

Д. І. Савчук¹, Ю. Є. Кинаш¹, О. С. Вітер², В. М. Мицишин²
Національний університет "Львівська політехніка",
¹ кафедра інформаційних технологій видавничої справи,
² кафедра комп'ютеризованих систем автоматички

СТВОРЕННЯ ГРАФІЧНИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ДИСКРЕТНОГО ПЕРЕТВОРЕННЯ ФУР'Є

<https://doi.org/>

© Савчук Д. І., Кинаш Ю. Є., Вітер О. С., Мицишин В. М., 2021

Розроблено систему для створення графічних об'єктів із застосуванням дискретного перетворення Фур'є та передачі даних у процесі проведення онлайн-конференцій, лекцій, зустрічей тощо. Розроблена система сприяє покращенню роботи викладачів із онлайн-дошками, оскільки мінімізує затримки при передачі зображень. Запропонований алгоритм можна застосувати для передавання звукових сигналів. Для створення системи використана мова Java Processing.

Ключові слова – комп'ютерна графіка, дискретне перетворення Фур'є, онлайн-дошки.

The system for creating graphic objects using discrete Fourier transform and data transmission during online conferences, lectures, meetings, etc. has been developed. The developed system improves the work of teachers with online boards, as it minimizes delays in the transmission of images. The proposed algorithm can be used to transmit audio signals. The Java Processing language was used to create the system.

Keywords: computer graphics, discrete Fourier transform, online boards.

1. Постановка проблеми

Дискретне перетворення Фур'є одним з перетворень, що широко застосовується у алгоритмах цифрової обробки сигналів, а його модифікації застосовуються в стисненні звуку MP3 формату, стисненні зображень JPEG і ін., а також для інших областей, що пов'язані з аналізом частот в дискретному сигналі. Дискретне перетворення Фур'є вимагає на вході дискретну функцію. Такі функції часто створюються шляхом дискретизації, тобто вибірки значень з безперервних функцій. Дискретне перетворення Фур'є допомагає рішати диференціальні рівняння і виконувати такі операції, як згортки. Дискретне перетворення Фур'є також активно використовуються в статистиці, при аналізі часових рядів. Існують також багатовимірні дискретні перетворення Фур'є.

Отже, дискретне перетворення Фур'є можна використати для роботи з графічними редакторами для яких є можливість передачі даних за допомогою цифрових алгоритмів обробки сигналів.

2. Застосування перетворення Фур'є для роботи із зображеннями

Перетворення Фур'є є інтегральним перетворенням однієї комплексної функції дійсної змінної на іншу. Воно тісно пов'язане з перетворенням Лапласа та аналогічне розкладу у ряд Фур'є для

неперіодичних функцій. Це перетворення розкладає дану функцію на осциляторні функції і використовується для розрахунків спектру частот сигналів, що є змінними у часі [1-3].

Перетворенням Фур'є (ФТ) функції $f(x)$ є функція $F(\omega)$, де:

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x} dx$$

а обернене перетворення Фур'є визначається так:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{i\omega x} d\omega$$

Це перетворення є переходом від одного до іншого набору базових функцій. Основними функціями перетворення Фур'є використовуються складні експоненти (синусоїди) різних частот.

Перетворення Фур'є часто записують $f(x) \leftrightarrow F(\omega)$, або $F(f(x)) = F(\omega)$, де F є оператором перетворення Фур'є. Якщо $f(x)$ розглядається як сигнал (тобто вхідні дані), то $F(\omega)$ називається спектром сигналу. Якщо f розглядається, як імпульсна характеристика фільтра, що працює на вхідних даних для отримання вихідних даних, то F називається частотною характеристикою фільтра.

Коли сигнал дискретний і періодичний, нам не потрібне безперервне перетворення Фур'є. Натомість ми використовуємо дискретне перетворення Фур'є або DFT (Discrete Fourier Transform). DFT є математична процедура, що використовується для визначення гармонічного, або частотного, складу дискретних сигналів [4-8]. Припустимо, наш сигнал дорівнює для $n = 0 \dots N - 1$, і $a_n = a_{n+jN}$ для всіх n та j :

$$A_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn} a_n$$

або у більш поширеній формі:

$$A_k = \sum_{n=0}^{N-1} W_N^{kn} a_n$$

де

$$W_N = e^{-i\frac{2\pi}{N}}$$

та W_N^k при $k = 0 \dots N - 1$ називаються N -м коренем з одиниці. Їх називають так, тому що $(W_N^k)^N = 1$ для всіх k . Вони представляють собою вершини правильного багатокутника, вписаного в одиничному колі комплексної площини з однією вершиною в точці $(1, 0)$. На рис.1 зображено графіки у комплексній площині комплексного кореня для $N = 2$, $N = 4$ та $N = 8$.

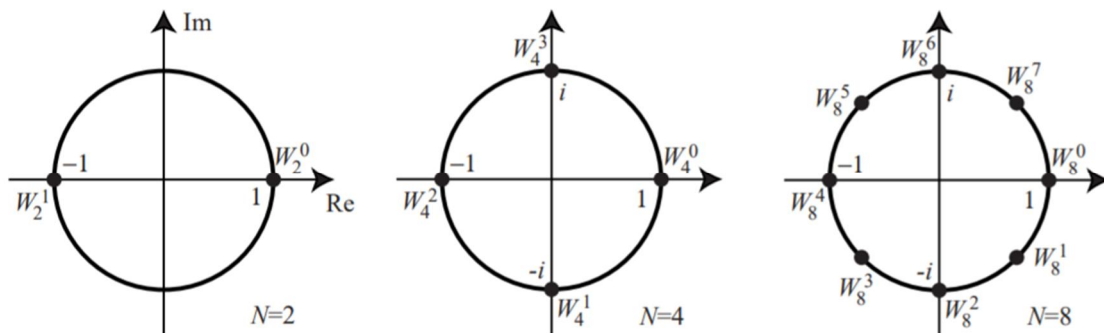


Рис. 1. Графіки у комплексній площині для $N = 2$, $N = 4$ та $N = 8$

N -ті комплексні корені є точками на комплексному одиничному колі через кожні $2\pi/N$ радіанів і множення на W_N еквівалентно повороту за цим кутом за годинниковою стрілкою. Множення на W_N^N є обертання на 2π радіанів, тобто взагалі без обертання. Загалом, $W_N^k = W_N^{k+jN}$ для всіх цілих чисел j . Таким чином, при піднятті W_N до степеня показник степеня може бути прийнятий за модулем N .

Послідовність A_k є дискретним перетворенням Фур'є послідовності a_n . Послідовність a_n є оберненим дискретним перетворенням Фур'є послідовності A_k . Формула для оберненого DFT:

$$a_n = \frac{1}{N} \sum_{k=0}^{N-1} W_N^{-kn} A_k$$

Формула є ідентичною, за винятком того, що a та A обмінялися ролями, як і k та n .

У випадку двоточкового DFT ($N = 2$) маємо:

$$W_2 = e^{-i\pi} = -1$$

$$A_k = \sum_{n=0}^1 (-1)^{kn} a_n = (-1)^{k \cdot 0} a_0 + (-1)^{k \cdot 1} a_1 = a_0 + (-1)^k a_1$$

Отже,

$$A_0 = a_0 + a_1$$

$$A_1 = a_0 - a_1$$

Для випадку чотирьохточкового DFT ($N = 4$) маємо:

$$W_4 = e^{-i\pi/2} = -i$$

$$A_k = \sum_{n=0}^3 (-i)^{kn} a_n = a_0 + (-i)^k a_1 + (-i)^{2k} a_2 + (-i)^{3k} a_3 = a_0 + (-i)^k a_1 + (-1)^k a_2 + i^k a_3$$

Отже,

$$A_0 = a_0 + a_1 + a_2 + a_3$$

$$A_1 = a_0 - ia_1 - a_2 + ia_3$$

$$A_2 = a_0 - a_1 + a_2 - a_3$$

$$A_3 = a_0 + ia_1 - a_2 - ia_3$$

Це також можна записати як матричне множення:

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Для зручного обчислення можемо виконати наступні перетворення:

$$A_0 = (a_0 + a_2) + (a_1 + a_3)$$

$$A_1 = (a_0 - a_2) - i(a_1 - a_3)$$

$$A_2 = (a_0 + a_2) - (a_1 + a_3)$$

$$A_3 = (a_0 - a_2) + i(a_1 - a_3)$$

Двовимірне дискретне перетворення Фур'є є простим узагальненням стандарту 1-D DFT:

$$A_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_M^{km} W_N^{ln} a_{m,n}$$

Це загальна формула, яка підходить для двовимірних зображень, розміри яких не є обов'язково степенями двох.

Для обчислення перетворення Фур'є зображення потрібно:

- Обчислити 1-D DFT кожного рядка
- Обчислити 1-D DFT кожного стовпця

Для цілей відображення потрібно циклічно перекласти зображення так, щоб піксель (0,0), який тепер містить частоту $(\omega_x, \omega_y) = (0, 0)$, перемістився до центру зображення. Також потрібно відобразити значення пікселів пропорційно до величини кожного комплексного числа. Для кольорових зображень потрібно виконувати перетворення для кожного з трьох каналів (R, G та B) незалежно.

3. Мета роботи

Метою роботи є розроблення системи для створення графічних об'єктів, що використовуються в онлайн-конференціях і передача їх зображень із застосуванням дискретного перетворення Фур'є.

4. Опис роботи системи

4.1. Вхідна та вихідна інформація

Вхідною інформацією при використанні системи виступає масив координат точок (пікселів), які створив користувач програми.

На виході отримуємо візуалізацію процесу створення користувачького малюнку за допомогою дискретних перетворень Фур'є, в результаті котрого утвориться ряд кіл, які мають:

- Частоту появи
- Період обертання
- Діаметр

Отримані кола рекурсивно передаються координати, щоб утворити нове коло, кількість котрих залежить від складності малюнка та кількості створених користувачем точок.

Процедура роботи програми є наступною:

- 1) Користувач запускає програму.
- 2) Користувач затискає ліву кнопку мишки, щоб залишатися у режимі «Створення» та починає відтворювати обраний набір символів чи об'єктів.
- 3) Програма обробляє масив точок, як цифровий сигнал, який розділяє на свої частоти. Далі генеруються комплексні дані з амплітудами, фазами, періодами для генерації епіциклів.
- 4) Епіцикли сортуються від більшого до меншого за амплітудою, відтворюються на полотні та відображають створену користувачем картинку.

4.2. Опис алгоритму задачі

Після запуску програма перебуває у режимі створення. Тобто користувач може створити відразу обраний двовимірний об'єкт. Одна ітерація відтворення відбувається до тих пір, поки користувач не відпустить ліву кнопку миші. Відразу після цього починається розрахунок та малювання.

Розрахунок відбувається наступним чином:

Отримуються координати кожної точки. За реальну компоненту береться координата x , за уявну – y .

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi kn}{N}} = \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N} kn\right) - i \cdot \sin\left(\frac{2\pi}{N} kn\right) \right]$$

Згідно формули, наведеної вище, частота при перетворенні виражається значенням k , амплітуда рахується за формулою:

$$\frac{1}{N} |X_k| = \frac{1}{N} \sqrt{\text{Re}(X_k)^2 + \text{Im}(X_k)^2}$$

Фаза вираховується арктангенсом подвійного аргументу:

$$\text{arg}(X_k) = \text{atan2}(\text{Im}(X_k), \text{Re}(X_k)) = -i \cdot \ln\left(\frac{X_k}{|X_k|}\right)$$

Отримується значення в степені експоненти за допомогою добутку частоти на момент часу з додаванням фази зміщення.

Відповідно до запропонованого алгоритму написано код програми для роботи із зображеннями.

4.3. Опис програмного продукту

Для створення програмного продукту було застосоване середовище розробки на основі мови програмування Java Processing, яка призначена для роботи з графікою та зображеннями.

Processing є безкоштовною графічною бібліотекою та інтегрованим середовищем розробки (IDE), що створені для електронного мистецтва, нових медіа та візуального дизайну з метою навчання основам комп'ютерного візуального програмування. Processing використовує мову програмування Java, з додатковими функціями, такими як додаткові класи і псевдоніми математичні функції та операції. Середовище також містить графічний інтерфейс для спрощення етапу компіляції та виконання. Мова Processing та IDE були попередником інших проєктів, включаючи Arduino, Wiring та p5.js. Це середовище використовується студентами, художниками, дизайнерами, дослідниками та любителями, для вивчення прототипування та виробництва [9-12].

Для коректної роботи програми у середовищі Processing, обов'язковими є 2 функції – setup і draw.

Функція Setup відповідає за підготовку даних до подальшої обробки. Ця функція використовується для завантаження зовнішніх даних, наприклад, зображення.

Функція Draw виконується з частотою 30 разів на секунду і відповідає за відображення та оновлення у режимі реального часу.

Також програма містить клас Complex, який містить в собі 5 полів:

- Реальна компонента
- Уявна компонента
- Частота
- Амплітуда
- Фаза

Цей клас містить конструктори для різних підібраних параметрів та різної їх кількості. Також містить методи для дій над полями:

- Додавання двох комплексних чисел
- Віднімання
- Приведення комплексного числа у степінь
- Множення комплексних чисел
- Копіювання комплексних чисел
- Ділення двох комплексних чисел
- Нормалізація комплексного числа
- Задання амплітуди
- Розрахунок амплітуди в квадраті
- Поворот відносно свого початку

Також є метод DFT, який приймає в себе колекцію ArrayList з комплексних чисел та виконує перетворення і повертає також колекцію ArrayList з комплексних чисел, але вже з впорядкованими даними частот, амплітуд, та компонент.

За допомогою інтерфейсу користувача екран персонального комп'ютера перетворюється у полотно, яке відповідає коміркам автомати, яка оновлюється 30 разів на секунду та показує наглядну симуляцію автомати.

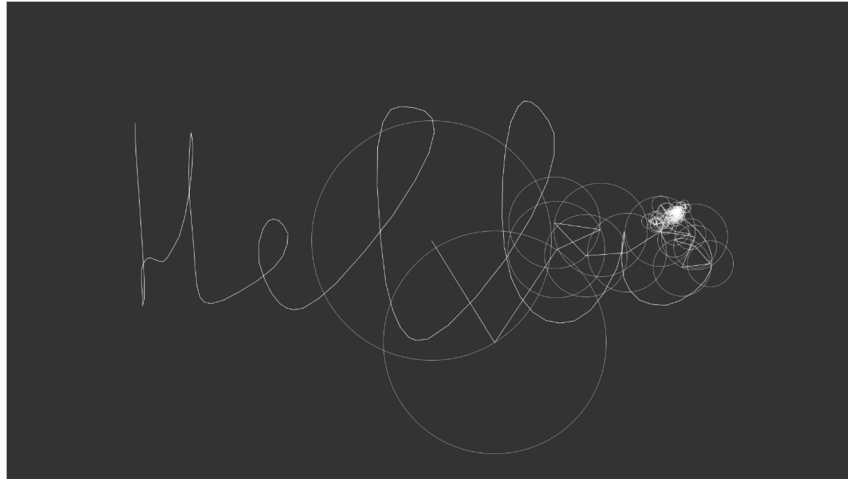


Рис. 2. Відображення процесу малювання

Клітинний автомат в загальному вигляді є впорядкованою множиною кінцевих автоматів, які обмінюються інформацією з сусідніми автоматами. У клітинному автоматі фон Неймана комірки впорядковані у вигляді двомірної прямокутної решітки і взаємодіють з чотирма прилеглими комірками, що утворюють окіл фон Неймана. Решітка має нескінченний розмір в обох напрямках, а комірки - ідентичними в плані правил переходу. Зміна станів всіх осередків відбувається синхронно.

На початку роботи програми користувач може вільно рухати мишкою та затискаючи ліву кнопку миші малює шлях, який має бути перетворений за допомогою DFT.

Спершу перед тим, як запускати програму, треба визначитись, яку версію використовувати. Processing експортує застосунок для 2 версій: x32(x86) та x64.

Кожна папка містить файл .exe, папки з бібліотеками та іншими файлами, які необхідні для правильного запуску.

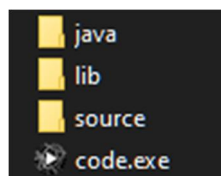


Рис. 3. Відображення папки з портативною версією програми

На рис. 4. показано результат застосування програми для малювання тексту на віртуальній дошці.

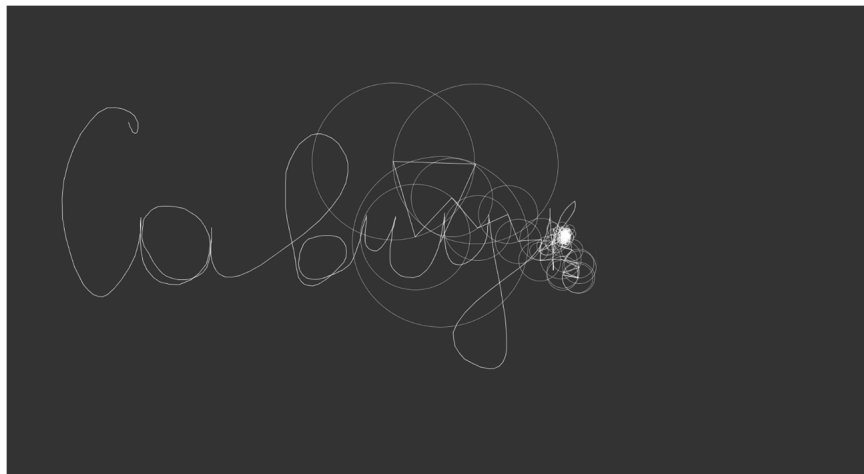


Рис. 4. Відображення результату виконання

Таким чином, ми бачимо відображення створення графічного зображення із використанням дискретного перетворення Фур'є.

5. Висновок

У роботі досліджено застосування дискретного перетворення Фур'є для роботи з графічними редакторами для яких є можливість передачі даних за допомогою цифрових алгоритмів обробки сигналів.

Результати дослідження є важливими, оскільки, під час пандемії великої популярності набрали сервіси для онлайн-конференції, лекцій, зустрічей тощо. Важливим функціоналом для навчального процесу є віртуальна дошка, на якій викладач, будує певний графічний об'єкт в двовимірній площині, далі завдяки розробленій системі, що використовує дискретне перетворення Фур'є, зображення перетворюється у сигнал. Далі зображення без втрат із високою швидкістю передачі та розгортки потрапляє на екран користувача. Розроблений алгоритм може значно покращити роботу викладачів із онлайн-дошками, оскільки мінімізує затримки при передачі зображень, а тому ми отримуємо мінімальну затримку між словами і діями викладача на дошці. Запропонований алгоритм можна також застосувати для передавання і звукових сигналів. Це дозволить використовуючи тандем перетворень Фур'є, як для звуку так і для відео, відправляти все єдиним пакетом, а отже при розгортці на персональному комп'ютері користувача затримка між діями викладача та словами, які він використовує для пояснення мінімізується до значень незначної похибки.

Отримані результати засвідчують, що розроблена система працює ефективно та надійно. Для виконання дій з користувацьким інтерфейсом у системі передбачено мінімальну кількість дій.

Список літератури

1. Глинченко, А. С. *Цифровая обработка сигналов*. – М.: ИПК СФУ, 2008. - 242с.
2. Lynn, Paul A., Fuerst, Wolfgang, *Introductory Digital Signal Processing with Computer Applications*, John Wiley & Sons, 1994.
3. R. Meyer, H.W. Schuessler, and K. Schwarz. (1990). *FFT Implmentation on DSP chips – Theory and Practice*. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
4. H.V. Sorensen and C.S. Burrus. (1993, March). *Efficient computation of the DFT with only a subset of input or output points*. *IEEE Transactions on Signal Processing*, 41(3), 1184-1200.
5. Joyce Van de Vegte, *Fundamentals of Digital Signal Processing*, Prentice Hall, 2002
6. Brigham, E. Oran (1988). *The fast Fourier transform and its applications*. Englewood Cliffs, N.J.: Prentice Hall
7. Cormen, Thomas H.; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2001). «Chapter 30: Polynomials and the FFT». *Introduction to Algorithms (Second ed.)*. MIT Press and McGraw-Hill.
8. Bradley W. Dickinson; Kenneth Steiglitz (1982). «Eigenvectors and functions of the discrete Fourier transform»
9. Shakib, J., Muqri, M., *Leveraging the Power of Java in the Enterprise*, American Society for Engineering Education, AC 2010-1701.
10. Dibble, P., *Real-Time Java Platform Programming*, Sun Microsystems Press, Prentice-Hall, June 2008.
11. Кей С. Хорстманн. *Java SE 9. Базовый курс*. К.; Вильямс, 2018.-576 с.
12. Palmer G., *Technical Java - Developing Scientific and Engineering Applications*, Prentice Hall, 2003.

References

1. Glinchenko, A. S. *Cifrovaya obrabotka signalov*. – М.: IPK SFU, 2008. - 242s.
2. Lynn, Paul A., Fuerst, Wolfgang, *Introductory Digital Signal Processing with Computer Applications*, John Wiley & Sons, 1994.
3. R. Meyer, H.W. Schuessler, and K. Schwarz. (1990). *FFT Implmentation on DSP chips – Theory and Practice*. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
4. H.V. Sorensen and C.S. Burrus. (1993, March). *Efficient computation of the DFT with only a subset of input or output points*. *IEEE Transactions on Signal Processing*, 41(3), 1184-1200.

5. Joyce Van de Vegte, *Fundamentals of Digital Signal Processing*, Prentice Hall, 2002
6. Brigham, E. Oran (1988). *The fast Fourier transform and its applications*. Englewood Cliffs, N.J.: Prentice Hall
7. Cormen, Thomas H.; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2001). «Chapter 30: Polynomials and the FFT». *Introduction to Algorithms (Second ed.)*. MIT Press and McGraw-Hill.
8. Bradley W. Dickinson; Kenneth Steiglitz (1982). «Eigenvectors and functions of the discrete Fourier transform»
9. Shakib, J., Muqri, M., *Leveraging the Power of Java in the Enterprise*, American Society for Engineering Education, AC 2010-1701.
10. Dibble, P., *Real-Time Java Platform Programming*, Sun Microsystems Press, Prentice-Hall, June 2008.
11. Kej S. Horstmann. *Java SE 9. Bазовуј курс. К.*; Vilyams, 2018.-576 s.
12. Palmer G., *Technical Java - Developing Scientific and Engineering Applications*, Prentice Hall, 2003.