

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

Дипломна робота

на здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

**ЗАСТОСУВАННЯ ГЕОМЕТРИЧНИХ СТРУКТУР ТА ІНСТРУМЕНТІВ
ОБЧИСЛЮВАЛЬНОЇ ГЕОМЕТРІЇ В КОМП'ЮТЕРНОМУ МИСТЕЦТВІ**

Виконав студент 4 курсу

Кочетков Денис Олександрович

(підпис)

Науковий керівник:

професор

Терещенко Василь Миколайович

(підпис)

Засвідчую, що в цій курсовій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

РЕФЕРАТ

Робота складається зі вступу, 3 розділів, висновків, списку використаних джерел (52 найменування). Робота містить 21 рисунок. Загальний обсяг становить 39 сторінок, основний текст роботи викладено на 25 сторінках.

Ключові слова: БІНАРНЕ ДЕРЕВО, ДІАГРАМА ВОРОНОГО, ЗАСТОСУВАННЯ, КОМП'ЮТЕРНЕ МИСТЕЦТВО, КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ, ОБЧИСЛЮВАЛЬНА ГЕОМЕТРІЯ, СЕРЕДОВИЩЕ РОЗРОБКИ, ТРІАНГУЛЯЦІЯ ДЕЛОНЕ.

Об'єкт роботи: способи застосування геометричних структур та інструментів обчислювальної геометрії в комп'ютерному мистецтві.

Мета роботи: ознайомлення аудиторії з особливостями застосування інструментів обчислювальної геометрії в комп'ютерному мистецтві з подальшим знаходженням шляхів просування кооперації, оскільки розвиток галузі є неможливим без залучення спеціалістів з області комп'ютерних наук; запропонувати власну реалізацію методів обчислювальної геометрії у контексті комп'ютерного мистецтва; зафіксувати свої спостереження щодо особливих топологічних властивостей тих геометричних структур, які виявляються корисними у використанні в комп'ютерному мистецтві.

Методи дослідження: історичний метод; аналіз літератури; конкретизація та систематизація існуючих засобів застосування геометричних структур та інструментів обчислювальної геометрії у комп'ютерному мистецтві, створення власних програм, що їх реалізують; аналіз існуючих середовищ розробки.

Інструменти: безкоштовне середовище розробки та мова програмування з відкритим кодом Processing, комерційне інтегроване середовище розробки JetBrains IntelliJ IDEA для мови програмування Java, бібліотеки для мови програмування Processing з відкритим кодом controlP5 та quickhull3d.

Результати роботи: проаналізоване застосування структур обчислювальної геометрії в сучасному комп'ютерному мистецтві. Розглянуті

прикладі використання таких структур як випадкові бінарні дерева, дерева квадрантів та дерева октантів, k-d дерева, триангуляція Делоне та діаграма Вороного в контексті комп'ютерного мистецтва — розглянуті їхні властивості та візуальні характеристики.

Запропонований новий метод використання динамічної діаграми Вороного в комп'ютерному мистецтві та створений програмний засіб, що реалізує запропонований підхід.

ЗМІСТ

РЕФЕРАТ	2
СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП.....	7
РОЗДІЛ 1. ІСТОРІЯ ТА РОЗВИТОК КОМП'ЮТЕРНОГО МИСТЕЦТВА.....	9
1.1 Початок розвитку	9
1.2 Перші виставки комп'ютерного мистецтва.....	11
1.3 Подальший розвиток.....	13
1.4 Питання пов'язані з комп'ютерним мистецтвом.....	14
РОЗДІЛ 2. СТРУКТУРИ ОБЧИСЛЮВАЛЬНОЇ ГЕОМЕТРІЇ В КОМП'ЮТЕРНОМУ МИСТЕЦТВІ.....	16
2.1 Дерева	16
2.1.1 Випадкове бінарне дерево.....	16
2.1.2 Дерево квадрантів	17
2.1.3 Дерево октантів	19
2.1.3 К-вимірне дерево	19
2.2 Триангуляція Делоне	20
2.3 Діаграма Вороного	21
2.3.1 Властивості ДВ для комп'ютерного мистецтва.....	22
2.3.2 Використання ДВ у відео	24
2.3.3 Декоративний дизайн	26
2.3.4 Фрактали Вороного.....	27
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	29
3.1 Огляд існуючих рішень	29

	5
3.1.1 openFrameworks	29
3.1.2 Cinder	29
3.1.3 Processing	30
3.2 Практична реалізація	31
3.2.1 Клітинна структура	31
3.2.2 Аналіз звукового сигналу.....	32
ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	35

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ДВ — діаграма Вороного;

ТД — триангуляція Делоне;

GUI — Grafical User Interface;

ПЗ — програмне забезпечення;

IDE — Integrated Development Environment;

FFT — Fast Fourier Transform.

ВСТУП

З розвитком комп'ютерних технологій, їхнє використання почало поширюватися на різні галузі поза межами комп'ютерних наук і досягло в цьому визначного успіху. На сьогоднішній день комп'ютерне мистецтво є надзвичайно важливим елементом сучасного дизайну, архітектури, музики, тощо. Проте питання про можливості використання комп'ютерних наук у мистецтві залишається практично недослідженим.

Комп'ютерне мистецтво — будь-яке мистецтво, в якому комп'ютери відіграють певну роль при виробленні або показі витвору мистецтва.

Існує багато статей стосовно використання різних конкретних комп'ютерних технологій у мистецтві, але я не зустрів жодної статті, яка б стосувалася загального огляду цієї тематики. Я вирішив систематизувати весь матеріал, що стосується комп'ютерного мистецтва, зокрема використання структур обчислювальної геометрії у комп'ютерному мистецтві, адже я вважаю, що в них є дуже великий естетичний потенціал. Це і визначає мотивацію моєї роботи.

Актуальність роботи полягає в першу чергу в знаходженні нових засобів застосування комп'ютерних технологій у мистецтві, а отже і збільшенню різноманіття витворів мистецтва. Комп'ютерні технології можуть надавати митцям більше впевненості, спрощувати складні технічні процеси, наділяти новими формами художнього мислення і втілення. Розвиток мистецтва зі свого боку є вкрай важливим через значну інтегрованість мистецтва у повсякденне життя.

Метою цієї роботи є в першу чергу ознайомлення аудиторії з особливостями застосування структур та інструментів обчислювальної геометрії в комп'ютерному мистецтві. Це сприяє подальшому розвитку галузі через залучення спеціалістів з області комп'ютерних наук.

Завдання цієї роботи — створення власного програмного засобу з використанням методів обчислювальної геометрії у контексті комп'ютерного мистецтва; фіксування своїх спостережень щодо особливих топологічних властивостей тих геометричних структур, які виявляються корисними у використанні в комп'ютерному мистецтві.

Існує багато різних понять пов'язаних з комп'ютерним мистецтвом. Термін «генеративне мистецтво» (від англ. — *generative art*) описує мистецтво створене повністю або частково за допомогою автономної системи. На сьогодні цей термін зазвичай пов'язують саме з алгоритмічним мистецтвом — тим, що чітко задане алгоритмом та згенероване комп'ютером, але в той же час генеративне мистецтво може бути створене використовуючи хімічні, біологічні, механічні системи, тощо.

Маргарет Боден та Ернест Едмондс в своїй статті 2009 року [1] про генеративне мистецтво стверджують, що з самого початку розвитку комп'ютерного мистецтва поняття «генеративне мистецтво» та «комп'ютерне мистецтво» стоять поруч та заміняють одне одного. Тому надалі в цій роботі я буду використовувати термін «комп'ютерне мистецтво», маючи на увазі як саме його, так і генеративне та алгоритмічне мистецтво.

РОЗДІЛ 1. ІСТОРІЯ ТА РОЗВИТОК КОМП'ЮТЕРНОГО МИСТЕЦТВА

1.1 Початок розвитку

Перший розвиток мистецтва пов'язаного з комп'ютерами почався в 1956 році. Саме тоді було згенероване перше зображення людини на екрані комп'ютера – це було зображення пін-ап дівчини (див. Рисунок 1) на воєнному комп'ютері SAGE, який коштував 238 мільйонів доларів та був розроблений для запобігання ядерних атак СРСР [2].

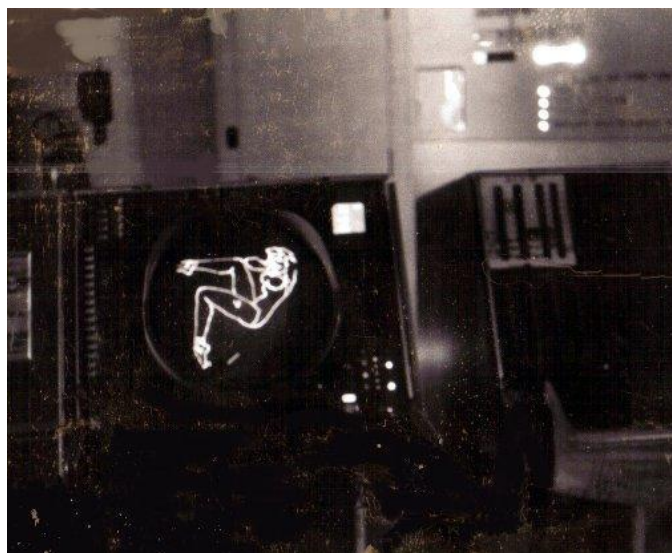


Рисунок 1 – Перше зображення людини на екрані комп'ютера

У 1960 році Десмонд Пол Генрі розробив машину для малювання — повністю механічна, вона не була схожа на комп'ютери того часу, оскільки не могла зберігати інформацію та бути запрограмована (див. Рисунок 2) [3]. Генрі розробляв машини навмисно неточно, щоб кожна похибка в роботі (наприклад, ослаблений гвинт) могла вплинути на результат роботи, а отже і на результуючий малюнок.

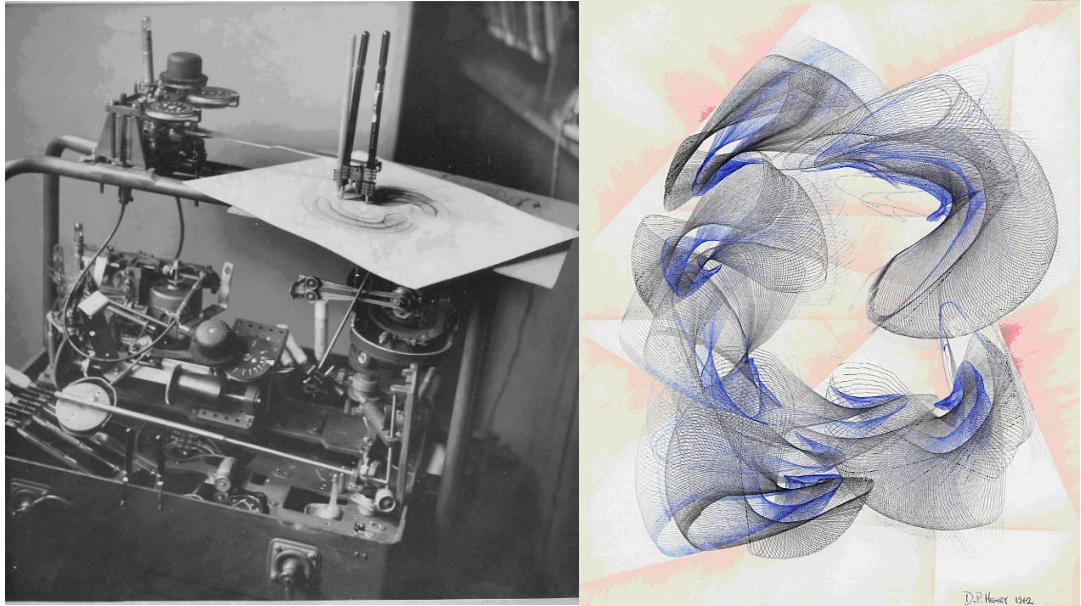


Рисунок 2 – Перша машина для малювання

До середини 1960-х років комп'ютерним мистецтвом могли займатися лише науковці та інженери, адже тільки вони мали доступ до комп'ютерів, що в той час знаходилися лише в дослідницьких лабораторіях при університетах. В 1962 році в компанії Bell Telephone Laboratories А. Майкл Нолл вперше запрограмував комп'ютер для генерації візуальних патернів [4]. Він використовував цифровий комп'ютер щоб створювати артистичні візерунки та формалізував використання алгоритмічних та випадкових процесів в створенні візуального мистецтва. Нолл використовував ранню алгоритмічну мову програмування FORTRAN для написання всіх програм.

Технології того часу ще не дозволяли генерувати малюнки безпосередньо на екрані комп'ютера. Для цього використовувалися плотери (або графопобудовники). Дані для плотера були згенеровані на комп'ютері та малювалися плотером безпосередньо на папірі.

Такий підхід дозволяв використовувати векторну графіку, тому за допомогою плотерів зазвичай малювалися об'єкти з великою кількістю ліній та кривих.

1.2 Перші виставки комп'ютерного мистецтва

У 1965 році починають проводитися виставки комп'ютерного мистецтва. В лютому в Штутгарті, Німеччина, Георгом Нісом була проведена перша у світі виставка комп'ютерного мистецтва “computer grafik” (Рисунок 3) [5]. Обчислення виконувались на комп'ютері компанії Siemens, вивід з комп'ютера подавався на автоматичну малювальну машину Zuse Graphomat Z64.

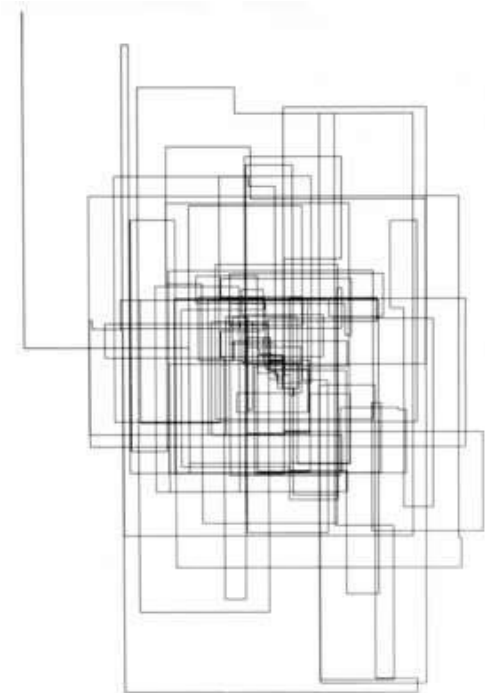


Рисунок 3 – Георг Ніс - 23-Еске (зліва), achsenparalleler irrweg (2) (справа)

Пізніше, в квітні 1965 року, в Нью-Йорку була проведена виставка “Computer-Generated Pictures” А. Майкла Нолла та Бели Жулеша (Рисунок 4) [4]. Це була перша подібна виставка в Сполучених Штатах. Хоча роботи, що були представлені на виставці, не продавалися і висвітлення пресою майже не було, Стюарт Престон з New York Times висловився про виставку з великим ентузіазмом, зазначивши, що прийде час, коли «фактичний дотик артиста більше не буде брати ніякої участі у створенні витвору мистецтва» [6].

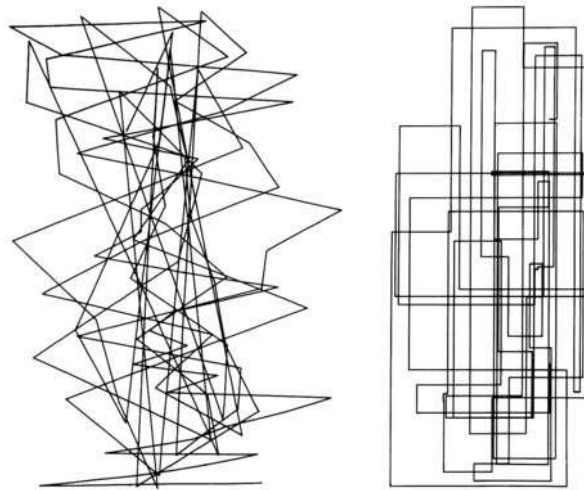


Рисунок 4 – А. Майкл Нолл - Gaussian-Quadratic (зліва), Vertical-Horizontal No. 3 (справа)

Третя виставка, “Computer-Grafik”, була проведена в Штутгарті в листопаді 1965 року та представляла роботи Фрідера Наке та Георга Ніса [7]. Одна з цих робіт - 13/9/65 Nr. 2 (“Hommage à Paul Klee”) Фрідера Наке (див. Рисунок 5) найбільш зустрічається в книгах та статтях про ранній період комп’ютерного мистецтва. Було зроблено близько 40 копій картини, кожна з яких може вважатися оригіналом. При генерації зображення Наке надихався масляною роботою Пауля Клее “Hauptweg und Nebenwege” (1929 р.) та роботами Клее з того періоду.

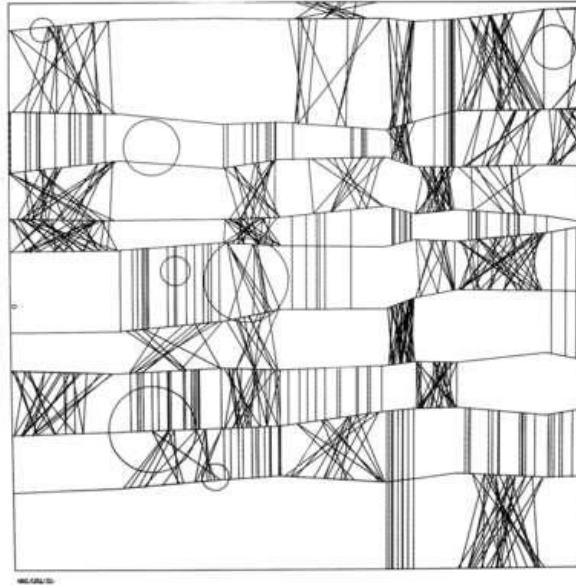


Рисунок 5 - Фрідер Наке - 13/9/65 Nr. 2 ("Hommage à Paul Klee")

Всі три виставки комп'ютерного мистецтва 1965 року майже не отримали реакції преси: вона була обмежена лише місцевими або регіональними газетами.

1.3 Подальший розвиток

У 1968 році Інститут сучасного мистецтва в Лондоні провів наймасштабнішу на той час виставку з комп'ютерного мистецтва «Кібернетична випадковість» (англ. "Cybernetic Serendipity"). На ній були представлені роботи дуже багатьох комп'ютерних митців того часу: Фрідера Наке, Георга Ніса, А. Майкла Нолла, Нам Джуна Пека, Джона Уїтні, Чарльза Чурі та інших. Виставка була поділена на частини: відділ, присвячений алгоритмам для генерації музики, відділ комп'ютерної графіки, відділ з комп'ютерами-генераторами віршів та есе.

Через рік після цієї виставки в Лондоні було засновано Товариство комп'ютерних мистецтв.

У 1970 році Кетрін Неш і Річард Х. Вільямс випускають статтю «Комп'ютерне програмування для артистів» [8], в якій описується, як митець може використовувати комп'ютер в своїх творчих цілях.

У 70-х роках починають з'являтися перші графічні інтерфейси користувачів (англ. GUI) — Xerox Star 8010.

В 1984 році виходить перший комп'ютер компанії Apple — Macintosh — що дозволяє графічним дизайнерам швидко навчитися робити витвори мистецтва безпосередньо на екрані комп'ютера.

1.4 Питання пов'язані з комп'ютерним мистецтвом

Розвиток комп'ютерного мистецтва характеризується наявністю теоретичних та філософських питань стосовно формалізації мистецтва та можливостей комп'ютерів створювати речі, які можуть називатися мистецтвом. Джон МакКормак в своїй статті [9] називає наступні десять питань найважливішими для розуміння та розвитку комп'ютерного генеративного мистецтва:

1. Чи може машина щось створити?
2. Як це – бути комп'ютером, який займається мистецтвом?
3. Чи можна формалізувати людську естетику?
4. Які нові види мистецтва дозволяє робити комп'ютер?
5. У якому сенсі генеративне мистецтво є репрезентативним і що воно представляє?
6. Яка роль випадковості в генеративному мистецтві?
7. Що може обчислювальне генеративне мистецтво сказати нам про творчість?
8. Що характеризує хороше генеративне мистецтво?
9. Що ми можемо дізнатись про мистецтво з генеративного мистецтва?
10. Які майбутні події змусять нас переглянути свої відповіді?

Згадані питання часто слугують мотивацією для розвитку галузі, пошуку істотно нових засобів застосування комп'ютерних технологій. Нові підходи

мають допомогти краще зрозуміти суть комп'ютерного мистецтва і можливостей комп'ютера у мистецтві.

РОЗДІЛ 2. СТРУКТУРИ ОБЧИСЛЮВАЛЬНОЇ ГЕОМЕТРІЇ В КОМП'ЮТЕРНОМУ МИСТЕЦТВІ

2.1 Дерева

2.1.1 Випадкове бінарне дерево

Випадкове бінарне дерево (від англ. — *random binary tree*) — це бінарне дерево вибране випадково згідно з деяким імовірнісним розподілом на бінарних деревах.

Є різні підходи до генерації випадкових бінарних дерев: формування звичайного бінарного дерева з випадкової перестановки чисел, вибір дерева з дискретного рівномірного розподілу, в якому всі різні дерева однаково ймовірні, або формування дерева з випадково розділеного на дві частини набору чисел.

Рівномірні випадкові двійкові дерева мають ймовірність обернено пропорційну до загальної кількості дерев з n вершинами, тобто n -того числа Каталана. В такій моделі очікувана довжина дерева буде пропорційна не логарифму від кількості вершин, як в звичайному бінарному дереві (при умові балансування), а квадратному кореню [10]. Такі дерева не використовуються в якості бінарних дерев пошуку через велику середню висоту, але знайшли застосування в якості синтаксичних дерев для алгебраїчних виразів в дизайні компіляторів [11] та для моделювання еволюційних дерев [12].

Формування дерев з випадковим розділенням набору чисел знайшло своє застосування в комп'ютерному мистецтві. Суть генерації такого дерева на n вершинах полягає в тому, що спочатку генерується дійсна випадкова змінна $x \in (0,1)$ і xn перших чисел приписуються до лівого піддерева, наступне число y — корінь, інші — у праве піддерево [13].

Якщо x розподілений рівномірно, то результат буде таким же, як і в генерації дерева на випадковій перестановці чисел. Але якщо використовувати

бета-розподіл, то візуалізація таких дерев нагадує реалістичні ботанічні дерева (Рисунок 6) [13].

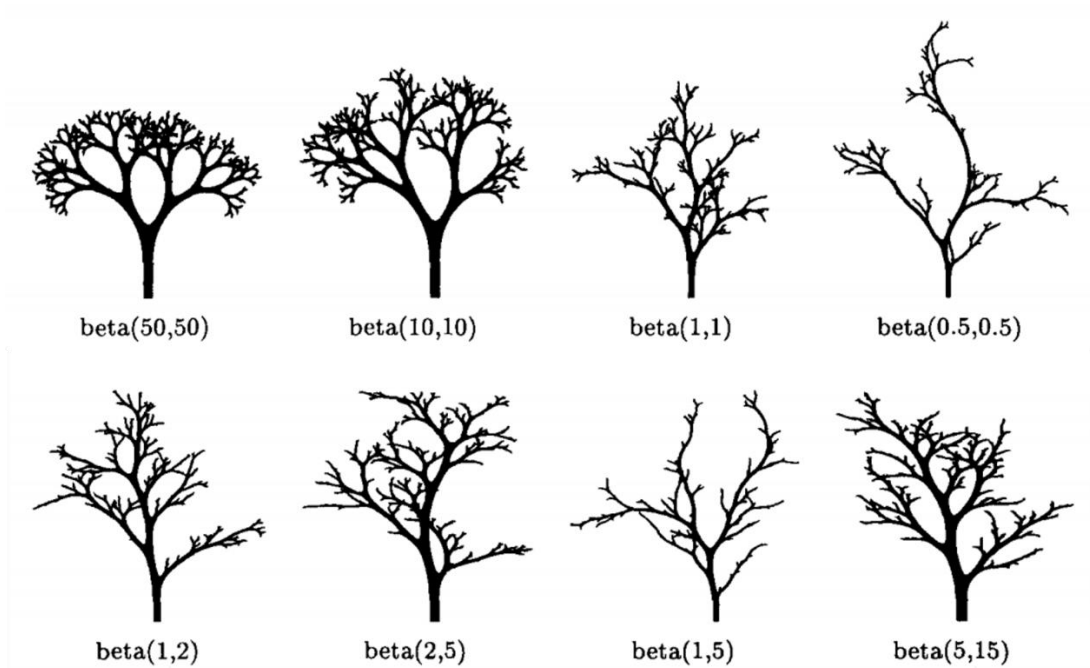


Рисунок 6 - Симуляції ботанічних дерев використовуючи бета-розподіл випадкових бінарних дерев

2.1.2 Дерево квадрантів

Дерево квадрантів — це дерево, в якому кожна вершина має рівно чотири нащадки. Ця структура даних використовується для рекурсивного розбиття двовимірного простору на 4 квадранти (області). Вперше ця структура зустрічається в роботі Рафаеля Фінкеля та Джона Бентлі в 1974 році [14].

Дерево квадрантів використовується в представленні зображень, просторових базах даних [15], у виявленні зіткнень у двовимірному просторі [16] та навіть симуляції гри Життя [17].

Елементами дерева квадрантів можуть бути як точки, так і області, лінії (прямі або криві), багатокутники. У візуальному мистецтві частіше за все використовується саме розбиття на регіони – ці регіони можуть бути не тільки квадратами або прямокутниками, але й мати довільну форму.

Найкращі приклади використання дерева квадрантів у візуальному мистецтві — проекти Равена Квока 189D0 [18] (для музичного кліпу Karma Fields – Sweat, див. Рисунок 7), Stickup [19] (для музичного кліпу Karma Fields & MORTEN – Stickup) та 1DDCB [20]. В цих проектах кожний рівень дерева квадрантів ділиться за певним заданим шаблоном. В проекті 1DDCB вершини кожної клітини в дереві квадрантів є генераторами діаграми Вороного (див. Рисунок 8).

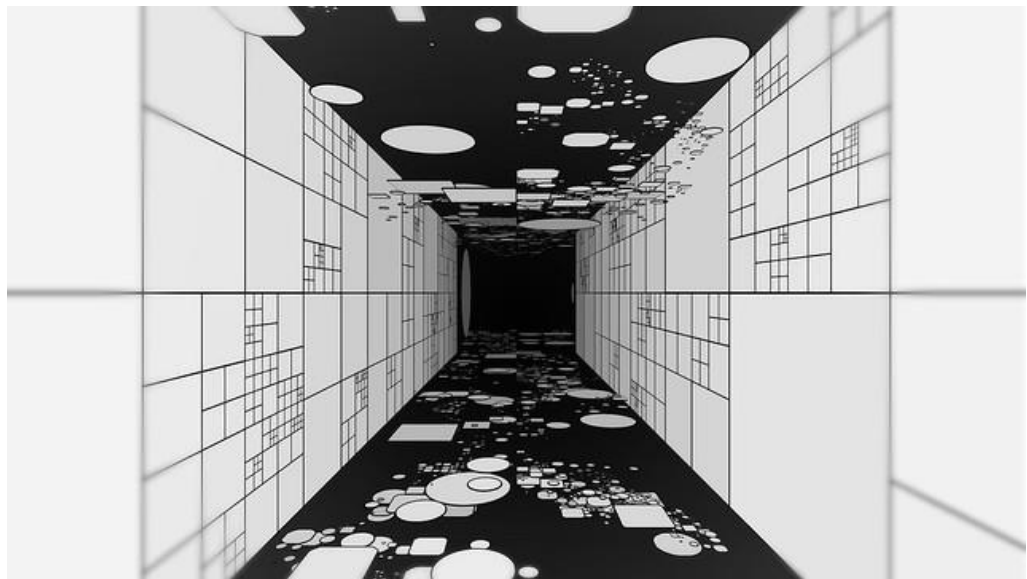


Рисунок 7 – Raven Kwok - 189D0

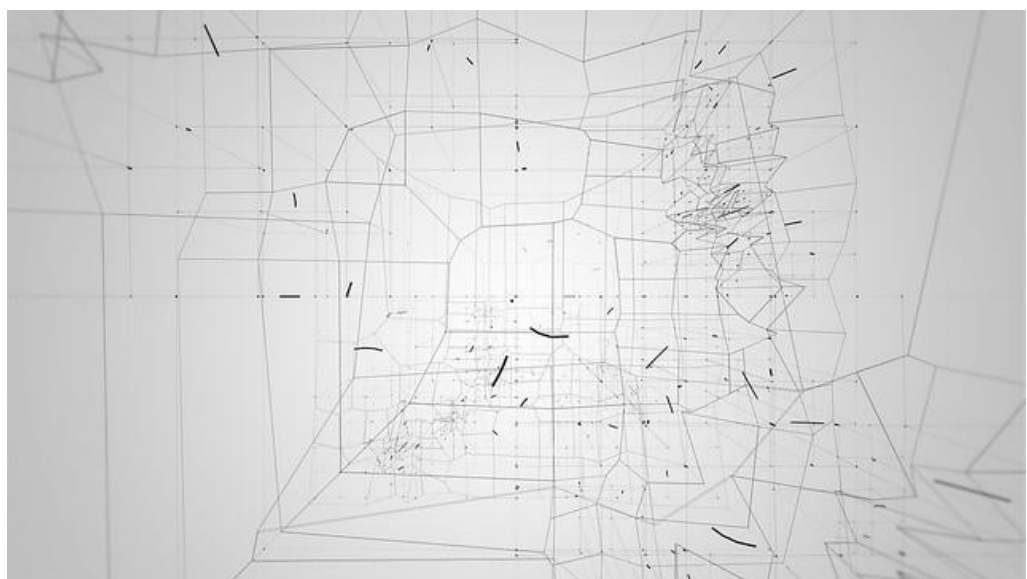


Рисунок 8 – Raven Kwok - 1DDCB

2.1.3 Дерево октантів

Дерево октантів — це структура даних, в якій кожна вершина дерева має рівно вісім нащадків. Дерево октантів – тривимірний аналог дерева квадрантів. Ця структура була вперше запропонована Дональдом Мігером в 1980 році для роботи з тривимірною комп’ютерною графікою [21].

Дерево октантів використовується для рекурсивного розбиття тривимірного простору на октанти. В обчислювальній геометрії цей підхід застосовується в ефективному виявленні зіткнень в тривимірному просторі, в пошуку найближчих сусідів [22].

В мистецтві дерево октантів, як і дерево квадрантів, зустрічається для візуального розбиття простору на елементи. Тільки тут, на відміну від дерева квадрантів, ми вже працюємо в тривимірному просторі. Кожний октант може бути основою для побудови більш складних структур всередині нього (наприклад, криві Безьє [23]).

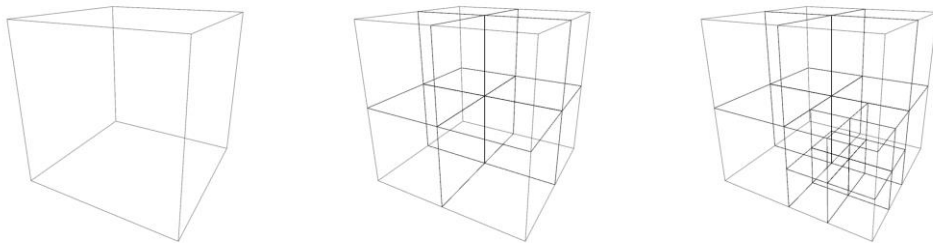


Рисунок 9 – Розбиття простору деревом октантів

2.1.3 К-вимірне дерево

К-вимірні дерева (k-d дерева, де k — вимірність пошукового простору) були запропоновані в 1975 році Джоном Бентлі як структура даних для асоційованого пошуку в багатовимірних просторах [24].

К-вимірні дерева — особливий вид двійкових пошукових дерев, де кожний листок дерева — це точка в k-вимірному просторі.

K-d дерева зазвичай використовуються в задачах пошуку найближчого сусіда [25] та інших пошукових алгоритмах, наприклад в базах даних [26]. Також ця структура даних використовується в інтерактивному відстеженні рис обличчя у відеокадрах [27], трасуванні променів [28] та навіть в розробці планувань планів поверхів з генетичними алгоритмами [29].

Візуалізація k-d дерев зустрічається в роботах деяких комп'ютерних митців таких як Маркус Вольц [30], Метт ДеЛор'є (див. Рисунок 10) та Равен Квок (Autotroph [31] та 1DE94 [32]).

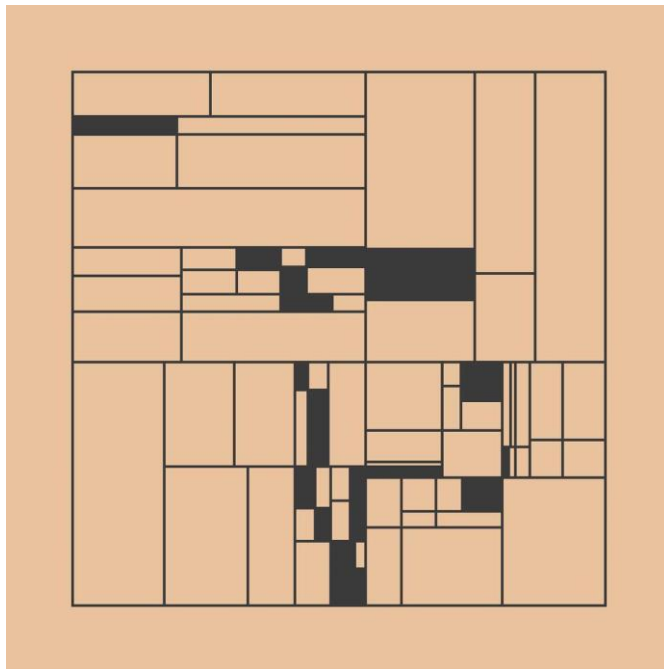


Рисунок 10 – Метт ДеЛор'є – k-d tree structure

2.2 Тріангуляція Делоне

Тріангуляція Делоне — це такий вид тріангуляції множини точок, що для кожного трикутника всі точки множини окрім вершин трикутника лежать зовні кола, описаного навколо трикутника. Вперше ця структура була описана Борисом Делоне в 1934 році [33].

Тріангуляція Делоне зустрічається в комп'ютерному мистецтві не тільки як самостійна структура (Рисунок 11), але й використовується для тріангуляцій зображень по точках.

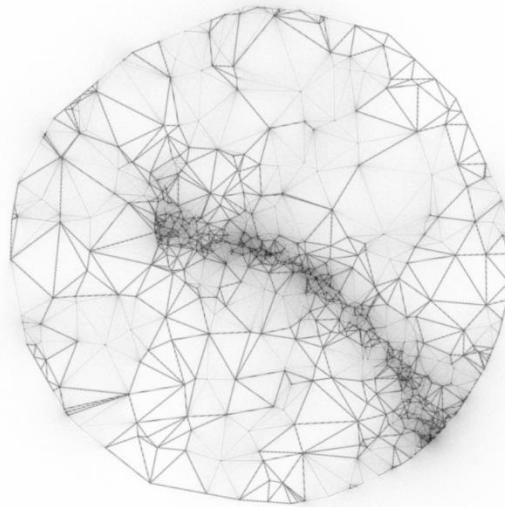


Рисунок 11 - Маркос Монтане - structure26

Триангуляцію Делоне також можна розглядати в контексті відео, тобто для побудови триангуляції на рухомих точках. В такому випадку триангуляція буде змінювати свою топологічну, а отже і візуальну структуру в моменти так званих топологічних подій: конциклічності чотирьох точок або колінеарності трьох точок на межі опуклої оболонки множини точок.

В моменти топологічних подій відбувається «фліп» ребра триангуляції Делоне – дуже різка зміна в візуальній структурі триангуляції, тому для відео частіше використовується діаграма Вороного (пункт 2.3.2).

2.3 Діаграма Вороного

Діаграма Вороного (теселяція Вороного) — це розбиття метричного простору (в нашому випадку найчастіше за все двовимірного евклідового) таким чином, що кожній точці p з множини точок P належить комірка Вороного $VR(p, P)$, яка утворена зі всіх точок простору, для яких p — найближча точка. Діаграма Вороного $VD(P)$ – це множина комірок всіх точок (див. Рисунок 12).

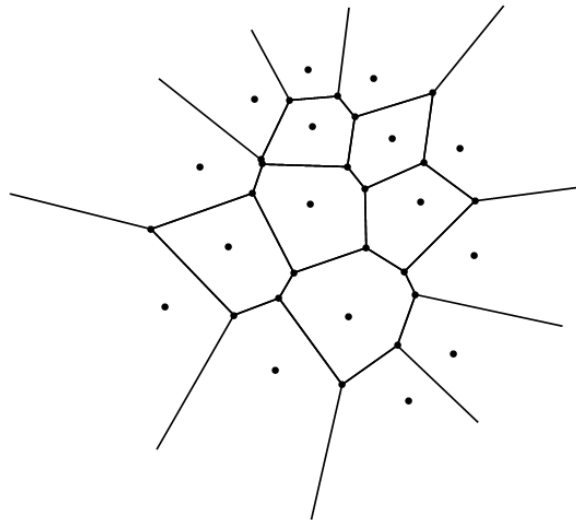


Рисунок 12 – Діаграма Вороного

Ця структура зустрічається з часів Рене Декарта, коли він описуючи сонячну систему, ілюстрував вихори як розбиття площини на опуклі регіони, що дуже схоже на діаграму Вороного. Формально ж ця структура була описана математиками Георгієм Вороним [34] та Йоганном Діріхле [35] для дослідження квадратичних форм.

2.3.1 Властивості ДВ для комп'ютерного мистецтва

Окрім загальних властивостей діаграми Вороного як структури в обчислювальній геометрії, вона має ще дві дуже важливих властивості для застосування в комп'ютерному мистецтві.

По-перше, діаграма Вороного деякої множини точок залежить тільки від відносного розташування точок, а не від їх абсолютної позиції в просторі чи орієнтації. Іншими словами, якщо зсунути або повернути всі точки разом, топологічна структура ДВ не зміниться.

Таким чином, розглядаючи симетрію точок множини, можна побачити, що діаграма Вороного симетрії точок це те ж саме що й симетрія самої діаграми Вороного [36].

По-друге, якщо розглядати діаграму Вороного на рухомих точках, то можна побачити, що невеликі зміни в позиції генераторів ДВ спричиняють

невеликі зміни в структурі самої діаграми, тобто невеликі зміни у формах комірок Вороного. Неформально можна сказати, що трансформація ДВ є неперервною функцією над множиною генераторів, адже невеликі зміни генераторів викликають невеликі зміни в діаграмі Вороного. Ця властивість завжди виконується в просторі \mathbb{R}^d , що було доведено Даніелем Рімом в статті [37].

Цей факт очевидний при візуальному порівнянні топологічних змін діаграми Вороного і триангуляції Делоне (див. Рисунок 13). Хоча ці структури є дуальними графами один одного, топологічні зміни виглядають в них зовсім по-різному.

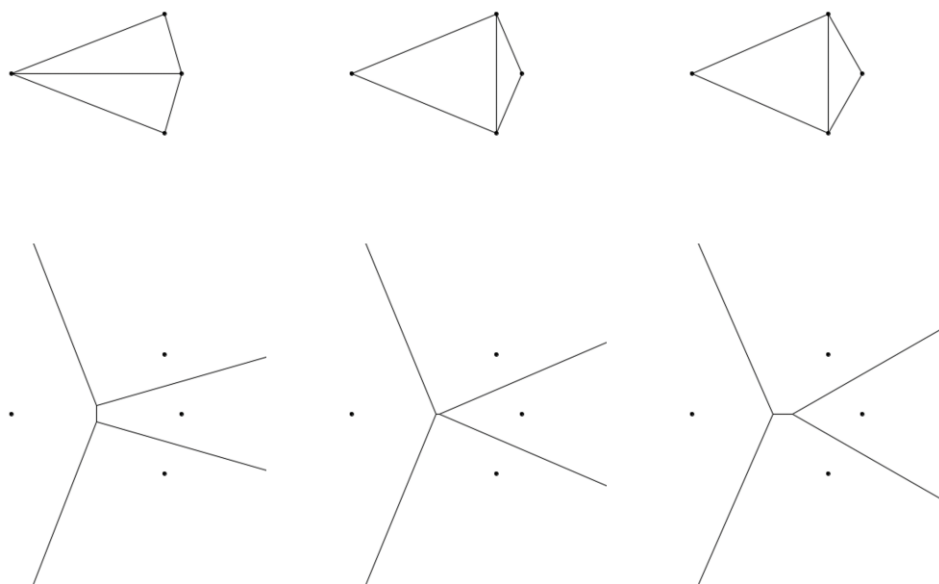


Рисунок 13 – Порівняння топологічних змін триангуляції Делоне та діаграми Вороного

Ця неперервність — дуже хороша властивість для розробки анімацій з діаграмою Вороного, адже вона забезпечує плавну трансформацію зображення. Якщо з'являється нова комірка Вороного, вона з'являється з точки і росте. Перед тим як зникнути, комірка Вороного перетворюється у точку.

2.3.2 Використання ДВ у відео

Діаграми Вороного дуже часто зустрічаються в візуальному комп'ютерному мистецтві. Вони використовуються для того, щоб отримати естетичну форму зображення, розбивши його на множину точок та побудувавши на них діаграму Вороного.

Неперервність динамічної діаграми Вороного, як зазначено вище, дозволяє робити плавні анімації зображень.

Серед прикладів використання діаграми Вороного в комп'ютерному мистецтві слід обов'язково розглянути проекти Равена Квока Skyline [38] та Greatness [39].

В проєкті Skyline (для музичного кліпу Karma Fields — Skyline) діаграма Вороного відіграє основну роль у побудові візуальної структури кліпу. Генератори діаграми розділені на типи, що мають різну поведінку та різні алгоритми трансформації. Генератори рухаються за наперед заданими математичними функціями та згідно з анімованим зображенням вокалістки (Рисунок 14) [38].

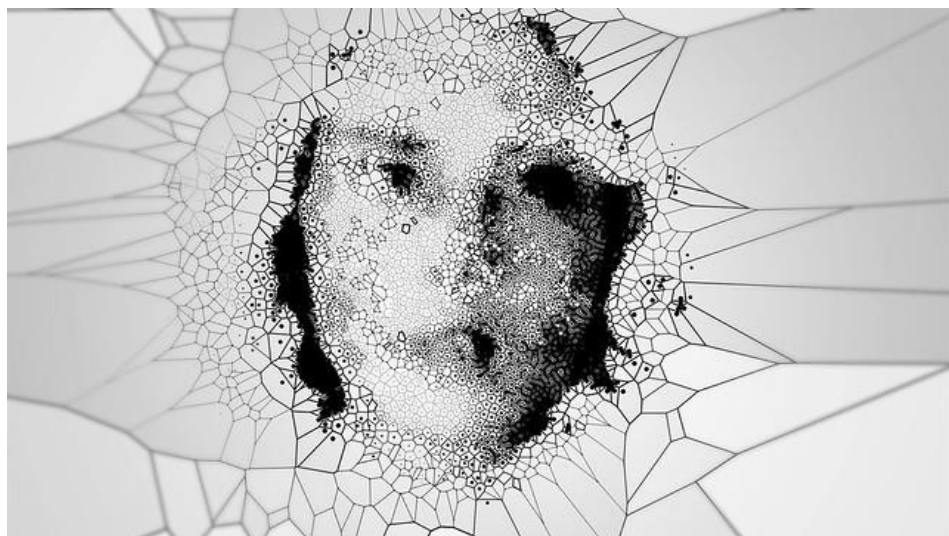


Рисунок 14

В проєкті Greatness (для музичного кліпу Karma Fields feat. Talib Kweli — Greatness) використовується схожа візуальна тенденція: діаграма Вороного — основний елемент візуальної складової кліпу (Рисунок 15). Однак тут додається новий тип генераторів — вершини гарнітури шрифта. Дані про вершини гарнітури шрифта (Рисунок 16) отримуються за допомогою бібліотеки Geomerative [40].



Рисунок 15

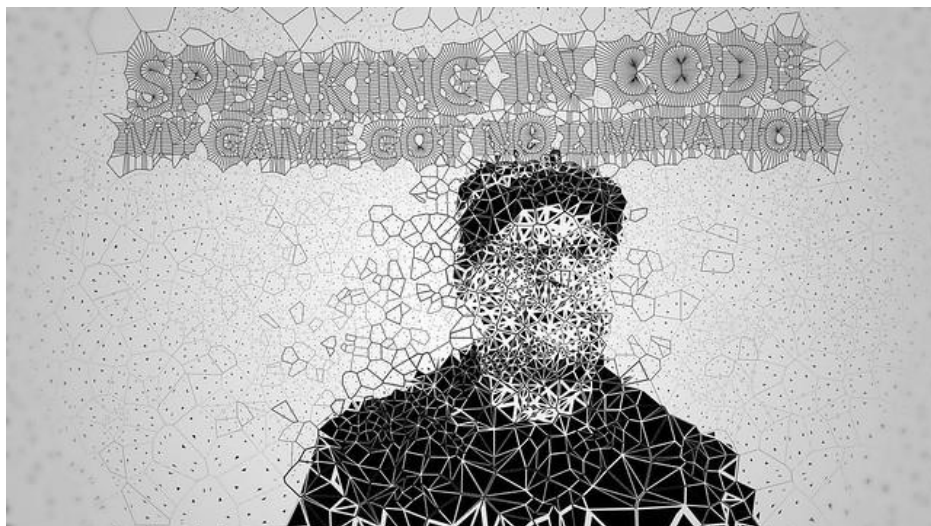


Рисунок 16

2.3.3 Декоративний дизайн

Використання діаграм Вороного в декоративному дизайні, тобто для дизайну орнаментів, є дуже зручним через вищезазначені властивості симетричності діаграми Вороного.

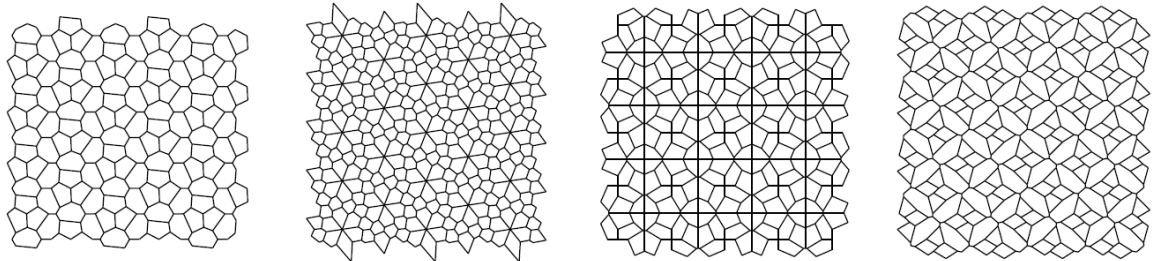


Рисунок 17 – Приклад орнаментів з діаграми Вороного на симетричних точках

В цьому випадку діаграма Вороного перетворює дуже просто визначену множину генераторів в складний орнамент. Широкий спектр орнаментів, які можна отримати таким чином, роблять діаграму Вороного потужним інструментом в орнаментальному дизайні.

З'єднуючи підходи декоративного дизайну та використання діаграми Вороного для генерації зображень, можна отримати мозаїчний ефект (Рисунок 18) [41].

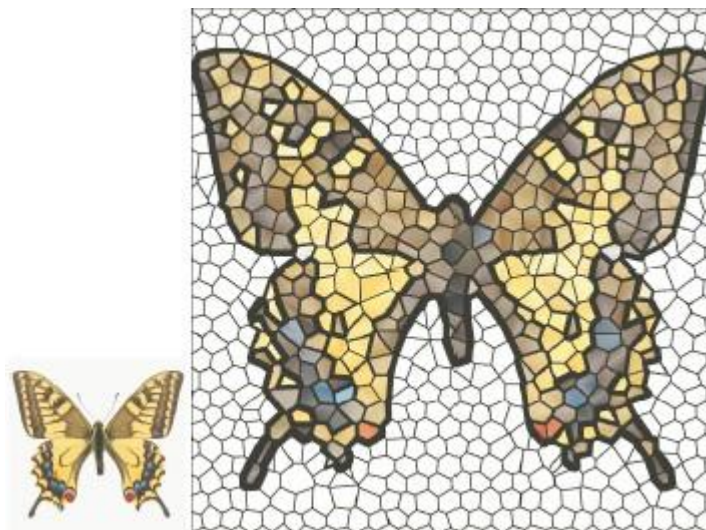


Рисунок 18

2.3.4 Фрактали Вороного

Фракталом називається самоподібна нерегулярна структура. Іншими словами, фрактал — це фігура, малі частини якої при збільшенні подібні до неї самої. Поняття фракталу було введено Бенуа Мандельбротом [42].

Щоб створити фрактал за допомогою діаграми Вороного, спочатку будуємо діаграму на деякій множині точок. Далі додаємо нові точки та створюємо меншу діаграму Вороного в кожному з вже існуючих регіонів Вороного. Рекурсивно повторюємо цю процедуру до отримання задовільного результату. Більш детально про фрактали Вороного розказано в [43]. Ілюстрацію процесу можна побачити нижче (Рисунок 19):

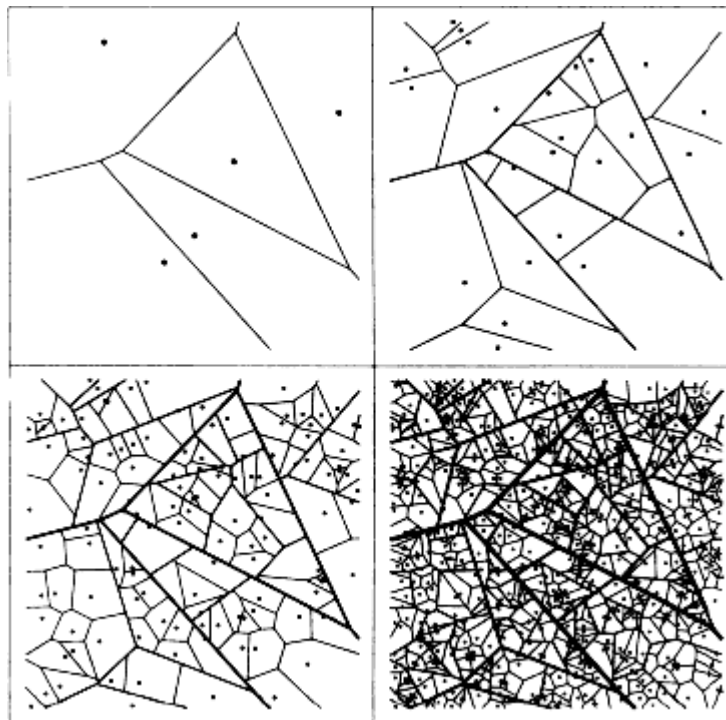


Рисунок 19 – Ілюстрація побудови фрактала

Ця техніка може бути застосована для генерації патернів, що схожі на автомобільні дороги, листки рослин, крила метеликів чи абстрактні твори мистецтва. Нижче (Рисунок 20) можна побачити приклад фрактала Вороного з трьома рівнями рекурсії на точках всередині кола. Перший рівень використовує 20 точок, другий та третій — 400 та 8000 відповідно.

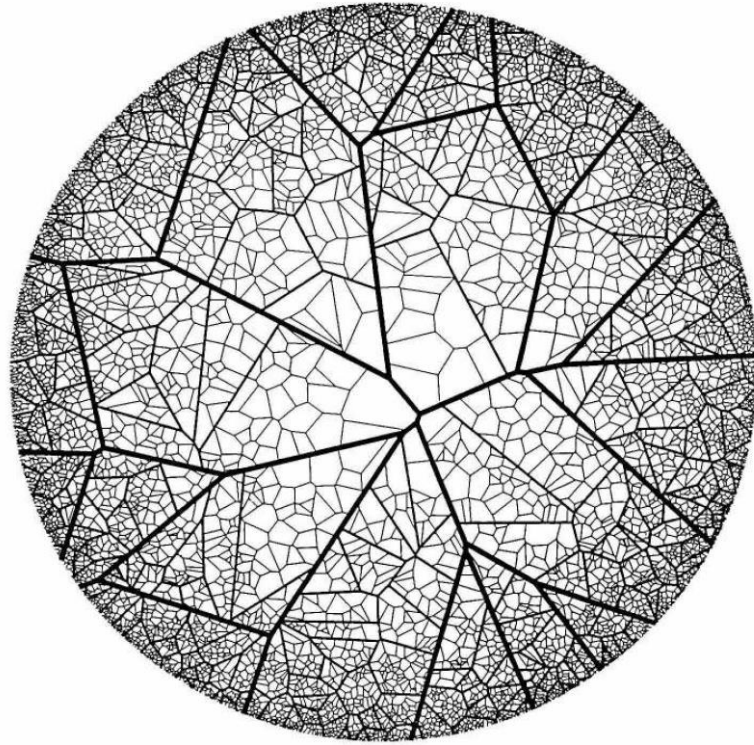


Рисунок 20

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Огляд існуючих рішень

В 1960-ті роки, як було зазначено вище, для перенесення згенерованих комп'ютером зображень на папір використовувались плотери. Їх використовують і зараз, але виключно для того, щоб повністю повторити оригінальний процес створення комп'ютерного витвору мистецтва на папірі. Серед сучасних комп'ютерних митців такі приклади використання плотерів можна знайти наприклад в Андерса Гоффа [44] та Тайлера Хоббса [45].

На сьогоднішній день є безліч інструментів, програм, фреймворків та мов, які спрощують створення власного комп'ютерного генеративного мистецтва. Розглянемо деякі з них.

3.1.1 openFrameworks

openFrameworks — це популярний відкритий набір інструментів для C++, зроблений для допомоги в творчому процесі завдяки інтуїтивному підходу та простору для експериментів [46]. Він був розроблений Захарі Ліберманом у 2005 році.

openFrameworks призначений для роботи як загальний інструмент який включає в себе багато бібліотек для роботи з комп'ютерною графікою, аудіо, шрифтами, відео та навіть комп'ютерним баченням.

3.1.2 Cinder

Cinder (libcinder) — бібліотека для C++ з відкритим кодом, яка призначена для генерації зображень та комп'ютерного мистецтва [47]. Сьогодні Cinder вважається C++ альтернативою Processing, який розглянуто в наступному пункті.

Основна відмінність Cinder та openFrameworks полягає в тому, що Cinder більше направлений на швидке виконання програм, тому використовує бібліотеки, специфічні від системи, на якій він розгорнутий. openFrameworks у свій час дозволяє легший контроль над своїми бібліотеками.

Cinder використовується в основному для великих проектів, таких як арт-інсталяції, де важлива швидкість генерації анімацій.

3.1.3 Processing

Для практичної реалізації поставленої задачі я використовував відкриту мову програмування та середовище розробки Processing [48].

Processing — це інструмент, створений для візуального електронного мистецтва, візуального та графічного дизайну, нових медіа. Він був створений з метою навчання людей основам комп'ютерного програмування у візуальному контексті. Processing являє собою легкий і швидкий інструментарій для програмування зображень, анімацій та інтерфейсів. Він використовується не тільки як засіб візуалізації, а і як професійний виробничий інструмент.

На сьогоднішній день Processing є найпопулярнішим інструментом для створення комп'ютерного мистецтва. Його використовують професійні художники, дизайнери, студенти та любителі.

Processing є доробкою об'єктно-орієнтовної мови програмування Java з відкритим кодом та має додаткові класи, що полегшують роботу з графічними елементами. Кожна програма, написана на Processing — це програма Java, яка використовує клас PApplet. Всі додаткові класи, визначені розробником в Processing є підкласами класу PApplet.

Мова Processing та відповідна IDE стали основою для розробки інших проектів, пов'язаних з програмуванням у візуальному контексті, наприклад p5.js, [49] Processing.py, Arduino та Wiring.

3.2 Практична реалізація

При дослідженні структур обчислювальної геометрії в комп'ютерному мистецтві можна помітити, що всі вони мають певні візуальні характеристики та можуть бути зображені в дуже різних контекстах.

Акцентуючи увагу на одній з цих структур — діаграмі Вороного, а саме її динамічній модифікації — бачимо, що в реалізаціях витворів комп'ютерного мистецтва найчастіше за все центри клітин Вороного рухаються згідно наперед заданих параметрів, наприклад, математичних функцій.

Запропонована в даній роботі модифікація використання діаграми Вороного в комп'ютерному мистецтві реалізує алгоритм руху точок-генераторів діаграми Вороного в залежності від звукового сигналу з зовнішнього синтезатора. Проводиться аналіз звукового сигналу — обраховується амплітуда сигналу та значення частоти основної ноти в сигналі. На основі цих параметрів визначаються параметри руху точок, на яких потім будується діаграма Вороного.

Реалізація виконувалася за допомогою IDE для мови програмування Java – JetBrains IntelliJ IDEA.

3.2.1 Клітинна структура

Даний проект використовує алгоритм Voids, описаний детально в роботі Крейга Рейнольдса [50]. Цей алгоритм симулює поведінку зграї птахів, реалізуючи наступні евристичні:

- Згуртованість: кожний представник зграї спрямовує свій рух до середнього положення найближчих сусідів
- Розділення: спрямування руху від скупчення великої кількості представників зграї
- Вирівнювання: спрямування своєї траєкторії в напрямку середнього курсу місцевих зграждань

- Спрямованість до центру: для того щоб всі точки перебували близько центра екрану — ця евристика була додана мною для покращення візуалізації системи

Набір таких евристик дозволяє створювати майже повністю автономну систему точок, які будуть завжди рухатися, але тим не менш створювати певну зв'язну візуальну структуру (Рисунок 21).

3.2.2 Аналіз звукового сигналу

Звуковий сигнал генерується зовнішнім синтезатором і потім аналізується за допомогою вбудованої в Processing бібліотеки для роботи зі звуком `processing.sound`. В цій бібліотеці реалізовані такі класи, що застосовуються для аналізу сигналу в моїй програмі: `FFT` — швидке перетворення Фур'є, `Amplitude` – для аналізу амплітуди сигналу та `AudioIn` — клас для передачі звукового сигналу зі звукової карти безпосередньо в Processing для аналізу.

Вхідний звуковий сигнал аналізується за допомогою класів `FFT` та `Amplitude`. Отримані параметри частоти основної ноти та амплітуди сигналу визначають параметри евристик запропонованого вище алгоритму `Voices`.

Наступним кроком ці точки використовуються як генератори для діаграми Вороного. Побудова діаграми Вороного відбувається методом побудови конусів. В процесі побудови діаграми Вороного використовується бібліотека `quickhull3d` [51] для швидкої побудови опуклої оболонки.

Серед візуальних характеристик діаграми Вороного на цьому наборі генераторів із запропонованим алгоритмом руху точок можна відмітити наступні:

- Точки в скупченні показані більшим розміром, ніж точки, які мають мало сусідів.
- Клітини діаграми Вороного візуалізовані не шляхом виведення на екран самих ребер діаграми Вороного – всі ребра зрушені в сторону центра клітини Вороного (точки-генератора).

- Вища амплітуда вхідного звукового сигналу збільшує динаміку системи та збільшує силу розділення точок.
- Вища частота основної ноти вхідного звукового сигналу збільшує непередбачуваність руху точок шляхом додавання випадковості в рух точок.

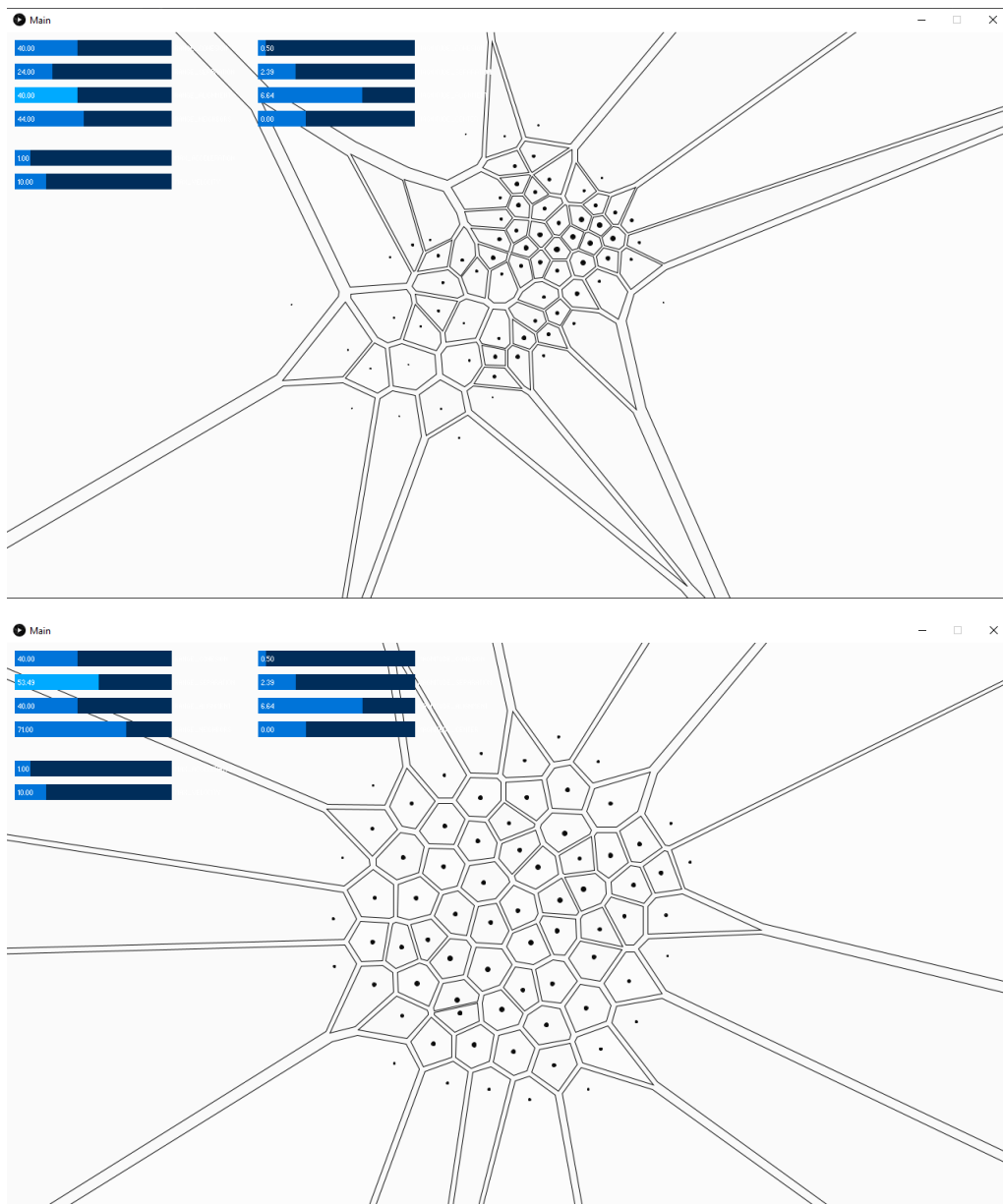


Рисунок 21 – Клітинна структура

ВИСНОВКИ

З розвитком сучасних комп'ютерних технологій, виникає все більше можливостей для розвитку комп'ютерного мистецтва. Структури обчислювальної геометрії часто використовуються у витворах мистецтва, згенерованих комп'ютером, однак досі не було загального огляду застосування цих структур та особливостей їх візуальної складової.

У рамках дипломної роботи були описані особливості використання геометричних структур і інструментів обчислювальної геометрії в комп'ютерному мистецтві, наведені приклади відповідних робіт комп'ютерного мистецтва та можливі використання. Були розглянуті такі структури обчислювальної геометрії: випадкове бінарне дерево, дерево квадрантів та дерево октантів, k-d дерево, триангуляція Делоне та діаграма Вороного.

Проаналізовано топологічні властивості геометричних структур, які впливають на їхню візуальну привабливість та відповідно можливість використання в мистецтві.

Було проведено огляд існуючих програмних забезпечень, що використовуються у комп'ютерному мистецтві.

Результатом роботи є запропонований підхід до використання динамічної діаграми Вороного в комп'ютерному мистецтві, що полягає в використанні звукового сигналу з синтезатора для контролювання параметрів руху точок-генераторів діаграми Вороного. Був розроблений програмний додаток, що реалізує запропонований підхід. Отриманий додаток наглядно демонструє естетичні характеристики діаграми Вороного та перспективи її використання в комп'ютерному мистецтві.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Boden M., Edmonds E., "What is Generative Art?," // *Digital Creativity*, Vol. 20, 2009. pp. 21-46.
2. Edward B. The Never-Before-Told Story of the World's First Computer Art (It's a Sexy Dame) // *The Atlantic*. 2013. URL: <https://www.theatlantic.com/technology/archive/2013/01/the-never-before-told-story-of-the-worlds-first-computer-art-its-a-sexy-dame/267439/>
3. O'Hanrahan E. Drawing Machines: The machine produced drawings of Dr. D. P. Henry in relation to conceptual and technological developments in machine-generated art (UK 1960–1968). 2005.
4. Noll A.M., "The Beginnings of Computer Art in the United States: A Memoir," // *Leonardo*, Vol. 27, No. 1, 1994. pp. 39-44.
5. Bense M.&N.G. rot 19. Computer-Grafik Stuttgart. 1965.
6. Preston S. Art ex Machina, // *The New York Times*. April 1965. P. X23.
7. // *Computer-Grafik (Nake & Nees)*: [сайт]. [1965]. URL: <http://dada.compart-bremen.de/item/exhibition/171>
8. Williams K.N.A.R.H., "Computer Program for Artists: ART 1," // *Leonardo*, Vol. 3, 1970. pp. 439-442.
9. McCormack J.A.B.O.A.D.A.A.M.J.A.M.G.A.W.M., "Ten Questions Concerning Generative Computer Art," // *Leonardo*, Vol. 47, Квітень 2014. pp. 135-141.
10. Knuth D.E. Draft of Section 7.2.1.6: Generating All Trees // In: *The Art of Computer Programming* / Ed. by Knuth D.E. 2005. P. 15.
11. Mahmoud H.M. *Evolution of Random Search Trees*. New York: John Wiley and Sons Ltd, 1992.

12. Aldous D. Probability Distributions on Cladograms // In: Random Discrete Structures / Ed. by Aldous D., Pemantle R. Springer-Verlag, 1996. pp. 1-18.
13. Devroye L.A.K.P. The botanical beauty of random binary trees // Graph Drawing: 3rd Int. Symp. September 1996. pp. 166-177.
14. Finkel R.A.B.J., "Quad Trees: A Data Structure for Retrieval on Composite Keys.," // Acta Inf., Vol. 4, Березень 1974. pp. 1-9.
15. Raptopoulou K.A.V.M.A.M.Y. Towards Quadtree-Based Moving Objects Databases Вересень 2004. pp. 230-245.
16. Figueiredo M.A.M.L.A.F.T., "A Survey on Collision Detection Techniques for Virtual Environments," // Proc. of V Symposium in Virtual Reality, Brasil, Vol. 307, Січень 2002.
17. Rokicki T.G. An Algorithm for Compressing Space and Time 2006. URL: <https://www.drdoobs.com/jvm/an-algorithm-for-compressing-space-and-t/184406478>
18. Kwok R. 189D0 2016. URL: <https://ravenkwok.com/189d0/>
19. Kwok R. Stickup 2015. URL: <https://ravenkwok.com/stickup/>
20. Kwok R. 1DDCB 2016. URL: <https://ravenkwok.com/1ddcb/>
21. Meagher D., "Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer," Жовтень 1980.
22. Elseberg J.A.M.S.A.S.R.A.N.A., "Comparison on nearest-neighbour-search strategies and implementations for efficient shape registration," // Journal of Software Engineering for Robotics (JOSER), Vol. 3, Січень 2012. pp. 2-12.
23. Kwok R. Form 333B 2015. URL: <https://openprocessing.org/sketch/256649>

24. Bentley J.L. Multidimensional Binary Search Trees Used for Associative Searching September 1975. Vol. 18. No. 9.
25. Moore A. An introductory tutorial on kd-trees January 1991.
26. Bentley J.L. K-d trees for semidynamic point sets Berkley, California, United States. 1990. pp. 187–197.
27. Buchanan A.A.F.A., "Interactive Feature Tracking using K-D Trees and Dynamic Programming," // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, Липень 2006. pp. 626-633.
28. Sajid Hussain H.G. Fast kd-Tree Construction for 3D-Rendering Algorithms Like Ray Tracing // In: Advances in Visual Computing / Ed. by Sajid Hussain H.G. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. pp. 681-690.
29. Knecht K.A.K.R. Generating Floor Plan Layouts with K-d Trees and Evolutionary Algorithms // XIII Generative Art Conference. 2010.
30. Volz M. k-d tree URL: <https://marcusvolz.com/#k-d-tree>
31. Kwok R. Autotroph 2018. URL: <https://ravenkwok.com/autotroph/>
32. Kwok R. 1DE94 2020. URL: <https://ravenkwok.com/1de94/>
33. Delaunay B., "Sur la sphère vide. A la mémoire de Georges Voronoï," // Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, No. 6, 1934. pp. 793-800.
34. Voronoi G., "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième Mémoire: Recherches sur les paralléloèdres primitifs," // J. Reine Angew. Math., No. 134, 1908. pp. 198–287.
35. Lejeune Dirichlet G., "Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen.," // Über die

Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen., Vol. 40, 1850. P. 40.

36. Kaplan C.S. Voronoi Diagrams and Ornamental Design // Proceedings of the First Annual Symposium of the International Society for the Arts, Mathematics, and Architecture (ISAMA 1999, San Sebastián, Spain, 71111 June 1999). 1999. pp. 277-283.
37. Reem D., "The geometric stability of Voronoi diagrams with respect to small changes of the sites," // Computing Research Repository - CORR, Березень 2011.
38. Kwok R. Skyline 2015. URL: <https://ravenkwok.com/skyline/>
39. Kwok R. Greatness 2016. URL: <https://ravenkwok.com/greatness/>
40. Marxer R. // Geomerative. URL: <http://www.ricardmarxer.com/geomerative/>
41. Dobashi Y.A.H.T.A.J.H.A.N.T., "A Method for Creating Mosaic Images Using Voronoi Diagrams," Січень 2002.
42. Mandelbrot B.B. The Fractal Geometry of Nature. Henry Holt and Company, 1983.
43. Shirriff K., "Generating fractals from Voronoi diagrams," // Computers & Graphics, Vol. 17, No. 2, March 1993. pp. 165-167.
44. Hoff A. Plotter Drawings URL: <https://img.inconvergent.net/plot/>
45. Hobbs T. // Selected Drawings & Paintings. URL: <https://tylerxhobbs.com/selected-works/drawings-and-paintings>
46. // openFrameworks: [сайт]. URL: <https://openframeworks.cc>
47. // Cinder: [сайт]. URL: <https://libcinder.org>
48. // Processing: [сайт]. [2020]. URL: <https://processing.org>
49. // p5.js: [сайт]. URL: <https://p5js.org>

50. Reynolds C.W., "Flocks, Herds and Schools: A Distributed Behavioral Model," // SIGGRAPH Comput. Graph., July 1987. pp. 25–34.
51. Lloyd J. QuickHull3D: A Robust 3D Convex Hull Algorithm in Java
URL: <https://www.cs.ubc.ca/~lloyd/java/quickhull3d.html>
52. Schlegel A. controlP5 2015. URL: <http://www.sojamo.de/libraries/controlP5/>