/M4F*, Москва, МЭИ, 2019, 543 с.
24. Roubieu F. L., Expert F., Boyron M. et al. A novel 1-gram insect based device measuring visual motion along 5 optical directions. *Sensors*, 2011 IEEE Proc., 2011, Limerick, Ireland. pp. 687–692. <https://doi.org/10.1109/ICSENS.2011.6127157>
25. Viel C., Viollet S. Fast normalized cross-correlation for measuring distance to objects using optic flow, applied for helicopter obstacle detection. *Measurement*, 2020, vol. 172, no. 5, pp. 108911–108921. <https://doi.org/10.1016/j.measurement.2020.108911>

Дата надходження рукопису до редакції 20.01 2022 р.

Опис статті для цитування:

Боровицький В. М., Антоненко В. А. Розрахунок дискретної кореляційної функції у фасетних системах технічного зору. Технологія і конструювання в електронній апаратурі, 2022, № 1–3, с. 8–21. <http://dx.doi.org/10.15222/ТКЕА2022.1-3.08>

Cite the article as:

Borovytsky V. N., Antonenko V. A. Calculation of discrete correlation function in facet systems of technical vision. *Tekhnologiya i Konstruirovaniye v Elektronnoy Apparature*, 2022, no. 1–3, pp. 8–21. <http://dx.doi.org/10.15222/ТКЕА2022.1-3.08>

CALCULATION OF DISCRETE CORRELATION FUNCTION IN FACET SYSTEMS OF TECHICAL VISION

The paper proposes a facet vision system composed from identical facet elements. Each facet element contains an optical system, several photodetectors, a preamplifier, and a universal microcontroller. In such a system, all facet elements operate independently of each other. Each facet element performs fast measurements of the angular velocity of objects in its field of view by calculating the discrete correlation functions of the signals from the photodetectors. The paper considers the possibility of using economical microcontrollers in facet elements for fast calculation of the discrete correlation functions. The authors perform a comparative analysis of the techniques based on the direct calculation and the calculation with fast Fourier transform. The investigation of the corresponding program code for microcontrollers in assembly language is done with calculations of the number of machine instructions and their execution time. The study confirms that economical universal microcontrollers are able to perform fast measurements by finding the maximum values of discrete correlation functions. In the case of receiving signals from 4 photodetectors, the calculation time is less than 10 milliseconds for input data arrays of 384 elements and less than 1.2 milliseconds for input data arrays of 128 elements. These results make the proposed facet vision systems applicable in navigation, orientation, and collision avoidance with moving and stationary objects in automatic vehicles, including unmanned aerial vehicles.

Keywords: vision systems, digital signal processing, angular velocity measurement, discrete correlation function, digital cameras, unmanned aerial vehicles.

REFERENCES

1. Custers B. *The Future of Drone Use*. Hague: TMC Asser Press, 2016. <https://doi.org/10.1007/978-94-6265-132-6>
2. D'Andrea R. Can drones deliver? *IEEE Trans. on Automation Science and Engineering*, 2014, no. 11(3), pp. 647–648. <https://doi.org/10.1109/TASE.2014.2326952>
3. Gallego G., Delbruck T., Orchard G. et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, vol. 44, iss. 1, pp. 154–180. <https://doi.org/10.1109/TPAMI.2020.3008413>
4. Barrios-Avilés J., Iakymchuk T., Samaniego J. et al. Movement detection with event-based cameras: Comparison with frame-based cameras in robot object tracking using powerlink communication. *Electronics*, 2018, no. 7(11), 304, pp. 1–19. <https://doi.org/10.3390/electronics7110304>
5. Döge J., Hoppe C., Reichel P. et al. A 1 megapixel HDR image sensor soc with highly parallel mixed-signal processing. *Proc. of 2015 IISW*, 2015, Vaals, Netherlands. <https://bit.ly/3xLbDhI>
6. Franceschini N. Small brains, smart machines: from fly vision to robot vision and back again. *Proc. of IEEE*, 2014, no. 102, pp. 1–31. <https://doi.org/10.1109/JPROC.2014.2312916>
7. Leitel R., Brückner A., Buss W. et al. Curved artificial compound-eyes for autonomous navigation. *Proc. of SPIE*, 2014, vol. 9130, pp. 91300H. <https://doi.org/10.1117/12.2052710>
8. Colonnier F., Ramirez-Martinez S., Viollet S. et al. A bio-inspired sighted robot chases like a hoverfly. *Bioinspiration&Biomimetics*, 2019, vol. 14, no. 3, 036002. <https://doi.org/10.1088/1748-3190/aaffa4>
9. Juston R., Viollet S. A miniature bio-inspired position sensing device for the control of micro-aerial robots. *Proc. of IEEE/RSJ*, 2012, Vilamoura, pp. 1118–1124. <http://dx.doi.org/10.1109/IROS.2012.6385937>
10. Colonnier F., Manecy A., Juston R. et al. A small-scale hyperacute compound eye featuring active eye tremor: Application to visual stabilization, target tracking, and short-range odometry. *Bioinspiration&Biomimetics*, 2015, no. 10(2), 026002. <https://doi.org/10.1088/1748-3190/10/2/026002>
11. Zhao J., Hu C., Zhang C. et al. A bio-inspired collision detector for small quadcopter. *Proc. of IJCNN 2018*, 2018, pp. 1–7. <https://doi.org/10.1109/IJCNN.2018.8489298>
12. Ruffier F., Viollet S., Amic S. et al. Bio-inspired optical flow circuits for the visual guidance of micro air vehicles. *Proc. of ISCAS 2003*, 2003, pp. III-III. <https://doi.org/10.1109/ISCAS.2003.1205152>
13. Aubépart F., Serres J., Dilly A. et al. Field programmable gate array (FPGA) for bio-inspired visio-motor control systems applied to micro-air vehicles. *Aerial Vehicles*. Delph, InTech, 2009. <https://doi.org/10.5772/6466>
14. Mockel R. *Bio-Inspired Optical Flow Vision Sensors for Visual Guidance of Autonomous Robots*. Dissertation... Doctor of Sciences – DISS. ETH no. 20736, 1981, 270 p. <https://www.research-collection.ethz.ch/handle/20.500.11850/153958>
15. Baklitskiy V.K. *Korrelyatsionno-ekstremal'nyye metody navigatsii i navedeniya* [Correlation-extreme methods of navigation and guidance]. Russia, Tver, Publisher book club, 2009, 360 p. (Rus)
16. Borovytsky V., Antonenko V. [Speed sensor for unmanned aerial vehicle] Pat.UA no. 142 242, 2020, bul. 14 (Ukr)
17. Borovytsky V., Antonenko V. Biologically inspired compound eye. *Proc. SPIE*, 2020, vol. 11369, pp. 113691T. <https://doi.org/10.1117/12.2553678>
18. Valavanis K.P., Vachtsevanos G. J. *Handbook of Unmanned Aerial Vehicles*. Heidelberg, Springer Publishing Company Inc., 2014, 3022 p. <https://doi.org/10.5555/2692452>
19. Nussbaumer G. *Bystroye preobrazovaniye Fur'ye i algoritmy vychisleniya sv'ortok* [Fast Fourier Transform and Convolution Algorithms]. Russia, Moscow, Radio and communications, 1985, 248 p. (Rus)
20. Microchip Technology Inc. 8-bits MCUs. AVR® MCUs, <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus>
21. *STMicroelectronics*. STM32F103, <https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html>
22. Revich Yu. V. *Prakticheskoye programmirovaniye mikrokontrolerov Atmel AVR na yazyke assemblera* [Practical programming of Atmel AVR microcontrollers in assembly language]. Russia, Saint-Petersburg, BHV-Petersburg, 2011, 352 p. (Rus)
23. Kozachenko V. F. *Prakticheskij kurs mikroprotsessornoy tekhniki na baze protsessornykh yader ARM-Cortex-M3/M4/M4F* [Practical course of microprocessor technology based on ARM-Cortex-M3/M4/M4F processor cores]. Russia, Moscow: MEI, 2019, 543 p. (Rus)
24. Roubieuf F. L., Expert F., Boyron M. et al. A novel 1-gram insect based device measuring visual motion along 5 optical directions. *Sensors, 2011 IEEE Proc.*, 2011, Limerick, Ireland. pp. 687–692. <https://doi.org/10.1109/ICSENS.2011.6127157>
25. Viel C., Viollet S. Fast normalized cross-correlation for measuring distance to objects using optic flow, applied for helicopter obstacle detection. *Measurement*, 2020, vol. 172, no. 5, pp. 108911–108921. <https://doi.org/10.1016/j.measurement.2020.108911>