

Пріоритет даної заявки заявляється відповідно до попередньої заявки №60/456,030, озаглавленої «Спосіб та пристрій для поліпшення якості відеозображення з низькою швидкістю передачі бітів», зареєстрованої 17 березня 2003 року і призначеної правонаступником даної заявки і, таким чином, навмисно об'єднаної з даною заявкою за допомогою посилань.

Винахід відноситься, загалом, до стиснення даних і, конкретніше, до основаних на блоках систем стиснення.

Кодування з перетворенням є загальною технологією стиснення зображення, яка ділить зображення на підзображення або блоки для обробки. Основане на блоках стиснення вносить артефакти між межами блоків, оскільки блоки кодуються незалежно. Отже, перетворення не бере до уваги кореляцію між межами блоків. Отже, технологія звичайно має наслідком зображення з низькою швидкістю передачі бітів, які піддані сильному стисненню, але містять сильні артефакти стиснення, такі як кубізація, шуми та розмазані рух.

У результаті були запропоновані декілька алгоритмів подальшої обробки для блокового згладжування, щоб зменшити і/або усунути артефакти стиснення. Проте, багато алгоритмів включають складні обчислення і можуть мати наслідком ефект повного розмивання меж вихідного зображення. Інші блокові фільтри не зберігають ефективно крайову інформацію і звичайно важко реалізуються в апаратному забезпеченні. Отже, існує необхідність простішого і/або ефективнішого способу блокового згладжування.

Спосіб та пристрій для обробки зображень, стиснутих при використанні основаного на блоках стиснення, можуть містити: визначення того, чи є два блоки сусідніми блоками; визначення того, чи є обидва сусідніх блоки поділеними; виконання фільтрації блокового згладжування на одному або більшій кількості крайових пікселів двох сусідніх блоків, якщо обидва сусідніх блоки не є поділеними. Визначення того, чи є обидва сусідніх блоки поділеними, може містити одержання значень дисперсії для кожного з двох сусідніх блоків; порівняння значень дисперсії з першим пороговим значенням; і визначення того, чи є обидва сусідніх блоки поділеними, основуючись на порівнянні значень дисперсії з першим пороговим значенням. Альтернативно, визначення того, чи є обидва сусідніх блоки поділеними, також може містити одержання присвоєного значення розміру блока; і використання присвоєного значення розміру блока, щоб визначити, чи є два сусідніх значення поділеними.

Спосіб та пристрій можуть додатково містити визначення того, чи є один з двох сусідніх блоків поділеним, якщо обидва сусідніх блоки не є поділеними; використання першого фільтра блокового згладжування на одному або більшій кількості крайових пікселів двох сусідніх блоків, якщо один з двох сусідніх блоків є поділеним; і використання другого фільтра блокового згладжування на одному або більшій кількості крайових пікселів двох сусідніх блоків, якщо жоден з двох сусідніх блоків не є поділеним.

Спосіб та пристрій можуть додатково містити одержання одного або більше значень відмінності для одного або більшої кількості крайових пікселів двох сусідніх блоків, якщо жоден з двох сусідніх блоків не є поділеним; порівняння одного або більше значень відмінності з другим пороговим значенням; і вибір другого фільтра блокового згладжування, основуючись на порівнянні одного або більше значень відмінності з другим пороговим значенням.

Одержання одного або більше значень відмінності може містити одержання значення відмінності між трьома крайовими пікселями двох сусідніх блоків; і вибір другого фільтра блокового згладжування може містити використання фільтра Гауса, якщо щонайменше два із значень відмінності більше, ніж друге порогове значення.

Різні втілення будуть детально описані з посиланням на наступні креслення, в яких однакові посилальні позиції відповідають однаковим елементам, причому:

Фіг.1 є одним прикладом компресора зображень;

Фіг.2 є одним прикладом декомпресора зображень;

Фіг.3 показує приклад способу визначення того, чи блок поділеним;

Фіг.4A-4D показують приклади поділення блока;

Фіг.5 показує приклад двох сусідніх блоків в зображенні;

Фіг.6 показує приклад способу визначення того, чи треба використовувати фільтр блокового згладжування;

Фіг.7 показує інший приклад способу визначення того, чи треба використовувати фільтр блокового згладжування;

Фіг.8A-8D показують розстановки ABSDCT для блока 16×16;

Фіг.9A та 9B показують приклади присвоєних даних розміру блока;

Фіг.10A та 10B показують інші приклади присвоєних даних розміру блока;

Фіг.11 показує приклад способу генерації присвоєних даних розміру блока для ABSDCT;

Фіг.12 є таблицею, яка показує різні порогові значення дисперсій;

Фіг.13 показує деякі змінювані визначення; і

Фіг.14 показує приклад способу визначення того, чи треба використовувати фільтр блокового згладжування в системах, що використовують ABSDCT.

В системах стиснення, що використовують основане на блоках Дискретне Косинусне Перетворення (DCT), потік даних ділиться на блоки пікселів і до них застосовується дискретне косинусне перетворення. Ця основана на блоках обробка вносить блокові артефакти між межами блоків, оскільки перетворення не бере до уваги кореляцію між межами блоків і кожний блок кодується незалежно.

Звичайно в системах стиснення, що використовують DCT, розмір кожного блока даних є постійним. Проте, існують технології динамічного стиснення зображення, які можуть запропонувати значне стиснення зберігаючи якість сигналів зображення при використанні блоків з адаптивним розміром і підблоків із закодованими за допомогою DCT даними. Такі технології будуть називатися DCT із змінним розміром блока. Одним прикладом DCT із змінним розміром блока є дискретне косинусне перетворення з адаптивним розміром блока (ABSDCT), розкриті в заявці США №5,021,891, названій «Спосіб та Система Стиснення Зображення з Адаптивним Розміром Блока». DCT технології також розкриті в заявці США №5,107,345, озаглавленій «Спосіб та Система

Стиснення Зображення з Адаптивним Розміром Блока», і використання ABSDCT технології в поєднанні з технологією Дискретного Перетворення Древа Квадрантів обговорюється в заявці США №5,452,104, названій «Спосіб та Система Стиснення Зображення з Адаптивним Розміром Блока». Адаптивні розміри блоків вибираються, щоб використати надмірність інформації в кадрі даних зображення. ABSDCT буде описано нижче детальніше.

Втілення, описані нижче, зменшують артефакти, допускаючи простий та ефективний спосіб блокового згладжування, який може бути легко реалізований в системах стиснення, що використовують основане на блоках DCT. Втілення особливо ефективні в DCT із змінним розміром блока.

У подальшому описі подані конкретні подробиці, щоб забезпечити всебічне розуміння втілень. Проте, фахівці в даній галузі техніки зрозуміють, що втілення можуть бути виконані без цих конкретних подробиць. Наприклад, в блок-схемах можуть бути показані схеми для того, щоб не затемняти втілення непотрібними подробицями. В інших прикладах добре відомі схеми, структури і технології можуть бути детально показані, щоб не затемняти втілення.

Також зазначимо, що втілення можуть бути описані як процес, який зображений у вигляді блок-схеми. Хоча блок-схема може описувати дії як послідовний процес, багато дій можуть виконуватися паралельно або одночасно. Додатково, послідовність дій може бути змінена. Процес завершується, коли дії завершені. Процес може відповідати способу, функції, процедурі, стандартній підпрограмі, підпрограмі тощо. Коли процес відповідає функції, його зупинка відповідає поверненню функції до викликаючої функції або головної функції.

Фіг.1 показує приклад компресора 100 зображень, а Фіг.2 показує приклад декомпресора 200 зображень, який симетричний компресору 100 зображень. Компресор 100 зображень містить модуль ПО DCT із змінним розміром блока (VBSDCT), модуль 120 квантування і модуль 130 кодування із змінною довжиною (VLC). Декомпресор 200 зображень містить модуль 210 декодування із змінною довжиною (VLD), модуль 220 зворотного квантування та модуль 230 зворотного VBSDCT. Декомпресор 200 зображень додатково містить модуль 240 фільтра блокового згладжування, щоб фільтрувати краї блока, якщо необхідно, і процесор 250, щоб керувати модулем 240 фільтра блокового згладжування.

Звичайно, потік даних, що вводяться в компресор 100 зображень, складається з кадрів зображення. Кадр зображення може, загалом, бути розділений на кристали, причому кристал може бути розділений на блоки даних, а блоки даних можуть бути розділені на пікселі, які є найменшими одиницями зображення. Кожний кадр зображення включає ціле число кристалів, і кожний кристал зображення представляє інформацію про зображення для набору з n послідовних рядків розгортки, таких як 16 послідовних рядків розгортки. У такому випадку, кожний блок даних відповідає 16×16 блоку пікселів в зображенні кадру. Також, кадр може бути поділений на парне і непарне число кристалів, таким чином, утворюючи парну половину кадру і непарну половину кадру. Крім того, піксель зображення звичайно може бути представлений в системі компонентів Червоного, Зеленого та Блакитного (RGB) кольорів. Проте, оскільки людське око більш чутливе до змін яскравості і менш чутливе до змін кольоровості, при стисненні відеозображення звичайно використовується YCbCr колірний простір, щоб представити пікселі зображення. YCbCr колірний простір є лінійним перетворенням RGB компонент, де Y є компонентою кольоровості, а Cb та Cr є компонентами кольору. Якщо кадр поділений на парні/непарні кадри, кадр зображення буде складатися з трьох парних половин кадру і трьох непарних половин кадру, відповідних компонентам Y, Cb та Cr.

У вищенаведеному описі кристал може представляти набір послідовних рядків розгортки, відмінних від 16 послідовних рядків розгортки. Також, блок даних може бути $n \times m$ блоком, де n не дорівнює m , якщо блок може бути поділений. Крім того, може бути використаний інший колірний простір з таким самим або іншим числом колірних компонент, щоб представити піксель зображення. Проте, нижче, з метою пояснення, будуть використані розмір блока з 16×16 пікселів та YCbCr колірний простір.

Посилаючись знову на Фіг.1: модуль 110 VBSDCT перетворює цифрову інформацію про зображення з просторової області в частотну область і генерує DCT коефіцієнти з відповідною інформацією про присвоєний розмір блока (BSA). Модуль 110 VBSDCT ділить та обробляє цифрову інформацію про зображення в блоках і, якщо необхідно, в підблоках. Фіг.3 показує один спосіб 300 визначення того, чи є блок поділеним. У способі 300 одержують (310) значення дисперсії DCT блока. Дисперсію потім порівнюють з набором TBS порогових значень для розміру блока (320). Визначають, чи є блок поділеним, основуючись на порівнянні значення дисперсії з пороговим значенням TBS. Тобто, якщо значення дисперсії більше, ніж TBS, тоді блок є поділеним (330 та 340). В іншому випадку, блок не є поділеним (350). Тут аналогічний спосіб може бути використаний для підблоків, щоб визначити, чи є підблок поділеним. У такому випадку, дисперсія порівнюється з набором порогових значень для розміру підблока.

BSA інформація показує як поділений блок, якщо він є поділеним. Наприклад, BSA інформація може показати, що 16×16 блок поділений на чотири 8×8 блоки, як на Фіг.4A, або, можливо, що 8×8 блок поділений на чотири 4×4 блоки, як на Фіг.4B. В інших системах BSA інформація може показати, що 16×16 блок поділений на чотири 8×8 блоки, як показано на Фіг.4C. Як показано, один з 8×8 блоків Фіг.4C поділений на чотири 4×4 блоки і додатково на чотири 2×2 блоки, як показано також на Фіг.4C, основуючись на конфігурації системи і/або необхідності. Модуль 120 квантування потім квантує DCT коефіцієнти, і VLC 130 стискає квантовані DCT коефіцієнти, використовуючи технологію кодування із змінною довжиною.

У декомпресорі 200 зображень модуль 210 VLD розгортає стиснуту інформацію про зображення, модуль 220 квантування квантує в зворотному порядку розгорнуту інформацію про зображення і модуль 230 зворотного VBSDCT перетворює квантовану інформацію про зображення з просторової області в частотну область, використовуючи інформацію про присвоєний розмір блока. Процесор 250 визначає, чи є два блоки зображення сусідніми блоками, як показано на Фіг.5. Процесор 250 потім визначає, чи" необхідне блокове згладжування для двох сусідніх блоків, основуючись на кількості крайової активності блока або зайнятості. Якщо вважається, що блокове згладжування необхідне, один або більше загальних крайових пікселів двох сусідніх блоків фільтруються за допомогою модуля 240 фільтра блокового згладжування. Після подальшої

обробки інформація про зображення виводиться на дисплей і/або зберігається для представлення.

Фіг.6 показує спосіб 600 обробки зображень, стиснутих при використанні основного на блоках стиснення. У способі 600 визначають, чи є обидва сусідніх блоки поділеними (610). Тут BSA інформація може бути використана, щоб визначити, чи є два сусідніх блоки поділеними. Якщо обидва сусідніх блоки не є поділеними, тоді на одному або більшій кількості крайових пікселів двох сусідніх блоків використовується фільтр блокового згладжування (620).

Фіг.7 показує інший спосіб 700 обробки зображень, стиснутих при використанні основного на блоках стиснення. У способі 700 визначають, чи є обидва сусідніх блоки поділеними (710). Якщо визначено, що два сусідніх блоки обидва є поділеними, фільтр блокового згладжування не використовується. Проте, якщо обидва сусідніх блоки не є поділеними, тобто щонайменше один з двох - сусідніх блоків не є поділеним, додатково визначають, чи є один з двох сусідніх блоків поділеним (720). Якщо один з двох сусідніх блоків є поділеним, тоді на одному або більшій кількості пікселів двох сусідніх блоків використовується фільтр блокового згладжування (730). Тут перший фільтр блокового згладжування може бути двоточковим усереднюючим фільтром, що використовується на двох крайових пікселях двох сусідніх блоків. Якщо жоден з двох сусідніх блоків не є поділеним, тоді на одному або більшій кількості пікселів двох сусідніх блоків використовується другий фільтр блокового згладжування.

Конкретніше, одержують значення відмінності між одним або більше відповідними крайовими пікселями двох сусідніх блоків (740). Значення відмінності представляє дисперсію вздовж границь блока, і воно може бути одержане і/або виведене, з використанням різних технологій. Може бути одержана проста відмінність першого порядку між двома відповідними сусідніми блоками. В інших втіленнях може бути одержана і використана відмінність другого порядку. Одне або більше значень відмінності порівнюють з пороговим значенням TD (750). Основуючись на порівнянні одного або більше значень відмінності з пороговим значенням TD, вибирають другий фільтр блокового згладжування (760).

Порогове значення TD звичайно залежить від яскравості і може бути заздалегідь встановлене для різних систем і/або різних типів зображень. В одному втіленні як порогове значення може бути використане середнє значення з середніх значень двох сусідніх блоків. Альтернативно, порогове значення TD може бути різницею середніх значень двох сусідніх блоків. Порогове значення TD може також бути оптимізоване, щоб мати справу із змінами інтенсивності в зображенні, використовуючи масштабний коефіцієнт α , який пропорційний ступеню контрастності таким чином:

$$\alpha = (\mu_c - \mu_n) / \mu_n$$

де μ_c є середнім значенням поточного блока, а μ_n є середнім значенням блока, що містить крайові пікселі, що використовуються при одержанні значень відмінності.

Значення α лежить в діапазоні від 0 до 1.

Крім того, в одному втіленні одержують значення відмінності між трьома крайовими пікселями двох сусідніх блоків і порівнюють їх з пороговим значенням TD. Якщо щонайменше два із значень відмінності більше, ніж TD, вибирається фільтр Гауса. Тобто, якщо три з трьох значень відмінності більше, ніж TD, використовується шеститочковий фільтр Гауса на шести крайових пікселях двох сусідніх блоків. Якщо два з трьох значень відмінності більше, ніж TD, тоді використовується чотириточковий фільтр Гауса на чотирьох крайових пікселях двох сусідніх блоків. Якщо одне з трьох значень відмінності більше, ніж TD, тоді використовується усереднюючий фільтр на двох крайових пікселях двох сусідніх блоків.

Посилаючись знову на Фіг.2: процесор 250, отже, може визначити, чи необхідне блокове згладжування. Як обговорювалося вище, процесор 250 може також вибрати, коли дозволяє система, різні фільтри блокового згладжування залежно від характеристик сусідніх блоків. Таким чином, модуль 240 фільтра блокового згладжування містить один або більше типів фільтрів, таких як усереднюючий фільтр і/або фільтр Гауса, але не обмежується ними.

Крім того, як обговорювалося вище, модуль 110 VBSDCT може бути реалізований за допомогою ABSDCT. Технології стиснення, що використовують ABSDCT, будуть описані нижче, використовуючи розмір блока, що дорівнює 16×16 пікселів. Звичайно, кожна з компонент яскравості та кольоровості передається на чергування блоків (не показано). В одному втіленні, як показано на Фігурах від 8A до 8D, 16×16 блок представлений для чергування блоків, яке упорядковує зразки зображень в 16×16 блоках, щоб створити блоки і складові підблоки даних для DCT аналізу. Одне 16×16 DCT застосовується для першої розстановки, чотири 8×8 DCT застосовуються для другої розстановки, 16 4×4 DCT застосовуються для третьої розстановки, і 64 2×2 DCT застосовуються для четвертої розстановки. DCT дія зменшує просторову надмірність, властиву джерелу зображення. Після виконання DCT велика частина енергії сигналу зображення прагне до зосередження в декількох DCT коефіцієнтах.

Для 16×16 блока і кожного підблока аналізують перетворені коефіцієнти, щоб визначити число бітів, необхідних для кодування блока або підблока. Потім вибирають блок або комбінацію підблоків, яка вимагає щонайменше декількох бітів для кодування, щоб представити сегмент зображення. Наприклад, можуть бути вибрані два 8×8 підблоки, шість 4×4 підблоків і вісім 2×2 підблоків, щоб представити сегмент зображення. Вибраний блок або комбінація підблоків потім належним чином упорядковуються.

Перетворені коефіцієнти аналізують і вибирають блок або комбінацію підблоків, щоб представити сегмент зображення. Таким чином, генерують інформацію про присвоєний розмір блока, яка представляє присвоєний розмір блока в $n \times n$ блоці. Для 16×16 блока даних ABSDCT технологія генерує дані, відомі як PQR інформація, яка представляє присвоєний розмір блока в 16×16 блоці. PQR інформація є даними із змінною бітовою шириною, і вона описує якою мірою поділяється 16×16 блок. R-біт PQR поля представляє, чи поділений 16×16 блок на чотири 8×8 блоки. Як показано на Фіг.9A, якщо R-біт дорівнює «0», блок залишається неподіленим. У цьому випадку додаткова PQR інформація не потрібна, і PQR поле має тільки 1 біт. Якщо R-біт дорівнює «1», тоді 16×16 блок поділений на чотири 8×8 блоки, як показано на Фіг.9B, і, в PQR полі будуть існувати щонайменше чотири додаткових біти.

Додаткові чотири біти називаються «Q»-інформація. Кожний Q-біт означає підрозділ 8×8 блока на чотири 4×4 блоки. Для кожного Q-біта, існують ще чотири біти «P», щоб вказати, якщо будь-який з 4×4 блоків поділений на 2×2. Отже, довжина PQR даних може бути від 1 до 21 біта, залежно від присвоєного розміру блока в 16×16 блоці. Якщо кожний 8×8 блок є поділений, тоді PQR інформація буде мати довжину, що дорівнює 21 бітам. Фіг.10A-B показують деякі приклади 16×16 блоків з відповідними PQR даними.

Отже, кожний блок може бути поділений на підблоки розмірів 8×8, 4×4 і/або 2×2 залежно від критерію присвоєвання. Критерієм підрозділу n×n блока є дисперсія блока:

$$\text{Block_variance} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x^2(m,n) - \left[\frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \right]^2$$

Блок пікселів розміром n×n буде поділений на 4n/2×n/2 підблоки, якщо дисперсія блока n×n блока перевищує визначене порогове значення. Тут, середнє по блоках може мати значення в діапазоні (0, 1023) для 10-бітового зображення. Таким чином, зображення ділиться на 12 елементів, і для кожного елемента для кожної компоненти кольору використовується набір порогових значень. Також порогові значення можуть бути визначені ґрунруючись на статистиці, зібраній з деякої кількості кадрів зображень різних типів. Звичайний набір порогових значень показаний на Фіг.12A-C.

Фіг.11 ілюструє зразковий спосіб 1100 для генерації PQR інформації для показаного 16×16 блока. Для кожного блока одержують (1110) середнє значення і дисперсію V16. Дисперсію V16 порівнюють з відповідним пороговим значенням T16 для відповідного середнього значення (1115). Якщо дисперсія V16 не перевищує порогове значення T16, R значення PQR даних встановлюють на 0, і спосіб завершується (1120). В іншому випадку, R значення встановлюють на 1 (1125). Потім одержують дисперсію V8(i), {i=1...4}, для кожного з чотирьох 8×8 підблоків від 0 до 3, як показано на Фіг.9B, і кожну дисперсію V8(i) порівнюють з відповідним пороговим значенням T16, щоб визначити Q значення для PQR даних (від 1130 до 1140). Якщо дисперсія V8(i) не більше, ніж порогове значення T8, відповідне Q(i) встановлюють на 0 (1145). В іншому випадку, Q(i) значення встановлюють на 1 (1150). Потім одержують дисперсію V4(j), {j=1...4}, для кожного з чотирьох 4×4 підблоків кожного 8×8 блока, для якого Q(i) встановлено на 1, і кожну дисперсію V4(j) порівнюють з відповідним пороговим значенням T4, щоб визначити P значення для PQR даних (від 1155 до 1165). Якщо дисперсія V4(j) не більше, ніж порогове значення T4, відповідне Q(j) встановлюють на 0 (1170). В іншому випадку, Q(j) значення встановлюють на 1 (1175).

Таким чином, PQR інформація може бути згенерована і використана для блокового згладжування зображень в декомпресорі, такому як декомпресор 200 зображень. PQR інформація використовується, щоб визначити крайовий зміст зображення. Чим більше крайова інформація в блоці, тим менше розмір блока і довше PQR код. Фіг.14 показує зразковий спосіб 1400 для обробки зображень, стиснутих при використанні ABSDCT, а Фіг.15 показує змінні визначення, що використовуються в способі 1400.

При визначенні того, чи треба використати фільтр блокового згладжування для двох сусідніх блоків, одержують PQR інформацію для кожного блока (1410). Якщо обидва PQR біти більше, ніж 5 бітів (1415), спосіб завершується. Тобто, визначено, що обидва блоки є поділеними і що вони містять достатню крайову інформацію. В іншому випадку, якщо один з PQR бітів більше, ніж 5 бітів, на {x1, y1} використовується двоточковий усереднюючий фільтр (1420 та 1425). Якщо жоден з PQR бітів не більше, ніж 5 бітів, тоді одержують (1430) значення d1, d2 та d3 відмінності. Якщо d1, d2 та d3 більше, ніж порогове значення TD, тоді на {x1, x2, x3, y1, y2, y3} використовується шеститочковий фільтр Гауса (1435 та 1440). Якщо d1 та d2 більше, ніж порогове значення TD, тоді на {x1, x2, y1, y2} використовується чотириточковий фільтр Гауса (1445 та 1450). Якщо d1 більше, ніж порогове значення TD, тоді на {x1, y1} використовується двоточковий усереднюючий фільтр (1455 та 1460).

В способі 1400 втілення не обмежується усереднюючим фільтром і/або фільтром Гауса. Замість усереднюючого фільтра і/або фільтра Гауса можуть використовуватися інші фільтри.

Як показано, модуль фільтра блокового згладжування може бути легко реалізований в декомпресорі. Отже, артефакти можуть бути значно ослаблені, і може бути поліпшена візуальна якість зображення. Зазначимо, що хоча показано, що модуль 240 фільтра блокового згладжування реалізований окремо від модуля 230 зворотного VBSDCT і від процесора 250, одна з цих компонент або комбінація модуля 240 фільтра блокового згладжування, модуля 230 зворотного VBSDCT і процесора 250 можуть бути реалізовані разом.

Також, втілення можуть бути реалізовані за допомогою апаратного забезпечення, програмного апаратного забезпечення, проміжного програмного забезпечення, мікрокоманд або будь-якої їх комбінації. При реалізації в програмному забезпеченні, програмно-апаратному забезпеченні, проміжному програмному забезпеченні або мікрокомандах, елементи втілення є програмним кодом або сегментами коду, щоб виконати необхідні задачі, і можуть бути збережені в машинозчитуваному середовищі (не показано). Сегмент коду може представляти процедуру, функцію, підпрограму, програму, стандартну програму, стандартну підпрограму, модуль, пакет програмного забезпечення, клас або будь-яку комбінацію команд, структур даних або оператора програми. Сегмент коду може бути об'єднаний з іншим сегментом коду або схемою апаратного забезпечення за допомогою передачі і/або прийому інформації, даних, аргументів, параметрів або вмісту пам'яті. Інформація, аргументи, параметри, дані тощо можуть бути передані, направлені або відправлені через будь-який відповідний засіб, включаючи спільне використання пам'яті, передачу повідомлень, передачу маркера, мережну передачу тощо. Також, машинозчитуване середовище може бути реалізоване в промисловому виробі для використання в комп'ютерній системі і може мати засоби у вигляді машинозчитуваного коду, реалізовані в машинозчитуваному середовищі.

Додатково, хоча втілення були описані, використовуючи DCT із змінним розміром блока, технологія блокового згладжування, як описано вище, може бути також реалізована в DCT з постійним розміром блока. У таких випадках BSA інформація генерується, але не використовується для дійсного DCT. Замість цього BSA інформація використовується в декомпресорі зображень, щоб визначити, чи необхідне блокове згладжування

для двох сусідніх блоків.

Потрібно зазначити, що вищезазначені втілення є тільки прикладами і не повинні тлумачитися як обмеження винаходу. Опис втілень призначений для ілюстрації і не обмежує обсяг формули винаходу. По суті, дані указання можуть бути легко застосовані до інших типів пристроїв і багатьох альтернатив, модифікацій та варіацій, які стануть очевидними для фахівців в даній галузі техніки.

Перелік посилальних позицій

Фіг. 1-2

110 VBSDCT модуль

120 Модуль квантування

130,210 VLC модуль

220 Модуль зворотного квантування

230 Зворотний VBSDCT модуль

240 Модуль фільтра блокового згладжування

250 Процесор Фіг.2

310 Одержати значення дисперсії блока

320 Порівняти значення дисперсії з відповідним пороговим значенням T_{BS}

330 Значення дисперсії $> T_{BS}$

340 Блок поділений

350 Блок не поділений Фіг.6

610 Визначити, чи є два сусідніх блоки поділеними

620 Використати фільтр розблокування, якщо обидва сусідніх блоки не є поділеними

Фіг.7

710 Обидва сусідніх блоки поділені

720 Один з сусідніх блоків поділений

730 Фільтр

740 Одержати одне або більше значень відмінності між одним або більшою кількістю крайових пікселів сусідніх блоків

750 Порівняти одне або більше значень відмінності з другим пороговим значенням

760 Фільтр, оснований на порівнянні Фіг.11

1110 Одержати дисперсію V16 блока

1135 Обчислити дисперсію i-того 8×8 блока

1160 Обчислити дисперсію j-того 4×4 блока

Фіг.14

1410 Одержати PQR інформацію

1415 Обидва $PQR > 5$ біт

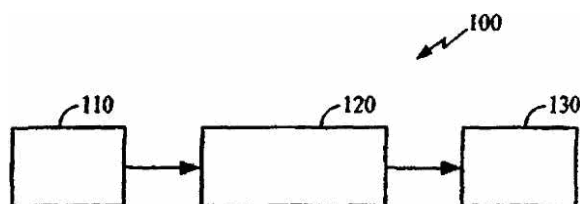
1420 Одне з $PQR > 5$ біт

1430 Одержати $d1, d2, d3$

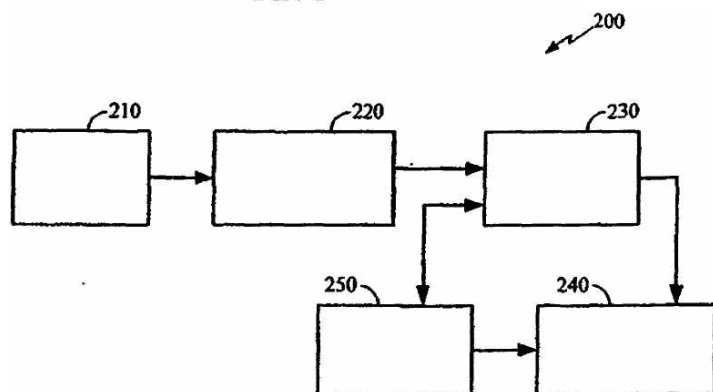
1440 Шеститочковий фільтр Гауса $x_i, y_i, i=1, 2, 3$ вздовж краю

1450 Чотириточковий фільтр Гауса $x_i, y_i, i=1, 2$ вздовж краю

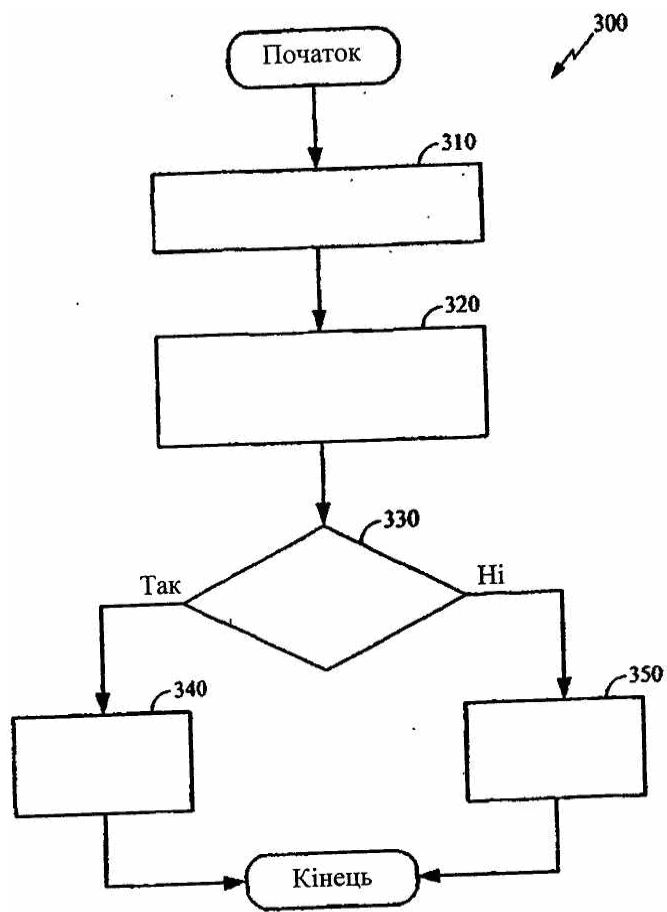
1460 Двоточковий фільтр Гауса $x_i, y_i, i=1$ вздовж краю



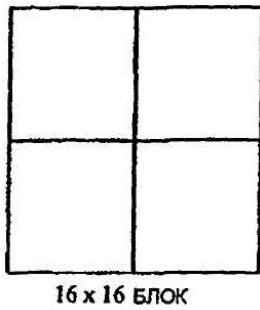
Фіг. 1



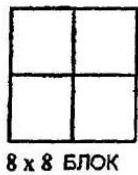
Фіг. 2



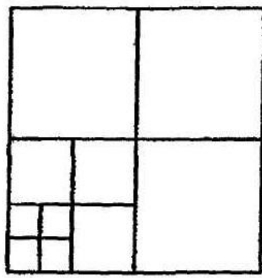
Фіг. 3



Фіг. 4А

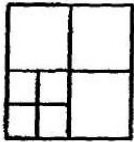


Фіг. 4В



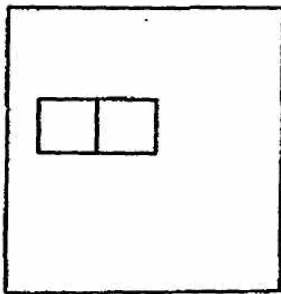
16 x 16 БЛОК

Фіг. 4С

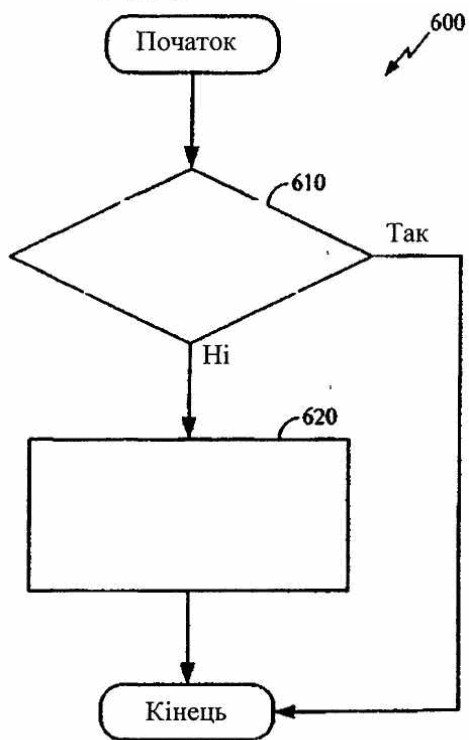


8 x 8 БЛОК

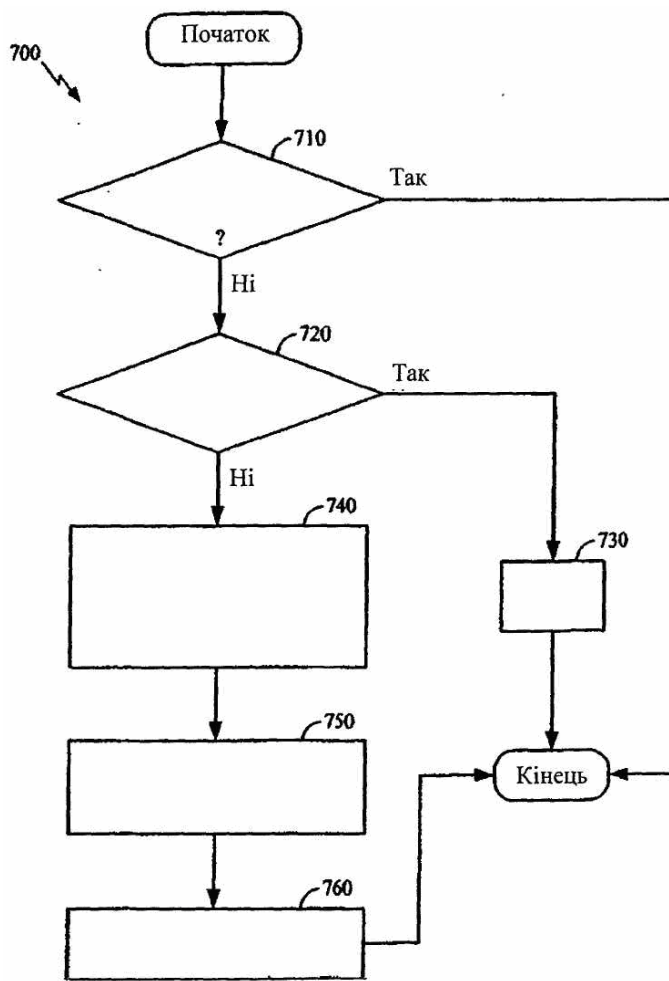
Фіг. 4D



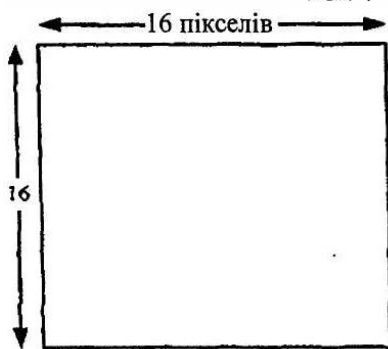
Фіг. 5



Фіг. 6

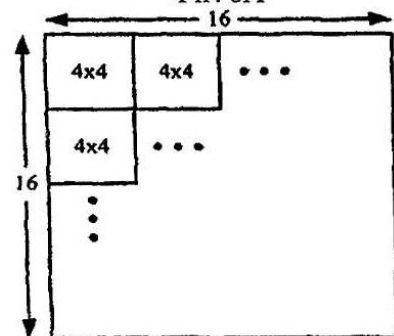


Фіг. 7



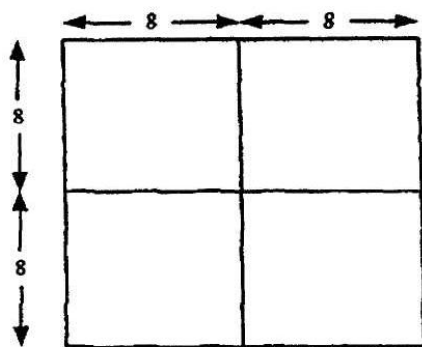
Перша розстановка

Фіг. 8A



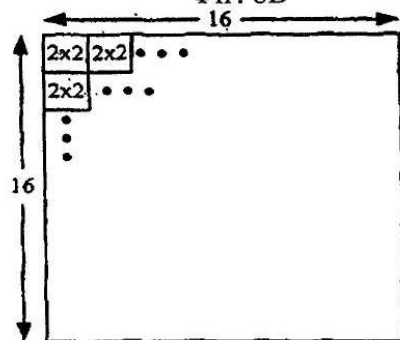
Третя розстановка

Фіг. 8C



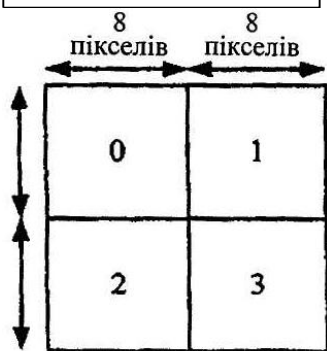
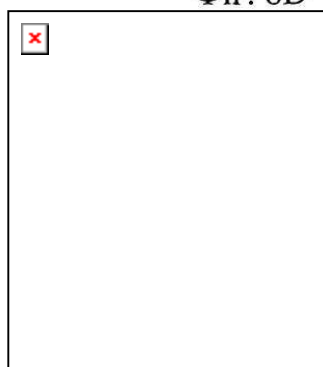
Друга розстановка

Фіг. 8B



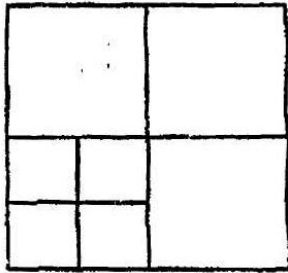
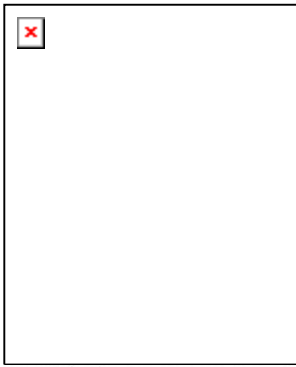
Четверта розстановка

Фіг. 8D



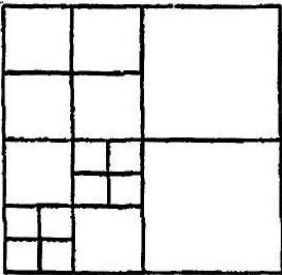
R = 1

Фіг. 9B



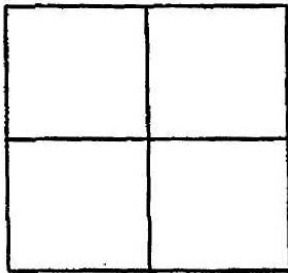
PQR = 0000 0100 1

Фиг. 10B



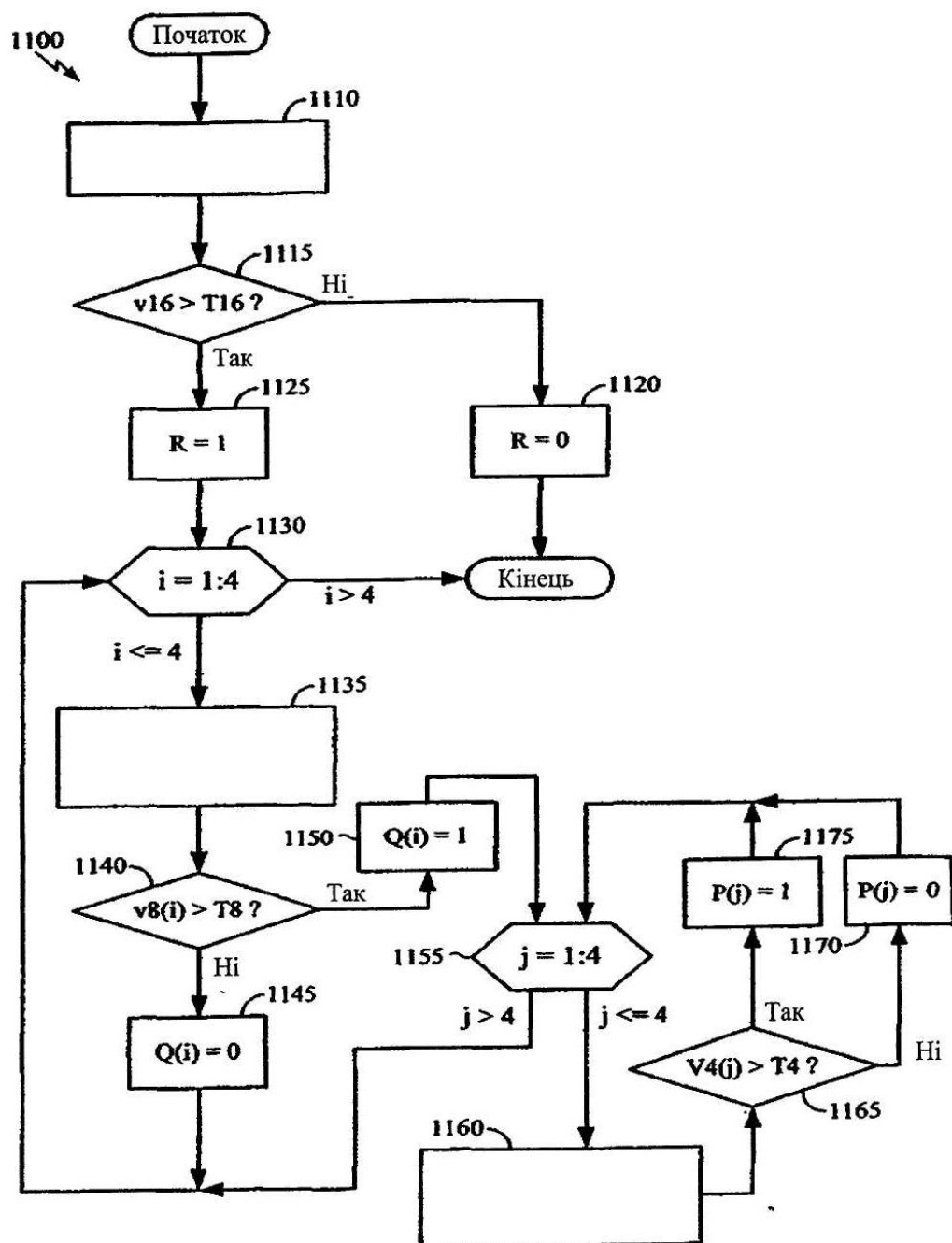
PQR = 0110 0000 0101 1

Фиг. 10C



PQR = 0000 1

Фиг. 10D



Фіг. 11

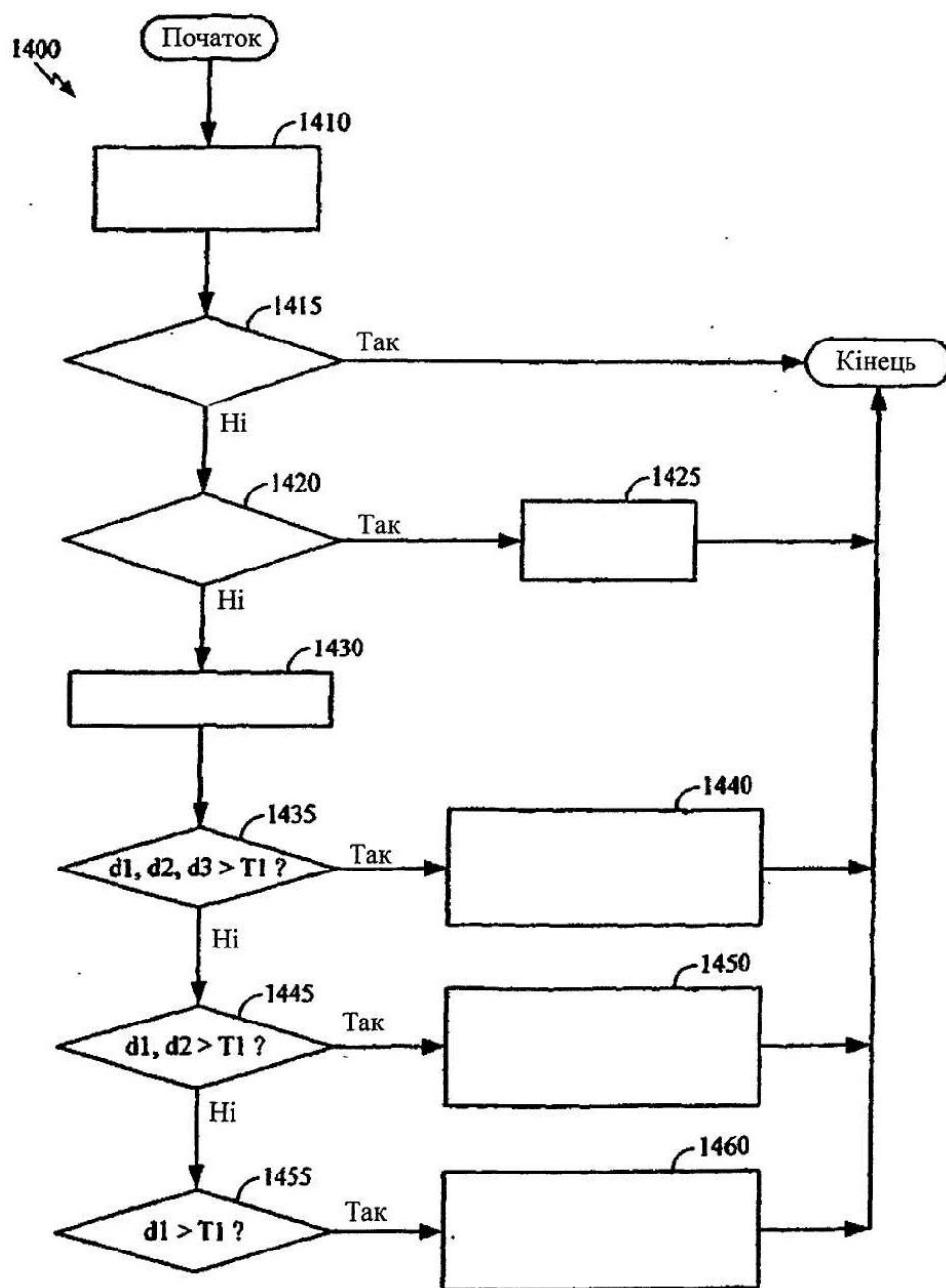
		Порогові значення дисперсії (16x16,8x8,4x4)		
Середнє значення		Y	Cb	Cr
0-70	T ₁₆	11	1118	2724
	T ₈	5	2627	1680
	T ₄	4	1266	457
71-100	T ₁₆	50	1388	629
	T ₈	71	446	2562
	T ₄	58	128	810
101-150	T ₁₆	344	5405	7231
	T ₈	1037	4741	445
	T ₄	553	2044	982
151-200	T ₁₆	1531	1000	642
	T ₈	5278	307	690
	T ₄	2082	211	180
201-300	T ₁₆	3644	12796	6039
	T ₈	13679	15389	5084
	T ₄	5499	18806	3874
301-400	T ₁₆	6439	17226	6374
	T ₈	25900	23978	5084
	T ₄	9319	20727	3422
401-500	T ₁₆	6918	2318	2450
	T ₈	27273	1299	1602
	T ₄	8786	282	775
501-600	T ₁₆	2000	2051	1271
	T ₈	19743	1076	661
	T ₄	5226	2211	230
601-700	T ₁₆	1080	7980	15062
	T ₈	5125	1631	1911
	T ₄	1015	1631	1911
701-800	T ₁₆	2725	35108	30063
	T ₈	14750	8415	19281
	T ₄	3731	12756	10165
801-900	T ₁₆	2171	44079	33059
	T ₈	6127	34443	2698
	T ₄	1940	15933	7407
901-1023	T ₁₆	416	2726	1546
	T ₈	1400	1142	392
	T ₄	550	653	85

Фиг. 12

x3	x2	x1	y1	y2	y3

$$d_i = x_i - y_i, i = 1, 2, 3$$

Фиг. 13



Фіг. 14