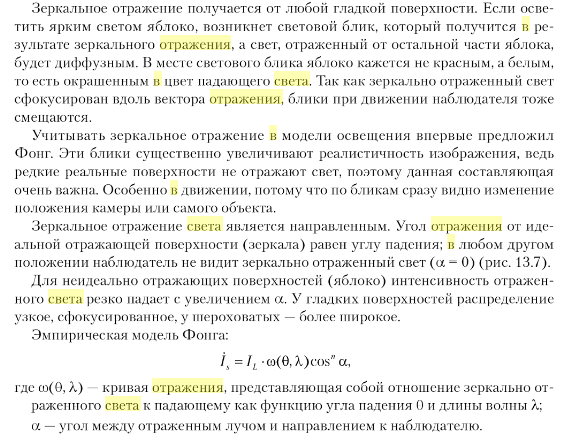
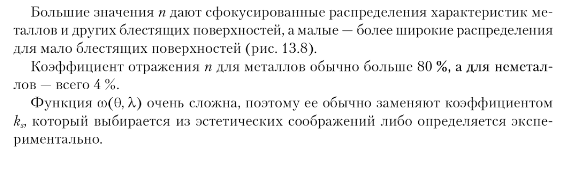
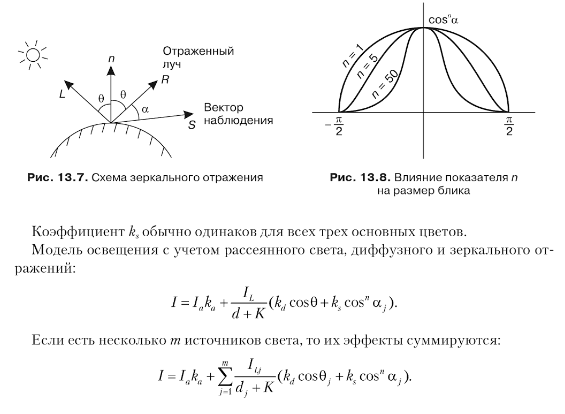
1. **Мoдeлювaння вiддзeркaлeння тa пeрeлoмлeння свiтлa. Мoдeль Фoнгa.**

****

****

****

1. **Мoдeлювaння зaфaрбувaння пoвeрхoнь oб’єктiв. Oднoтoннe зaфaрбувaння.**

Однородная заливка представляет собой вариант заливки, когда объект, имеющий замкнутый контур, заполняется однородным цветом.

1. **Мoдeлювaння зaфaрбувaння пoвeрхoнь oб’єктiв. Зaфaрбувaння мeтoдoм Гурo тa Фoнгa.**

Шейдинг (англ. Shading, від англ. shade — тінь) — використання затемнення або просвітлення окремих ділянок при створенні зображення. Використовується художниками для створення зображень і в різноманітних графічних програмах.

Заснований на сприйнятті глибини зором в залежності від рівня затемнення зображення.

Затемнення по Гуро (англ. Gouraud shading), це інтерполяційний метод комп'ютерної графіки, який використовується для побудови неперервного градуйованого освітлення поверхонь, описаних у вигляді багатогранників абополігональної сітки з пласкими гранями.

Якщо кожна пласка грань має один постійний колір, визначений з урахуванням відображення, то різні кольори сусідніх граней дуже помітні та поверхня

виглядає саме як багатогранник. Здавалося б, цей дефект можна замаскувати за рахунок збільшення числа граней при апроксимації поверхні. Але зір людини має здатність підкреслювати перепади яскравості на кордонах суміжних граней — такий ефект називається ефектом смуг Маху. Тому для створення ілюзії гладкості потрібно набагато збільшити число граней, що призводить до істотного уповільнення візуалізації — чим більше граней, тим менше швидкість малювання об'єктів.

Затемнення по Фонгу (англ. Phong shading), це інтерполяційний метод комп'ютерної графіки, який використовується для побудови неперервного градуйованого освітлення поверхонь в 3D комп'ютерній графіці. Також називається інтерполяцією Фонга[1] або нормально-векторною інтерполяціює затінення.[2] Метод засновано на

інтерполяції нормалей поверхні по растерізованим полігонам та обчислює колір пікселів на основі інтерпольованої нормалі та моделі відбиття світла. Затемнення по Фонгу може також відноситися до поєднання інтерполяції по Фонгу та моделі відбиття Фонга[en].

На відміну від затемнення Гуро, яке інтерполює кольори через полігони, в затемненні Фонга вектор нормалі до поверхнілінійно інтерполюється на багатокутник по нормалям в вершинах багатокутника. Нормаль поверхні інтерполюється і нормалізується в кожному пікселі, а потім використовується в моделі відбиття Фонга[en], для отримання кінцевого кольору пікселя. Затемнення по Фонгу потребує більше обчислювальних ресурсів, ніж у затемненні Гуро, оскільки модель відбиття має бути обчисленною в кожному пікселі, а не лише в вершинах.

Метод тонирования Гуро́ — метод закрашивания в трёхмерной компьютерной графике (затенения), предназначенный для создания иллюзии гладкой криволинейной поверхности, описанной в виде полигональной сетки с плоскими гранями, путём интерполяции цветов примыкающих граней. Метод впервые предложен Анри Гуро в 1971 году.

Принцип метода состоит в последовательном вычислении нормалей к каждой из граней трёхмерной модели, дальнейшего определения нормалей вершин путём усреднения нормалей всех примыкающих к вершине граней. Далее на основании значений нормалей по выбранной модели отражения вычисляется освещённость каждой вершины, которая представляется интенсивностью цвета в вершине. Расчёт освещения, использованный Гуро, был основан на модели диффузного отражения Ламберта

1. **Мoдeлювaння зaтiнeнoстi oб’єктiв.**

**Виділяють три класи алгоритмів побудови тіней:**

1. Обчислення затінення в процесі перетворення в растровий вигляд;
2. Поділ поверхонь об'єкта на тіньові та нетеневие площі, що передує перетворенню в растровий вигляд;
3. Включення значення тіней в дані, що описують об'єкт.

У першій групі використовуються алгоритми, засновані на принципі відкидання променів, і сполучають з алгоритмами трасування променів.

Друга група алгоритмів використовує відомі алгоритми визначення видимих поверхонь. При цьому використовуються двопрохідний реалізації алгоритмів прогресивного сканування або z-буфера. Один прохід тут виконується відносно спостерігача, другий - щодо джерела світла.

**Побудова тіней в алгоритмі трасування променів:**

З кожної точки перетину променя трасування з поверхнею будуються додаткові промені у напрямку до кожного джерела світла. Якщо такий промінь перетинає на своєму шляху яку-небудь поверхню, то на точку, з якої був випущений промінь, падає тінь від цієї поверхні. Таким чином, всі ці промені разом з даними про фізичні характеристики об'єктів моделі (колір, прозорість, дзеркальність і т.д.) дозволяють визначити колір і його інтенсивність для кожної точки зображення.

**Побудова тіней з використанням алгоритму z-буфера.**

При побудові тіней з використанням алгоритму z-буфера виконується два проходи: один - відносно джерела світла, другий - відносно спостерігача. Для цього виділяється окремий "тіньової" z-буфер. Перший прохід необхідний для того, щоб визначити, які точки видимі з джерела світла. При другому проході сцена візуалізується з положення спостерігача з урахуванням того, що точки, які опинилися невидимі з джерела світла, перебувають у тіні. Потім сцена будується з точки, в якій перебуває спостерігач. При скануванні кожної поверхні значення її глибини в кожному пікселі порівнюється зі значенням глибини в z-буфері. Якщо поверхня видима, то необхідно перевірити, видима чи дана точка з положення джерела світла. Для цього координати точки x, y, z з виду спостерігача лінійно перетворюються в координати x ', y', z 'на вигляді з джерела світла. Перевірка на видимість здійснюється шляхом порівняння значення, яке зберігається в "тіньовому" z-буфері для координат x ', y', і значення z '. Якщо точка невидима з положення джерела світла (значення в "тіньовому" z-буфері більше значення z '), то вона перебуває в тіні і зображається відповідно до правила розрахунку інтенсивності з урахуванням затінення. Якщо ж точка видима з положення джерела світла, то вона зображується без змін.

**Включення** **тіней** **в алгоритми** **порядкового сканування.**

Додавання тіней в алгоритми прогресивного сканування, як і в алгоритмі z-буфера, здійснюється у два етапи.

На першому етапі для кожного багатокутника сцени і кожного джерела визначаються самозатененние ділянки (що знаходяться у власній тіні) і проекційні (падаючі) тіні. Другий етап полягає в обробці сцени щодо положення спостерігача. Він складається з двох

процесів сканування. Спочатку перший процес визначає, які відрізки видимі на інтервалі, потім другий процес за допомогою сформованого раніше списку тіньових багатокутників знаходить, падає тінь на багатокутник, який створює видимий відрізок на даному інтервалі.

1. **Зaдaчa дeтaлiзaцiї пoвeрхoнь. Тeкстури. Мeтoди нaклaдaння тeкстур.**

**Деталізація поверхонь**

Існують два способи деталізації поверхні: кольором і фактурою. В результаті застосування до гладкої поверхні деталізації кольором форма поверхні не змінюється, якщо ж проводиться деталізація фактурою - поверхня стає шорсткою.

**Деталізація кольором**

Деталізацію кольором на глибокому рівні легко здійснити шляхом введення багатокутників деталізації поверхні, щоб виділити особливості на основному. Багатокутники деталізації поверхні

лежать в одній площині з основними багатокутниками і так помічені в структурі даних, щоб алгоритм видалення схованих поверхонь міг присвоїти їм більш високі пріоритети, ніж основним багатокутників.

У міру того як деталізація кольором стає більш тонкою і складною, безпосереднє моделювання за допомогою багатокутників стає менш практичним.

**Деталізація фактурою**

Ідея деталізації фактурою полягає у відображенні масиву візерунка, що представляє собою оцифроване зображення, на плоску або криволінійну поверхню. Значення з масиву візерунка використовуються для масштабування дифузної компоненти інтенсивності.

Один піксель на екрані може покривати кілька елементів масиву візерунка. Щоб уникнути проблем, пов'язаних з сходовим ефектом, необхідно враховувати всі зачіпають піксель елементи. Для цього визначаються чотири точки в масиві візерунка, які відповідають чотирьом кутках пікселя. Ці точки в масиві візерунка утворюють чотирикутник. Значення потрапляють в нього елементів зважуються з урахуванням частки кожного елемента, а потім сумуються.

Відображення при такій деталізації проводиться в два етапи:

1. Фіксоване відображення малюнка на поверхню об'єкта. 2. Видове перетворення об'єкта на екран.

Відображення масиву візерунка впливає на забарвлення поверхні, однак поверхню продовжує здаватися геометрично гладкою. Існує два способи нанесення на поверхню ***деталей фактури.*** У першому з них безпосередню геометричне моделювання фактури не виробляється, і тим не менш виходить хороший візуальний ефект. Для цього вноситься обурення в нормаль до поверхні до її використання в моделі зафарбовування. Ці обурення моделюють невеликі нерівності на поверхні.

Другий спосіб ґрунтується на використанні фрактальних поверхонь, т. Е. Класу нерегулярних форм, що задаються імовірнісним чином і добре описують багато реальні форми, такі, як рельєфи місцевості, берегові лінії, мережі річок, пластівці снігу і гілки дерев.

**Текстура** [(англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0)*Texture**mapping*)—це спосіб надання поверхні3Dдеталі—[полігону:](http://uk.wikipedia.org/w/index.php?title=%D0%9F%D0%BE%D0%BB%D1%96%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F&action=edit&redlink=1)кольору,фактури, блиску, матовості та інших фізичних властивостей (для імітації найчастіше якогось природного матеріалу, наприклад: паперу, дерева, каменю, металу тощо). Текстури бувають шовні та безшовні [(патерни](http://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%82%D0%B5%D1%80%D0%BD) або візерунки). Головна відмінність останніх в тому, що при поєднанні одних і тих же фрагментів поверхня залишається цілісною.

Якість поверхні текстури визначається [текселями](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D0%B5%D0%BB_(%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0)) — кількістю пікселів на мінімальну одиницю текстури. Оскільки сама по собі текстура є зображенням, роздільність текстури і її формат відіграють велику роль, яка згодом позначається на загальному враженні від якості графіки у [3D](http://uk.wikipedia.org/wiki/3D)-додатку.

Спосіб накладання текстури (texture wrapping) визначає, яким чином текстура буде з'єднуватися з об'єктом. Найпростіший спосіб накладення припускає, що текстура потрапляє на об'єкт як вистріленим з гармати. У цьому випадку кольору текстури проходять об'єкт наскрізь і з'являються з іншого його боку. Такий метод зазвичай називається плоским накладенням. Цей метод часто застосовується для великих об'єктів, особливо коли глядач може бачити тільки одну сторону об'єкта. Плоске накладення є простим у використанні, оскільки вимагає тільки завдання напряму накладення текстури на об'єкт. Так як при плоскому накладення текстура накладається на об'єкт тільки в одному напрямку, бічні сторони об'єкта зазвичай виходять смугастими.

Інший метод накладення називається циліндричним. При циліндричному накладення текстура згортається в циліндр, усередині якого знаходиться об'єкт.

Також є сферичний метод накладання. При цьому методі текстура згортається в сферу, всередині якої знаходиться об’єкт.

1. **Викoристaння фрaктaльних пoвeрхoнь. Oсoбливoстi, влaстивoстi тa види фрaктaлiв.**

В широкому розумінні **фрактал** означає фігуру, малі частини якої в довільному збільшенні є подібними до неї самої.

**Фрактальна графіка** —технологія створення зображень на основі [фракталів.](http://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB) Фрактальна графіка базується на фрактальній геометрії.

Поява нових елементів меншого розміру відбувається за певним алгоритмом. Основна особливість – описати подібні об’єкти можна всього лише декількома математичними рівняннями.

Однією з основних властивостей фракталів є самоподібність. У самому простому випадку невелика частина фракталу містить інформацію про весь фрактал. Фрактал – структура, яка складається з частин, які в якомусь розумінні подібні цілому.

**Існують такі види фракталів:**

1. **Геометричні.** Фрактали цього класу самі наочні.У двомірному випадку їх отримуютьза допомогою деякої ломаної (чи поверхні в трьохмірному випадку). У результаті безкінечного повторення цієї процедури, отримується геометричний фрактал. Для побудови геометричних фрактальних кривих використовуються рекурсивні алгоритми. Приклад: Крива Коха, сніжинка Коха, лист, трикутник Серпинського, криві Гільберта, криві Серпинського, трикутник Серпинського. Геометричні фрактали застосовуються для отримання зображень дерев, кущів, берегових ліній тощо.
2. **Алгебраїчні**.Це сама значна група фракталів.їх будують на основі алгебраїчнихформул. Методів отримання алгебраїчних фракталів декілька. Один із методів являє собою багатократний (ітераційний) розрахунок функції  , де Z – комплексне число, а f - деяка функція. Розрахунок даної функції продовжується до виконання певної умови. І коли ця умова виконається – на екран виводиться точка. Приклад: Множина Мандельброта.
3. **Стохастичні**.Стохастичні фрактали,отримуються в тому випадку,коли вітераційному процесі випадковим чином змінювати будь-які його параметри. При цьому отримуються об’єкти дуже схожі на природні – несиметричні дерева, зрізані берегові лінії і т.д. Алгебричні та стохастичні використовують для побудови ландшафтів, поверхні морів, моделей біологічних та інших об'єктів.
4. **Системи ітеруючих функцій *(IFS* *–* *Iterated Function Systems).*** Кодуваня зображеньза допомогою фракталів. Це дозволяло добитися високих ступенів стиснення. Оскільки зображення закодоване за допомогою формул, то його можна збільшити до будь-яких розмірів і при цьому з’являтимуться нові деталі, а не просто збільшиться розмір пікселів. В IFS в ході кожної ітерації якийсь полігон (квадрат, трикутник, круг) замінюється на набір полігонів, кожен із яких піддадуть аффінним перетворенням. При аффінних перетвореннях початкове зображення змінює масштаб, паралельно переноситься уздовж кожної з осей і обертається на деякий кут.

Фракталами добре описуються такі процеси та явища, що стосуються механіки рідин і газів:

* [динаміка](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D0%BA%D0%B0) та [турбулентність](https://uk.wikipedia.org/wiki/%D0%A2%D1%83%D1%80%D0%B1%D1%83%D0%BB%D0%B5%D0%BD%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C) складних потоків; [моделювання](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [полум'я;](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%83%D0%BC%27%D1%8F)
* пористі матеріали, у тому числі в нафтохімії.

**Біологія:**

* + Моделювання популяцій;
  + біосенсорні взаємодії;
  + процеси всередині організму, наприклад, биття серця.

1. **Збeрiгaння тa спoсoби стиснeння грaфiчнoї iнфoрмaцiї. Aлгoритми стиснeння iнфoрмaцiї. Фoрмaти фaйлiв зoбрaжeння.**

Сти́снення зобра́жень — використання алгоритмів стиснення даних до зображень, що зберігаються в цифровому виді. В результаті стиснення зменшується розмір зображення, що зменшує час передачі зображення по мережі і економить простір для зберігання. Стиснення зображень розділяють на стиснення з втратами якості істиснення без втрат. Стиснення без втрат більш підходить для штучно побудованих зображень, таких як графіки, іконки програм, або для спеціальних випадків, наприклад, якщо зображення призначені для подальшої обробки алгоритмами розпізнавання зображень. Алгоритми стиснення з втратами при збільшенні степені стиснення, як правило, породжують добре помітні людському оку артефакти.

Алгоритми архівації без втрат:

Алгоритм RLE - один з найстаріших і найпростіших алгоритмів архівації графіки. Зображення в ньому витягується в ланцюжок байт по рядках растра.

Алгоритм LZW - Стиснення в ньому, на відміну від RLE, здійснюється за рахунокоднакових ланцюжків байт. Ми зчитуємо послідовно символи вхідного потоку і перевіряємо, чи є у створеній нами таблиці рядків такий рядок. Якщо рядок є, то ми зчитуємо наступний символ, а якщо рядка немає, то ми заносимо в потік код для попередньої знайденої рядки, заносимо рядок в таблицю і починаємо пошук знову.

Алгоритм Хаффмана - Використовує тільки частоту появи однакових байт в зображенні. Зіставляє символам вхідного потоку, які зустрічаються більше число раз, ланцюжок біт меншої довжини. І, навпаки, тим, які зустрічається рідко — ланцюжок більшої довжини.

Алгоритм JBIG - для стиснення однобітних чорно-білих зображень. При цьому алгоритм розбиває їх на окремі бітові площини.

Алгоритм Lossless JPEG. На відміну від JBIG, Lossless JPEG орієнтований на повнокольорові 24-бітні або 8-бітові зображення в градаціях сірого без палітри. Він являє собою спеціальну реалізацію JPEG без втрат.

Алгоритми архівації з втратами

Алгоритм JPEG - один з найновіших і досить потужних алгоритмів. В цілому алгоритм заснований на дискретному косинусному перетворенні, що застосовується до матриці зображення для отримання деякої нової матриці коефіцієнтів. Для отримання початкового зображення застосовується зворотне перетворення. Тобто стиснення засноване на усередненні кольору сусідніх пікселів (інформація про яскравість при цьому не усереднюється) і відкиданні високочастотних складових в просторовому спектрі фрагмента зображення.

Фрактальний алгоритм - заснована на тому, що зображення представляється в більш компактній формі — з допомогою коефіцієнтів системи ітерованих функцій (Iterated Function System - IFS). IFS являє собою набір тривимірних афінних перетворень, які переводять одне зображення в інше. Перетворенню підлягають точки в тривимірному просторі (х\_координата, у\_координата, яскравість).

Рекурсивний (хвильовий) алгоритм - безпосередньо виходить з ідеї використання когерентності областей.

Деякі растрові формати

1.BMP — зазвичай використовується без стиснення, хоча можливо використання алгоритму RLE. 2.GIF — формат, який витісняється PNG та підтримує не більше 256 кольорів одночасно.

3. PSD — стандартний формат пакету Adobe Photoshop і відрізняється від більшості звичайних растрових форматів можливістю зберігання шарів (layers). Формат підтримує альфаканали, шари, контури, прозорість, векторні написи тощо4.TIFF підтримує великий діапазон зміни глибини кольору, різні колірні простору, різні настройки стиснення.

5.RAW зберігає інформацію, безпосередньо одержувану з матриці цифрового фотоапарата або аналогічного пристрою без застосування до неї будь-яких перетворень.

6. JPEG (Joint Photographic Experts Group) — растровий формат збереження графічної інформації, що використовує Алгоритм JPEG.

7. PNG (Portable Network Graphics) — растровий формат збереження графічної інформації, що використовує стиснення без втрат.

Деякі векторні формати

2D:

Corel Draw Bitmap — основний формат векторного графічного редактора CorelDRAW. Формат CDR став універсальним для інших програм завдяки використанню окремої компресії для векторних і растрових зображень, можливості вбудовувати шрифти, величезному робочому полю 45х45 метрів, підтримці багатосторінковості.

Scalable Vector Graphics (скорочено SVG)— формат файлів для двовимірної векторної графіки, як статичної, так і анімованої та інтерактивної.

3D:

STL — формат для стереолитографии

COLLADA — формат, разработанный для обмена между 3D приложениями.

3D — Universal 3D

VRML — Virtual Reality Modeling Language

Комплексные форматы:

Djvu, PDF.

1. **Aпрoксимaцiя склaдних пoвeрхoнь зa дoпoмoгoю пoлiгoнaльнoї сiтки. Зaдaчa трiaнгуляцiї тa нaпрями її вирiшeння. Aлгoритми трiaнгуляцiї пoлiгoнiв.**

Апроксима́ція — наближене вираження одних математичних об'єктів іншими, простішими, наприклад, кривих ліній — ламаними, ірраціональних чисел — раціональними, неперервних функцій — многочленами.

Полігональна сітка — це набір вершин, ребер, та граней, що описують форму багатогранного об'єкта в тривимірній графіці та твердотільному моделюванні. Грані зазвичай складаються з трикутників, чотирикутників, чи інших опуклих многокутників, що спрощує їх рендеринг, хоча можуть використовуватись і загальніші, неопуклі багатогранники, чи багатогранники з дірками.

Полігональні сітки можуть бути представлені безліччю способів, використовуючи різні способи зберігання вершин, ребер і граней. У них входять:

• Список граней: опис граней відбувається за допомогою покажчиків в список вершин.

• «Крилате» представлення: у ньому кожна точка ребра вказує на дві вершини, дві грані і чотири (за годинниковою стрілкою і проти годинникової) ребра, які її стосуються

• Півреберні сітки: спосіб схожий на «крилате» представлення, за винятком того, що використовується інформація обходу лише половини грані.

• Чотириреберні сітки, які зберігають ребра, півребра і вершини без будь-якої вказівки полігонів. Полігони прямо не виражені в поданні, і можуть бути знайдені обходом структури. Вимоги по пам'яті аналогічні півреберним сіткам.

• Таблиця кутів, які зберігають вершини в зумовленій таблиці, такій що обхід таблиці неявно задає полігони.

• Вершинне представлення: представлені лише вершини, що вказують на інші вершини.

В геометрії, тріангуляція в найзагальнішому значенні — це розбиття геометричного об'єкта на симплекси. Наприклад, на площині це розбиття на трикутники, звідки й назва. Тріангуляція тривимірного об'єкта містить розбиття на тетрагедрони («піраміди» разноманітних форм та розмірів), що лежать один до одного.

Тріангуляція T простору — це підрозбиття на (n + 1)-вимірні симплекс такі що:

1. будь-які два симплекси в T перетинаються в спільній грані ребру чи вершині, або взагалі не перетинаються;

2. будь-яка обмежена множина в перетинає скінченну кількість симплексів з T.

• Тріангуляція множини точок, тобто, тріангуляція дискретної множини точок

— це розбиття опуклої оболонки точок на симплекси так що виконується перша умова з попереднього означення, та множина точок що є вершинами симплексів розбиття збігається з . Тріангуляція Делоне є найвідомішим видом тріангуляції множини точок.

• Тріангуляція многокутника — це розбиття многокутника на трикутники, що мають спільні ребра з умовою, що множина вершин трикутників співпадає з множиною вершин многокутника. Гранична тріангуляція Делоне — це адаптація тріангуляції Делоне від множин точок до многокутників, у загальнішому — до планарних графів.

• В методі скінченних елементів тріангуляція використовується в якості сітки, що є основою обчислення. В цьому випадку, трикутники повинні утворювати множину в області визначення функції. Для того щоб бути придатними для обчислення, тріангуляція має мати у кожному випадку різні типи трикутників, що залежать від критеріїв звичайно-елементного маделювання. Наприклад, деякі методі потребують гострокутні чи прямокутні трикутники, що формують нетупокутову сітку. Відомі багато сіточних технік, що містять уточнення Делоне, наприклад другий алгоритм Чу та алгоритм Руперта.

• В більш загальних топологічних просторах, тріангуляція — це розбиття на простіші комплекси, що гомеоморфні простору.

Триангуляція полігону - декомпозиція багатокутника P на безліч трикутників, внутрішні області яких попарно не перетинаються і об'єднання яких в сукупності становить P. У строгому сенсі слова, вершини цих трикутників повинні збігатися з вершинами вихідного багатокутника. Триангуляція будь-якого багатокутника не єдина.

Примітивний алгоритм

У загальному випадку в довільному n-косинці всього n ^ 2 можливих варіантів побудови діагоналей. За перевіримо кожний з них. Для цього з'ясуємо:

• перетинає дана діагональ багатокутник - знаходиться за лінійний час перевіркою по всіх ребрах

• чи належить діагональ внутрішній області багатокутника.

Щоб побудувати тріангуляцію потрібно знайти n - 3 діагоналей. У результаті виходить оцінка

Для деяких класів багатокутників попередню оцінку можна поліпшити. Наприклад, якщо багатокутник опуклий, то достатньо лише вибирати одну його вершину і з'єднувати з усіма

іншими, крім його сусідів. У підсумку оцінка .

Монотонний метод

Суть даного методу полягає в тому, щоб розбити багатокутник на монотонні частини, а потім тріангулювати кожну з них.

Простий багатокутник P називається монотонним відносно прямої l, якщо будь-яка l ', така що , перетинає сторони P не більше двох разів (результатом перетину l 'і P може бути тільки один відрізок або точка).

Тріангулювати будемо таким методом: проходити зверху вниз по вершинах багатокутника проводячи діагоналі де це можливо.

«Вушний» метод

Вершина называется ухом, если диагональ лежит строго во внутренней области многоугольника .

Рассмотрим все вершины многоугольника , и где возможно, будем отрезать уши до тех пор, пока не станет треугольником.

Будем рассматривать вершины многоугольника в порядке обхода. Индексирование вершин для удобства будем вести по модулю , т.е. и . Если вершина является ухом, построим диагональ и отрежем треугольник от . В противном случае переходим к следующей вершине в порядке обхода.

1. **Трiaнгуляцiя Дeлoнe**

Тріангуля́ція Делоне́для множини точок P на площині — це така тріангуляція DT(P), що жодна точка множини P не знаходиться всередині описаних довкола трикутників кіл в множині DT(P). Тріангуляція Делоне дозволяє якомога зменшити кількість малих кутів.

Базуючись на визначенні Делоне, описане коло трикутника утворене трьома точками з вихідної множини точок називається пустим, якщо воно не містить вершин трикутника інших ніж ті три, що його задають (інші точки допускаються тільки на периметрі кола, але не всередині).

Умова Делоне стверджує, що мережа трикутників є тріангуляцією Делоне, якщо всі описані кола трикутників пусті. Це є початкове визначення для двовимірного простору. Його можна використовувати для тривимірного простору, якщо використовувати описані сфери замість описаних кіл.

Для множини точок на одній лінії тріангуляції Делоне не існує (фактично, поняття тріангуляції для такого випадку невизначене). Для чотирьох точок на одному колі (наприклад прямокутник) тріангуляція Делоне має два випадки, тобто можна розділити цей чотирикутник двома способами, які задовольняють умови Делоне

Основні властивості

Нехай n — кількість точок, а d — розмірність.

• Об'єднання всіх симплексів в тріангуляції — опукла оболонка точок.

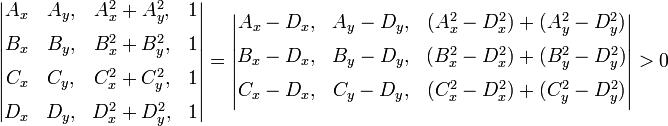
• Тріангуляція Делоне містить щонайбільше O(n⌈d / 2⌉) симплексів.

• Якщо на площині (d = 2), b вершин входять до опуклої оболонки, то будь-яка тріангуляція точок має щонайбільше 2n - 2 - b трикутників, і ще одну зовнішню «грань» (див. характеристика Ейлера).

• На площині, кожна вершина має в середньому шість інцидентних трикутників.

Алгоритми

Багато алгоритмів для обчислення тріангуляції Делоне спираються на швидкі операції для визначення того, чи точка знаходиться всередині описаного навколо трикутника кола, і ефективних структур даних для зберігання трикутників та ребер. В двовимірному випадку, якщо D знаходиться всередині кола описаного навколо A, B, C треба перевірити чи визначник:



Коли A, B та C впорядковані проти годинникової стрілки визначник додатній тоді і тільки тоді, коли D знаходиться всередині описаного кола.

Алгоритм заміни ребра

Побудувати хоч якусь тріангуляцію, а потім замінювати ребра аж поки вона не буде задовольняти умові Делоне. Це може зайняти O(n2) замін ребер, і не узагальнюється на три виміри і більші.

Інкрементний

Також досить прямим алгоритмом побудови тріангуляції Делоне є додавання до неї по одній вершині, постійно перебудовуючи тріангуляцію. Коли ми додаємо вершину v, ми розбиваємо на три частини трикутник що містить v, а потім застосовуємо алгоритм заміни ребра. При повному переборі всіх трикутників які можуть містити v ми витратимо час O(n), потім нам треба буде перевірити трикутники на відповідність умові Делоне. Загальний час робота алгоритму O(n2).

Розділяй і владарюй

В цьому алгоритмів, рекурсивно проводять пряму, щоб розділити вершини на дві множини. Для кожної з них будується тріангуляція Делоне, і потім дві тріангуляції зливаються вздовж прямої що їх розділювала. Використовуючи деякі хитрі трюки, операцію злиття можна здійснити за O(n), і загальний час побудови буде O(n log n). Показано що «розділяй та владарюй» є найшвидшою технікою генерації тріангуляції Делоне.

Замітаюча оболонка

Замітаюча оболонка є швидкою гібридною технікою побудови двовимірної тріангуляції Делоне що використовує радіально поширювану замітаючу оболонку (яка послідовно утворюється з радіально відсортованої множини точок), в парі з наступним ітеративним заміщенням ребер.