

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК 004.942

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Візуалізація об’єму в медицині”

(назва згідно з наказом ректора)

Спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

БР.ІПЗ - 22.00.00.000

(позначення)

Студент

ІПЗ-42 _____ /**Андрій ЖУРАВЛЬОВ** /
(шифр групи) (підпис) (дата)(розшифровка підпису)

Науковий керівник

к.т.н., а. к. ПСТ _____ /**Максим ТКАЧЕНКО** /
(посада) (підпис) (дата) (розшифровка підпису)

Консультант з питань нормоконтролю

Фахівець _____ /**Тамара ЧАПОВСЬКА** /

Допускається до захисту

Завідувач кафедри

д.т.н., доц. _____ /**Олексій БИЧКОВ** /
(посада) (підпис) (дата) (розшифровка підпису)

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра програмних систем і технологій
Освітньо-кваліфікаційний рівень бакалавр
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

_____ (Олексій БИЧКОВ)
(підпис) (прізвище та ініціали)

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Журавльову Андрію Єгоровичу

(прізвище, ім'я, по-батькові)

1. Тема бакалаврської роботи “Візуалізація об'єму в медицині”

керівник проекту (роботи) Ткаченко Максим Васильович, к.т.н., а.к. ПСТ

затверджені наказом вищого навчального закладу від “ 11 ” листопада 2020 р. № 6

2. Строк подання студентом роботи 31 травня 2021

3. Вихідні дані до проекту (роботи) програмне забезпечення для візуалізації комп'ютерних томографічних зображень

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз формату raw data

2. Аналіз архітектури графічного процесору, системи програмування GPU

3. Аналіз алгоритмів рендерингу об'ємних 3D моделей

4. Розробка програмного коду для візуалізації комп'ютерних томографічних зображень у форматі raw data

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. МРТ головного мозку (рис. 1.3.1)

2. Архітектура графічного процесора G80 (рис. 2.1.1)

3. Концепція SIMD (рис. 2.1.2)

4. Схема взаємодії центрального та графічного процесорів (рис. 2.2.1)

5. Принципи поділу пам'яті (рис. 2.3.1)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Ткаченко М.В.	Ткаченко М.В.	Ткаченко М.В.
2	Ткаченко М.В.	Ткаченко М.В.	Ткаченко М.В.
3	Ткаченко М.В.	Ткаченко М.В.	Ткаченко М.В.
3	Ткаченко М.В.	Ткаченко М.В.	Ткаченко М.В.

7. Дата видачі завдання _____ 13 жовтня 2019 р.

Керівник _____ (Максим ТКАЧЕНКО)

Завдання прийняв до виконання _____ (Андрій ЖУРАВЛЬОВ)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури	01.04.2021	виконано
2	Аналіз можливостей GPU, програмування шейдерів	10.04.2021	виконано
3	Підбір інструментів розробки ПЗ, їх вивчення	25.04.2021	виконано
4	Аналіз алгоритмів 3D візуалізації	14.05.2021	виконано
5	Програмна реалізація алгоритмів 3D візуалізації	20.05.2021	виконано

Студент – бакалавр _____ (Андрій ЖУРАВЛЬОВ)

Керівник роботи _____ (Максим ТКАЧЕНКО)

АНОТАЦІЯ

Випускна кваліфікаційна бакалаврська робота: 55 с., 12 рис., 1 додат., 15 джерел.

Тема: Візуалізація об'єму в медицині

Об'єкт дослідження: дані комп'ютерної томографії raw data

Мета роботи: розробка програмного забезпечення для роботи та візуалізації томографічних зображень у форматі raw data

Предмет дослідження: технології візуалізації даних raw data.

Результати дослідження:

Досліджено технології та алгоритми роботи з даними комп'ютерної томографії raw data, проаналізовано можливості рендерингу об'ємних 3D моделей частин тіла у форматі raw data. Розроблене ПЗ дозволяє відображувати будь-які томографічні зображення у форматі raw data, що дає змогу без оперативного втручання проаналізувати моделі тіла та діагностувати захворювання. ПЗ також можна використовувати в цілях навчання медперсоналу.

Висновок

В результаті досліджень було сформовано алгоритми рендерингу об'ємних 3D моделей частин тіла у форматі raw data, розроблено ПЗ для зчитування та відображення будь-яких томографічних зображень.

КОМП'ЮТЕРНА ТОМОГРАФІЯ, ТОМОГРАФІЧНІ ЗОБРАЖЕННЯ, ШЕЙДЕРИ,
RAYMARCHING, РЕНДЕРИНГ, RAW DATA

ABSTRACT

Final qualification Bachelor's work: 55 p., 12 images., 1 add., 15 sources.

Topic: Volume visualization in medicine

Object of research: computed tomography raw data

Objective: develop software for working and visualizing tomographic images in raw data format

Subject of research: raw data visualization technologies.

Research results:

Technologies and algorithms for working with raw data computed tomography data are investigated, and the possibilities of rendering three-dimensional 3D models of body parts in raw data format are analyzed. The developed software allows you to display any tomographic images in raw data format, which allows you to analyze body models and diagnose diseases without surgery. The software can also be used for training medical staff.

Conclusion

As a result of the research, algorithms for rendering three-dimensional 3D models of body parts in raw data format were formed, and software was developed for reading and displaying any tomographic images.

COMPUTED TOMOGRAPHY, TOMOGRAPHIC IMAGES, SHADERS,
REIMARCHING, RENDERING, RAW DATA

АННОТАЦИЯ

Выпускная квалификационная бакалаврская работа: 55 с., 12 рис., 1 доп., 15 джерел.

Тема: Визуализация объема в медицине

Объект исследования: данные компьютерной томографии raw data

Цель работы: разработка программного обеспечения для работы и визуализации томографических изображений в формате raw data

Предмет исследования: технологии визуализации данных raw data.

Результаты исследования:

Исследованы технологии и алгоритмы работы с данными компьютерной томографии raw data, проанализированы возможности рендеринга объемных 3D моделей частей тела в формате raw data. Разработанное ПО позволяет отображать любые томографические изображения в формате raw data, что дает возможность без оперативного вмешательства проанализировать модели тела и диагностировать заболевание. ПО также можно использовать в целях обучения медперсонала.

Вывод

В результате исследований были сформированы алгоритмы рендеринга объемных 3D моделей частей тела в формате raw data, разработано ПО для считывания и отображения любых томографических изображений.

КОМПЬЮТЕРНАЯ ТОМОГРАФИЯ, ТОМОГРАФИЧЕСКИЕ ИЗОБРАЖЕНИЯ,
ШЕЙДЕРЫ, RAYMARCHING, РЕНДЕРИНГ, RAW DATA

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
РОЗДІЛ 1	
ПРОЦЕС МЕДИЧНОЇ ВІЗУАЛІЗАЦІЇ ТА ЇХ ТИПИ	13
1.1. Поняття медичної візуалізації	13
1.2. Радіографія	15
1.3. Магнітно-резонансна томографія	16
1.4. Ядерна медицина	19
1.5. Ультразвук	20
1.6. Еластографія	22
1.7. Фотоакустична візуалізація	22
1.8. Ехокардіографія	22
1.9. Висновки до розділу	23
РОЗДІЛ 2	
СТРУКТУРА ТА МОЖЛИВОСТІ ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ З ГРАФІЧНИМ ПРОЦЕСОРОМ	24
2.1. Архітектура графічного процесора	24
2.2. Взаємодія графічного та центрального процесорів	26
2.3. Ієрархія пам'яті, яка доступна центральному та графічному процесорам	28
2.4. Висновки до розділу	31
РОЗДІЛ 3	
ПОТОЧНО-ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ GPU	32
3.1. Розпаралелювання розрахунків	32
3.1.1. Розпаралелювання по задачам	33
3.1.2. Розпаралелювання за інструкціями	33
3.1.3. Розпаралелювання за даними	34

3.2. Переваги графічних процесорів при паралельних розрахунках	35
3.3. Принцип програмування SIMD на прикладі піксельного шейдера	38
3.4. Висновки до розділу	40
РОЗДІЛ 4	
ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПЗ	42
4.1. Інструменти розробки ПЗ	42
4.1.1. Ігровий рушій Unity	42
4.1.2. Мова програмування C#	43
4.1.3. Мова програмування шейдерів CG	44
4.2. Техніка Volume Rendering	44
4.3. Техніка Raymarching	45
4.4. Метод проекції максимальної інтенсивності	46
4.5. Isosurface rendering	48
4.6. Direct Volume Rendering	49
4.7. Висновки до розділу	50
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕКГ	-	електрокардіографія
MPT	-	магнітно-резонансна томографія
GPU	-	графічний процесор
КТ	-	комп'ютерна томографія
ООП	-	об'єктно-орієнтоване програмування
ОС	-	операційна система
ПЗ	-	програмне забезпечення
SIMD	-	Концепція програмування, яка полягає в потоковій обробці даних
ЕОМ	-	електронно-обчислювальна машина
API	-	набір підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення

ВСТУП

Візуалізація медичних даних займає важливе місце в сучасній медицині та біології. Наука не стоїть на місці, тому існує постійне зростання темпів досліджень. З кожним днем спостерігається зростання обсягу вироблених тривимірних даних, які потребують методів тривимірної візуалізації, обробки, реконструкції та аналізу.

Довгий час в медицині використовувалися двомірні методи візуалізації, які досі знаходять своє застосування на практиці. Однак після появи 3D томографії, яка пропонувала низьку вартість, високу роздільну здатність, менші дози радіації і більше якісний результат, стали все частіше використовуватися методи тривимірної візуалізації. Можливість перетворення декількох сотень двомірних зображень в інтерактивну тривимірну модель стало справжнім проривом. Можливо не тільки обертати, переміщати об'єкт, але і побачити окремий зріз, якщо це необхідно. Така модель дозволяє лікарям більш точно діагностувати захворювання і краще проводити планування лікування пацієнту.

Актуальність роботи

За допомогою методів медичної візуалізації можна без оперативного втручання відтворити внутрішні структури тіла людини, спостерігати за функцією деяких органів або тканин. Це дозволяє діагностувати захворювання, вжити необхідних заходів при плануванні лікування пацієнта без хірургічного втручання. Галузь застосування методів візуалізації широка: дослідження і розробка в області приладобудування, отримання зображень, біомедична інженерія, медична фізика та інформатика.

Мета і задачі дослідження

Метою бакалаврської роботи є розробка програмного забезпечення для роботи та візуалізації томографічних зображень у форматі raw data.

Програмне забезпечення має зчитувати дані томографічних 3D зображень у форматі raw data та відображувати моделі частин тіла згідно цих даних. Крім того

має бути реалізована функція slice rendering, тобто відображення 2D зображення на площину, задану користувачем.

Досягнення мети включало розв'язання таких **задач**:

- 1) Імпортування та зчитування raw data
- 2) Вивчення можливостей рендерингу на GPU за допомогою шейдерів
- 3) Аналіз алгоритмів рендерингу об'ємних 3D-моделей
- 4) Аналіз алгоритмів рендерингу на 2D площину
- 5) Вибір релевантного алгоритму рендерингу та обґрунтування його використання

Об'єктом дослідження є дані комп'ютерної томографії raw data.

Предметом дослідження є технології візуалізації даних raw data.

Методи дослідження

Для досягнення цілі – візуалізації комп'ютерних томографічних зображень використовуються методи Raymarching, Direct Volume Rendering, Isosurface Rendering, Transfer functions.

Наукова новизна отриманих результатів

Досліджено можливості візуалізації raw data зображень, спроектовано та програмно реалізовано алгоритми Raymarching, Direct Volume Rendering, Isosurface Rendering, Transfer functions для відображення 3-D томографічних зображень у ігровому рушії Unity.

Практичне значення одержаних результатів

Програмне забезпечення дозволяє відображувати будь-які томографічні зображення у форматі raw data, що дає змогу без оперативного втручання проаналізувати моделі тіла та діагностувати захворювання. ПЗ також можна використовувати в цілях навчання медперсоналу.

Особистий внесок студента

Основним результатом є:

1. Розроблене ПЗ для візуалізації томографічних зображень у форматі raw data
2. Проаналізовані алгоритми візуалізації томографічних зображень

Структура та обсяг роботи

Робота викладена на 68 сторінках друкованого тексту, який складається із вступу, трьох розділів, висновків, списку використаних джерел (23 найменування). Робота містить 12 таблиць, 32 рисунки та 3 додатки, обсягом 12 стор.

РОЗДІЛ 1

ПРОЦЕС МЕДИЧНОЇ ВІЗУАЛІЗАЦІЇ ТА ЇХ ТИПИ

1.1. Поняття медичної візуалізації

Медична візуалізація - це метод і процес створення візуальних уявлень внутрішньої частини тіла для клінічного аналізу та медичного втручання, а також візуального представлення функцій деяких органів або тканин (фізіологія). Медична візуалізація спрямована на виявлення внутрішніх структур, прихованих під шкірою і кістками, а також для діагностики та лікування захворювань. Медична візуалізація також створює базу даних нормальної анатомії та фізіології, щоб зробити можливим виявлення відхилень. Хоча візуалізація віддалених органів і тканин може виконуватися за медичними показаннями, такі процедури зазвичай вважаються частиною патології замість медичної візуалізації.

Як дисципліна і в найширшому сенсі вона є частиною біологічної візуалізації і включає радіологію, в якій використовуються технології візуалізації рентгенівської радіографії, магнітно-резонансна томографія, ультразвук, ендоскопія, еластографія, тактильна візуалізація, термографія, медична фотографія, ядерна медицина методи функціональної візуалізації, такі як позитронно-емісійна томографія (ПЕТ) і одноразова фотонно-емісійна комп'ютерна томографія (ОФЕКТ).

Методи вимірювання та запису, які не призначені в першу чергу для отримання зображень, наприклад електроенцефалографія (ЕЕГ), магнітоенцефалографія (МЕГ), електрокардіографія (ЕКГ) та інші представляють інші технології, які виробляють дані, які можна представити у вигляді графіка залежності параметра від часу або карт, які містять дані про місця вимірювання. У обмеженому порівнянні ці технології можна розглядати як форми медичної візуалізації в іншій дисципліні.

Станом на 2010 рік у всьому світі було проведено 5 мільярдів досліджень з медичної візуалізації. Радіаційне опромінення від медичних зображень в 2006 році

склало близько 50% від загального іонізуючого випромінювання в Сполучених Штатах. Медичне обладнання для візуалізації проводиться з використанням технологій напівпровідникової промисловості, включаючи CMOS-інтегральні схеми мікросхеми, силові напівпровідникові пристрої, датчики такі як датчики зображення (зокрема , CMOS-датчики) і біосенсори, а також процесори , такі як мікроконтролери , мікропроцесори , Процесори цифрових сигналів, медіапроцесори і пристрої система на кристалі . Станом на 2015 рік Щорічні поставки чіпів для медичної візуалізації склали 46 мільйонів одиниць і 1,1 мільярда доларів.

Медична візуалізація часто сприймається як позначення набору методів, які неінвазивно створюють зображення внутрішньої частини тіла. У цьому обмеженому сенсі медична візуалізація може розглядатися як рішення математичних задач . Це означає, що причина (властивості живої тканини) виводиться з сліdstва (спостережуваного сигналу). У випадку медичного ультразвуку датчик складається з ультразвукових хвиль тиску та Ехо-сигналів, які проходять всередину тканини, щоб показати внутрішню структуру. У разі проекційної рентгенографії зонд використовує рентгенівське випромінювання, яке поглинається з різною швидкістю різними типами тканин, такими як кістки, м'язи і жир.

Термін «неінвазивний» використовується для позначення процедури, при якій в тіло пацієнта не вводиться ніякий інструмент, що характерно для більшості використовуваних методів візуалізації.

У клінічному контексті медична візуалізація "невидимого світла" зазвичай прирівнюється до радіології або «клінічна Візуалізація», і практикуючий лікар, відповідальний за інтерпретацію (а іноді і отримання) зображень, є радіологом . Медична візуалізація в "видимому світлі" включає цифрове відео або нерухомі зображення, які можна побачити без спеціального обладнання. Дерматологія та лікування ран-це два методи, в яких використовуються зображення у видимому світлі. Діагностична рентгенографія позначає технічні аспекти медичної візуалізації

і, зокрема, отримання медичних зображень. рентгенолог або технолог-радіолог зазвичай відповідає за отримання медичних зображень діагностичної якості, хоча деякі радіологічні втручання виконуються радіологами .

В якості галузі наукових досліджень медична візуалізація являє собою субдисципліну біомедичної інженерії, медичної фізики або медицини в залежності від контексту: дослідження і розробки в області приладобудування, отримання зображень (наприклад, радіографія), моделювання та кількісна оцінка зазвичай є прерогативою біомедичної інженерії, медичної фізики та інформатики ; Дослідження щодо застосування та інтерпретації медичних зображень зазвичай є прерогативою радіології та медичної дисципліни, що стосується стану здоров'я або галузі медицини (неврологія , кардіологія , Психіатрія, Психологія тощо) в стадії розслідування. Багато методів, розроблених для медичної візуалізації, також мають наукове та промислове застосування

1.2. Радіографія

У медицині використовуються два види рентгенографічних зображень. Проекційна рентгенографія і рентгеноскопія, остання корисна для катетеризації. Ці 2D-методи досі широко використовуються, незважаючи на розвиток 3D-томографії через низьку вартість, високу роздільну здатність і, в залежності від застосування, більш низьких доз опромінення в 2D-методиці. Цей метод візуалізації використовує широкий пучок рентгенівських променів для отримання зображення і є першим методом візуалізації, доступним в сучасній медицині.

Рентгеноскопія виробляє зображення внутрішніх структур тіла в реальному часі аналогічно рентгенографії, але використовує постійне введення рентгенівських променів при більш низькій потужності дози. Контрастні середовища, такі як барій, йод і повітря, використовуються для візуалізації роботи внутрішніх органів. Рентгеноскопія також використовується в процедурах під візуальним контролем, коли потрібен постійний зворотний зв'язок під час процедури. Приймач зображення

необхідний для перетворення випромінювання в зображення після того, як воно пройшло через цікаву область. Раніше це був флуоресцентний екран, який поступився місцем підсилувача зображення (ІА), який являв собою велику вакуумну трубку з приймальним кінцем, покритим йодидом цезію, і дзеркалом на протилежному кінці. Зрештою дзеркало було замінено телекамерою. Проекційні рентгенограми, більш відомі як рентгенівські знімки, часто використовуються для визначення типу I ступеня перелому, а також для виявлення патологічних змін в легенях. За допомогою рентгеноконтрастних речовин, таких як барій, їх також можна використовувати для візуалізації структури шлунка і кишечника - це може допомогти діагностувати виразки або певні типи раку товстої кишки

1.3. Магнітно-резонансна томографія

Прилад магнітно-резонансної томографії (сканер МРТ) або "ядерний магнітний резонанс (ЯМР)" сканер зображення", як він був спочатку відомий, використовує потужні магніти для поляризації і збудження ядер водню (тобто одиночних протонів) молекул води в тканинах людини, створюючи виявляється сигнал, який просторово закодовані, в результаті чого виходять зображення тіла. Апарат МРТ випромінює радіочастотний (РЧ) імпульс на резонансній частоті атомів водню на молекулах води. Радіочастотні антени («РЧ-котушки») посилюють імпульс в досліджувану область тіла. Радіочастотний імпульс поглинається протонами, викликаючи зміну їх напрямку по відношенню до первинного магнітного поля. Коли РЧ-імпульс вимкнений, протони "розслабляються" назад, вирівнюючись з первинним магнітом, і при цьому випромінюють радіохвилі. Це радіочастотне випромінювання атомів водню у воді реєструється і перетворюється в зображення. Резонансна частота обертового магнітного диполя (одним із прикладів якої є протони) називається ларморовскою частотою і визначається силою основного магнітного поля і хімічним оточенням досліджуваних ядер. МРТ використовує три електромагнітні поля: дуже сильне (зазвичай від 1,5 до 3 тесла) статичне магнітне

поле для поляризації ядер водню, зване первинним полем; поля градієнта, які можна змінювати для зміни в просторі і часі (близько 1 кГц) для просторового кодування, часто звані просто градієнтами; і просторово однорідне радіочастотне (RF) поле для маніпулювання ядрами водню з метою отримання вимірюваних сигналів, зібраних через RF-антену .

Подібно до КТ, традиційно МРТ створює двомірне зображення тонкого зрізу тіла і тому вважається методом томографічної візуалізації. Сучасні інструменти МРТ здатні створювати зображення у вигляді тривимірних блоків, які можна розглядати як узагальнення одношарової томографічної концепції. На відміну від КТ, МРТ не передбачає використання іонізуючого випромінювання і, отже, не пов'язана з такими ж небезпеками для здоров'я. Наприклад, оскільки МРТ використовується лише з початку 1980-х років, немає відомих довгострокових ефектів впливу сильних статичних полів (це предмет деяких дискусій; див. «Безпека» в МРТ), і тому немає обмежень на кількість сканувань, які людина може піддати, на відміну від X-гау та СТ. Однак існують чітко визначені ризики для здоров'я, пов'язані з нагріванням тканин від впливу радіочастотного поля і присутністю імплантованих пристроїв в організмі, наприклад, кардіостимуляторів. Ці ризики суворо контролюються як частина конструкції приладу і використовуваних протоколів сканування.

Оскільки КТ і МРТ чутливі до різних властивостей тканин, зовнішній вигляд зображень, отриманих за допомогою цих двох методів, помітно відрізняється. При КТ рентгенівські промені повинні блокуватися будь-якої щільною тканиною для створення зображення, тому якість зображення при перегляді м'яких тканин буде низьким. У МРТ, хоча можна використовувати будь-яке ядро з чистим ядерним спіном, протон атома водню залишається найбільш широко використовуваним, особливо в клінічних умовах, тому що він настільки поширений і дає великий

сигнал. Це ядро, присутнє в молекулах води, забезпечує чудовий контраст м'яких тканин, що досягається за допомогою МРТ, зображено на Рисунку 1.3.1



Рис. 1.3.1 МРТ головного мозку

Ряд різних імпульсних послідовностей може використовуватися для конкретної діагностичної візуалізації МРТ (багатопараметрична МРТ або mpMRI). Можна диференціювати характеристики тканин, комбінуючи дві або більше з наступних послідовностей зображень, в залежності від запитуваної інформації: T1-зважена (T1-MRI), T2-зважена (T2-MRI), дифузійно-зважена візуалізація (DWI-MRI).), динамічне посилення контрасту (DCE-MRI) і спектроскопія (MRI-S).

Наприклад, візуалізацію пухлин простати краще виконувати за допомогою T2-MRI і DWI-MRI, ніж тільки T2-зважену візуалізацію. Кількість застосувань mrMRI для виявлення захворювань у різних органах продовжує розширюватися, включаючи дослідження печінки, пухлини грудей, пухлини підшлункової залози, а також оцінку ефектів судинні агенти, що руйнують ракові пухлини.

1.4. Ядерна медицина

Ядерна медицина охоплює як діагностичну візуалізацію, так і лікування захворювань, і може також називатися молекулярною медициною або молекулярною візуалізацією і терапія. Ядерна медицина використовує певні властивості ізотопів і енергетичних частинок, що випускаються радіоактивним матеріалом, для діагностики або лікування різних патологій. На відміну від типової концепції анатомічної радіології, ядерна медицина дозволяє оцінювати фізіологію. Цей функціональний підхід до медичного обстеження має корисні застосування в більшості вузьких спеціальностей, особливо в онкології, неврології та кардіології. Гамма-камери і ПЕТ-сканери використовуються, наприклад, в сцинтиграфія, ОФЕКТ і ПЕТ для виявлення областей біологічної активності, які можуть бути пов'язані із захворюванням. Пацієнту вводять відносно короткоживучий ізотоп ізотоп, такий як Tc. Ізотопи часто переважно абсорбуються біологічно активною тканиною організму і можуть використовуватися для ідентифікації пухлин або точок перелому в кістці. Зображення отримуються після того, як колімовані фотони виявляються кристалом, який випромінює світловий сигнал, який, в свою чергу, посилюється і перетворюється в дані рахунку.

SPECT-це метод тривимірної томографії, який використовує дані гамма-камери з багатьох проекцій і може бути реконструйований в різних площинах. Гамма-камера з двома детекторними головками в поєднанні зі сканером комп'ютерної томографії, який забезпечує локалізацію функціональних даних SPECT, називається камерою SPECT-CT і показала свою корисність в просуванні

області молекулярної візуалізації. У більшості інших методів медичної візуалізації енергія проходить через тіло, а реакція або результат зчитуються детекторами. При візуалізації ОФЕКТ пацієнту вводять радіоактивний ізотоп, найчастіше талій ^{201}Tl , технецій $^{99\text{m}}\text{Tc}$, йод ^{123}I і галій ^{67}Ga . Радіоактивні гамма-промені випромінюються через тіло, оскільки має місце природний процес розпаду цих ізотопів. Емісія гамма-променів уловлюється детекторами, які оточують тіло. По суті, це означає, що джерелом радіоактивності тепер є людина, а не медичні пристрої візуалізації, такі як рентгенівські промені або комп'ютерна томографія.

Позитронно-емісійна томографія (ПЕТ) використовує виявлення збігів для зображення функціональних процесів. Короткоживучий ізотоп, що випромінює позитрони, такий як $^{\text{F}}18$, включається в органічну речовину, таку як глюкоза, створюючи $^{\text{F}}18$ -фтордезоксиглюкозу, яку можна використовувати в якості маркера метаболічного використання. Зображення розподілу активності по тілу можуть показати швидкозростаючі тканини, такі як пухлина, метастази або інфекція. Зображення ПЕТ можна розглядати в порівнянні зі сканованими зображеннями комп'ютерної томографії, щоб визначити анатомічний корелятор.

Фідуціарні маркери використовуються в широкому спектрі додатків медичної візуалізації. Зображення одного і того ж об'єкта, отримані за допомогою двох різних систем візуалізації, можуть бути корельовані (так звана реєстрація зображення) шляхом розміщення фідуціарного маркера в області, що відображається обома системами. У цьому випадку необхідно використовувати маркер, який видно на зображеннях, отриманих за допомогою обох методів візуалізації. За допомогою цього методу функціональна інформація з СПЕКТ або позитронно-емісійної томографії може бути пов'язана з анатомічною інформацією, наданою магнітно-резонансною томографією

1.5. Ультразвук

Медичне ультразвукове дослідження використовує високочастотний широкосмуговий доступ звукові хвилі в діапазоні мегагерц , які відображаються тканиною в різному ступені, створюючи (до 3D) зображення. Це зазвичай пов'язано з візуалізацією плоду у вагітних жінок. Однак використання ультразвуку набагато ширше. Інші важливі застосування включають візуалізацію органів черевної порожнини, серця, грудей, м'язів, сухожиль, артерій і вен. Хоча він може надавати менше анатомічних деталей, ніж такі методи, як КТ або МРТ, він має кілька переваг, які роблять його ідеальним у багатьох ситуаціях, зокрема, він вивчає функцію рухомих структур в реальному часі , не випускає іонізуючого випромінювання і містить спекл, який можна використовувати в еластографії . Ультразвук також використовується в якості популярного дослідницького інструменту для збору необроблених даних , які можуть бути доступні через інтерфейс ультразвукових досліджень, з метою визначення характеристик тканин і впровадження нових методів обробки зображень. Концепції ультразвуку відрізняються від інших методів медичної візуалізації тим, що в їх основі лежить передача і прийом звукових хвиль. Високочастотні звукові хвилі направляються в тканину і залежать від складу різних тканин; сигнал буде послаблюватися і повертатися через окремі інтервали. Шлях відбитих звукових хвиль в багатосаровій структурі може бути визначений вхідним акустичним імпедансом (ультразвукова звукова хвиля) і коефіцієнтами відображення і передачі відповідних структур. Він дуже безпечний у використанні і не викликає ніяких побічних ефектів. Це також відносно недорого і швидко у виконанні. Ультразвукові сканери можуть бути доставлені тяжкохворим пацієнтам у відділеннях інтенсивної терапії, що дозволяє уникнути небезпеки, що виникає при переміщенні пацієнта у відділення радіології. Отримане рухоме зображення в реальному часі можна використовувати для проведення процедур дренажу і біопсії. Можливості доплера на сучасних сканерах дозволяють оцінювати кровотік в артеріях і венах.

1.6. Еластографія

Еластографія-це відносно новий метод візуалізації, який відображає еластичні властивості м'яких тканин. Ця модальність з'явилася в останні два десятиліття. Еластографія корисна при медичній діагностиці, оскільки еластичність дозволяє відрізнити здорову тканину від нездорової для певних органів / новоутворень. Наприклад, ракові пухлини часто бувають більш твердими, ніж навколишні тканини, а хвора печінка жорсткіше, ніж здорова. Існує кілька методів еластографії, заснованих на використанні ультразвуку, магнітно-резонансної томографії і тактильної візуалізації. Широке клінічне використання ультразвукової еластографії є результатом впровадження технології в клінічні ультразвукові апарати. Основні галузі ультразвукової еластографії включають квазістатичну еластографію / візуалізацію деформації, візуалізацію пружності зсувної хвилі (SWEI), візуалізацію імпульсу сили акустичного випромінювання (ARFI), візуалізацію надзвукового зсуву (SSI) і перехідну еластографію. В останнє десятиліття спостерігається неухильне зростання активності в області еластографії, що демонструє успішне застосування технології в різних областях медичної діагностики та моніторингу лікування.

1.7. Фотоакустична візуалізація

Фотоакустична візуалізація-це нещодавно розроблений гібридний метод біомедичної візуалізації, заснований на фотоакустичному ефекті. Він поєднує в собі переваги оптичного абсорбційного контрасту з ультразвуковим просторовим дозволом для отримання глибоких зображень в (оптичному) дифузному або квазидифузійному режимі. Недавні дослідження показали, що фотоакустична візуалізація може використовуватися *in vivo* для моніторингу ангіогенезу пухлини, картування оксигенації крові, функціональної візуалізації мозку, виявлення меланоми шкіри і т. д.

1.8. Ехокардіографія

Коли для візуалізації серця використовується ультразвук, це називається ехокардіограмою. Ехокардіографія дозволяє побачити детальні структури серця, включаючи розмір камери, функцію серця, клапани серця, а також перикард (мішок навколо серця). Ехокардіографія використовує 2D, 3D і доплерівське зображення для створення зображень серця і візуалізації крові, що протікає через кожен з чотирьох серцевих клапанів. Ехокардіографія широко використовується у багатьох пацієнтів, від тих, хто відчуває такі симптоми, як задишка або біль у грудях, до тих, хто проходить курс лікування раку. Доведено, що трансторакальне ультразвукове дослідження безпечно для пацієнтів будь-якого віку, від немовлят до літніх людей, без ризику шкідливих побічних ефектів або радіації, що відрізняє його від інших методів візуалізації. Ехокардіографія-один з найбільш часто використовуваних методів візуалізації в світі через її портативності і використання в різних додатках. В екстрених випадках ехокардіографія є швидкою, легко доступною і може виконуватися біля ліжка хворого, що робить її методом вибору для багатьох лікарів.

1.9. Висновки до розділу

Медична візуалізація - це метод і процес створення візуальних уявлень внутрішньої частини тіла для клінічного аналізу та медичного втручання, а також візуального представлення функцій деяких органів або тканин (фізіологія). Медична візуалізація спрямована на виявлення внутрішніх структур, прихованих під шкірою і кістками, а також для діагностики та лікування захворювань.

Як дисципліна і в найширшому сенсі вона є частиною біологічної візуалізації і включає радіологію, в якій використовуються технології візуалізації рентгенівської радіографії, магнітно-резонансна томографія, ультразвук, ендоскопія, еластографія, тактильна візуалізація, термографія, медична фотографія, ядерна медицина методи функціональної візуалізації, такі як позитронно-емісійна томографія (ПЕТ) і одноразова фотонно-емісійна комп'ютерна томографія (ОФЕКТ).

РОЗДІЛ 2

СТРУКТУРА ТА МОЖЛИВОСТІ ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ З ГРАФІЧНИМ ПРОЦЕСОРОМ

2.1. Архітектура графічного процесора

Конструктивно графічний процесор являє собою обчислювальний пристрій, що працює окремо від центрального процесора, паралельно з ним. Зазвичай графічні процесори розміщують на окремих друкованих платах з власною системою охолодження, які називають графічними прискорювачами (або відеокартами). Разом з GPU на платі графічного прискорювача розташована відеопам'ять - спеціалізована оперативна пам'ять, в якій зберігаються оброблювані графічним процесором масиви даних.

На рисунку 1.1.1 як приклад показана архітектура графічного процесора G80 - одного з процесорів компанії NVIDIA. Головною (і загальною для всіх графічних процесорів) характеристикою цієї архітектури є те, що GPU являє собою систему з паралельних обчислювальних пристроїв, кожне з яких застосовує задану, єдину для всіх пристроїв, програму (обчислювальне ядро, англ. kernel) до різних елементів вхідних масивів даних, розташованих в загальній пам'яті.



Рис. 2.1.1 Архітектура графічного процесора G80

Паралельна архітектура графічних процесорів орієнтована на виконання алгоритмів, в яких елементи великих вхідних масивів обробляються однаковою чином незалежно або майже незалежно один від одного, тобто використовують розпаралелювання обчислень за даними. Безлічі елементів, що піддаються однотипній незалежній обробці, називають потоками (даних або результатів), тобто графічні процесори здійснюють поточно-паралельну обробку даних.

Концепція програмування, яка полягає в потоковій обробці даних, відома під аббревіатурою SIMD (від англ. Single Instruction - Multiple Data рис. 1.1.2). Процесор, що працює за принципом SIMD, перетворює потік даних в потік результатів, використовуючи програму як функцію перетворення.

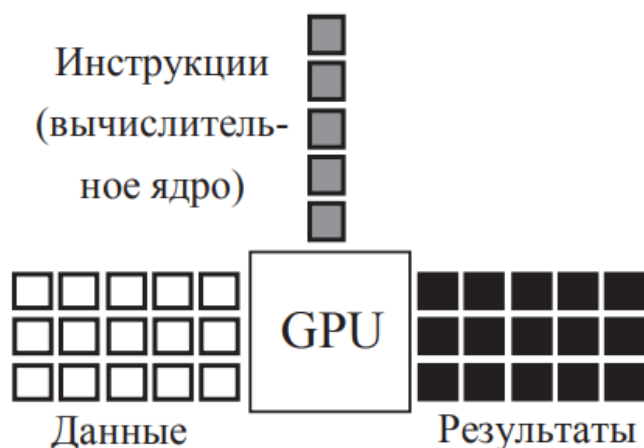


Рис. 2.1.2 Концепція SIMD

Вибір концепції SIMD для графічних процесорів обумовлений тим, що вона забезпечує паралельне використання великої кількості «обчислювачів» без явного управління ними: розподілу завдань, синхронізації обчислень і комунікації між паралельними розрахунками. Розробникам GPU це дозволяє за рахунок спрощення архітектури домагатися більшої продуктивності, а при програмуванні скорочує роботу.

С 2001 по 2006 роки графічні процесори включали в себе «обчислювачі» двох типів: вершинні і піксельні шейдери. Перші були призначені для проектування на площину екрану вершин, які задають поверхні, а другі - для розрахунку кольорів пікселів на екрані. Так випущені в 2006 році процесори ATI Radeon X1900-1950 мали 8 верхових і 48 піксельних конвеєрів. У 2007 році виробники GPU перейшли до уніфікованих потокових процесорів, які замінюють як вершинні, так і піксельні конвеєри. Графічні процесори 2014-2015 років випуску ATI Radeon FijiXT і NVIDIA GM200-400 включають 4096 і 3072 потокових процесора (для 32-бітових обчислень «одинарної» точності) при кілька різної внутрішньої архітектурі. При цьому продуктивність графічних процесорів продовжує швидко збільшуватися.

2.2. Взаємодія графічного та центрального процесорів

Графічний процесор не має засобів прямої взаємодії з пристроями введення-виведення (крім монітора), а також доступу до оперативної пам'яті комп'ютера. Тому управління графічним процесором здійснюється тільки через центральний процесор. Схема взаємодії центрального та графічного процесорів приведена на рис. 1.2.1

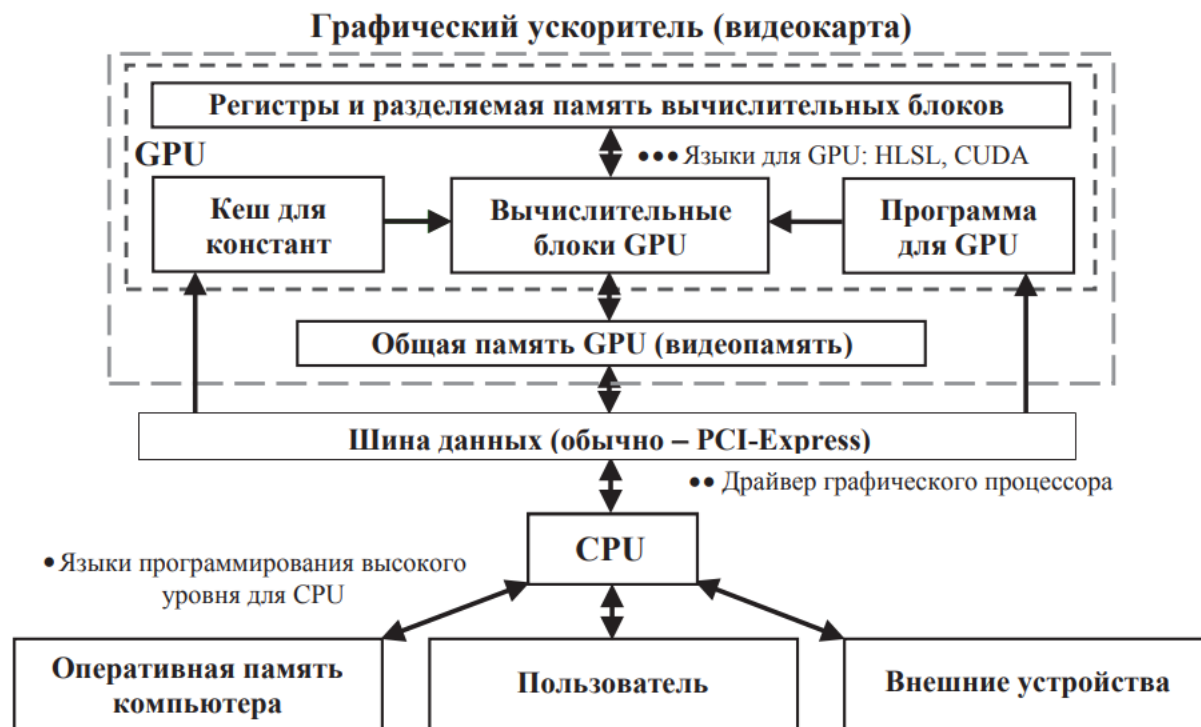


Рис. 2.2.1 Схема взаємодії центрального та графічного процесорів

Графічні прискорювачі підключаються до системної плати персонального комп'ютера через високошвидкісну шину даних (в даний час PCI-Express). За допомогою цієї шини центральний процесор отримує доступ до відеопам'яті, а також до деяких розділів кеш-пам'яті, розташованої на самому графічному процесорі. Через цю ж шину CPU завантажує в графічний процесор програму і запускає її. Перед запуском програми, що виконується на GPU, центральний процесор передає графічному процесору дані двох видів:

- значення констант, використовуваних в програмі;
- один або кілька великих масивів даних для потокової обробки.

До констант необхідний постійний і швидкий доступ, тому вони записуються в кеш-пам'ять (або реєстри), розташовану на кристалі графічного процесора. Масиви даних часто бувають настільки великі, що цілком в кеш-пам'ять не поміщаються. Однак при найпростішій потоковій обробці кожен з елементів масивів даних використовується тільки один раз. Так що для зберігання цих масивів призначена відеопам'ять (загальна пам'ять), що представляє собою окремі мікросхеми на платі графічного прискорювача. Вона працює повільніше кеш-пам'яті і реєстрів, зате має більший обсяг, до декількох гігабайт. Результати своєї роботи графічний процесор може відразу записувати в розділ відеопам'яті, званий буфером кадру, звідки вони передаються на монітор. Але існує також можливість взагалі не відображати розрахунок на екрані, а копіювати результати з відео-пам'яті в оперативну пам'ять комп'ютера, де вони стають доступними для подальшої обробки центральним процесором. На цьому і ґрунтується використання графічних процесорів для обчислень загального призначення, не пов'язаних з обробкою графіки. На схемі (див. рис. 1.2.1) вказані також типи програм, які виконуються центральним і графічним процесорами на різних етапах обробки даних.

2.3. Ієрархія пам'яті, яка доступна центральному та графічному процесорам

Основні види пам'яті обчислювального пристрою, позначені на рис. 2.3.1

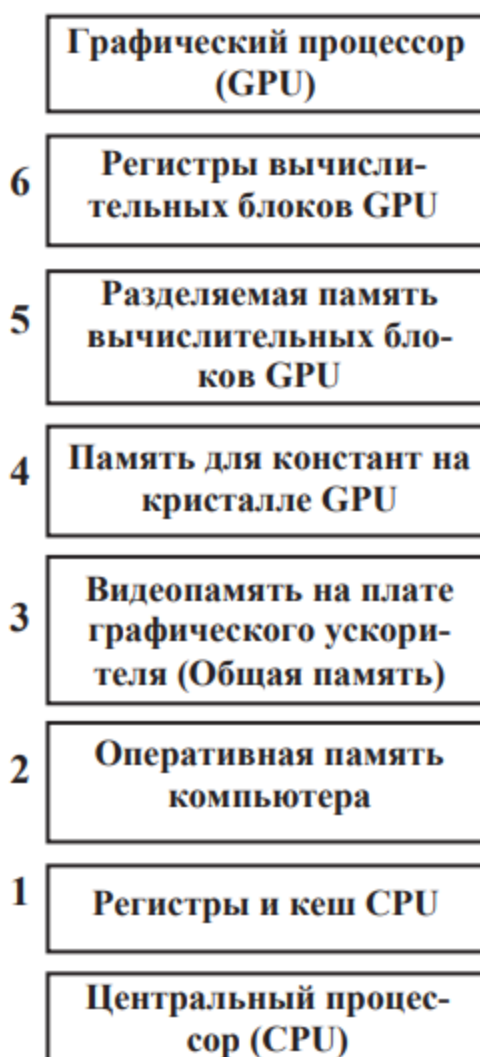


Рис. 2.3.1 Принципы поділу пам'яті

1. Регістри і кеш CPU - пам'ять невеликого (до декількох мегабайтів) обсягу, розташована на кристалі центрального процесора. Доступна тільки для центрального процесора.
2. Оперативна пам'ять комп'ютера - пам'ять великого (до десятків гігабайтів) обсягу в формі мікросхем, розташованих на системній платі. Має середню латентність і пропускну здатність. Доступна тільки центральному процесору. Використовується для зберігання всіх оброблюваних CPU даних, що не поміщаються в кеші.

3. Відеопам'ять - пам'ять, в яку CPU копіює дані, призначені для обробки графічним процесором. Сюди ж GPU записує результати розрахунку. Відеопам'ять має великий обсяг (до декількох гігабайт) і максимально можливу пропускну здатність для потокового доступу до даних. Розташована на платі графічного прискорювача, але не на самому кристалі GPU. Доступна як графічному, так і центральному процесорам для читання і запису.

4. Пам'ять для констант на кристалі GPU. Доступна центрального процесора для запису констант, а графічного процесора - для їх читання.

5-6. Колективна пам'ять і регістри обчислювальних блоків GPU. Розташовані на кристалі графічного процесора і доступні тільки йому. Призначені для довільного доступу до даних з боку паралельних обчислювальних процесів

Використання пам'яті різних типів обумовлено необхідністю балансу між об'ємом пам'яті і швидкістю доступу до даних. Різновиди пам'яті, мають найбільшу ємність, зазвичай характеризуються більшим часом доступу до даних, і навпаки. Швидкодія пам'яті, в свою чергу, визначається двома характеристиками – латентністю і пропускнуою спроможністю.

Латентність - це час доступу до пам'яті, точніше, час очікування процесором даних після запиту. Латентністю визначається продуктивність обчислень при вирішенні завдань, що вимагають частого звернення до різних по розташуванню неврегульованим осередків пам'яті (довільний доступ до пам'яті). Такий обмін з пам'яттю характерний для інтерактивних додатків (програми, що інтенсивно взаємодіють з іншими додатками і з користувачами), а також для додатків, які керують складними процесами. Подібні алгоритми зазвичай виконуються центральним процесором. Як можна менша латентність пам'яті необхідна сучасним центральним процесорам, які працюють на дуже високих частотах, ще й тому, що таким частотам повинна відповідати висока швидкість доступу до даних.

Пропускна здатність пам'яті характеризує обсяг даних, які можуть бути передані до процесора або від процесора за одиницю часу. Висока пропускна здатність виявляється ефективніше низькою латентності в задачах, що дозволяють організувати послідовний доступ до пам'яті - зчитування (або запис) даних з комірок пам'яті, розташованих один за одним, безперервним потоком. При поточно-паралельній обробці даних переважний саме послідовний доступ до пам'яті, тому відеопам'ять, призначена для обміну даними з графічним процесором, повинна володіти максимальною пропускною здатністю навіть на шкоду латентності.

2.4. Висновки до розділу

Графічний процесор являє собою обчислювальний пристрій, що працює окремо від центрального процесора, паралельно з ним. GPU являє собою систему з паралельних обчислювальних пристроїв, кожне з яких застосовує задану, єдину для всіх пристроїв, програму (обчислювальне ядро, англ. kernel) до різних елементів вхідних масивів даних, розташованих в загальній пам'яті. Концепція програмування, яка полягає в потоковій обробці даних, відома під аббревіатурою SIMD. Процесор, що працює за принципом SIMD, перетворює потік даних в потік результатів, використовуючи програму як функцію перетворення.

Основні види пам'яті обчислювального пристрою:

1. Регістри і кеш CPU
2. Оперативна пам'ять комп'ютера
3. Відеопам'ять
4. Пам'ять для констант на кристалі GPU
- 5-6. Колективна пам'ять і регістри обчислювальних блоків GPU

РОЗДІЛ 3

ПОТОЧНО-ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ GPU

3.1. Розпаралелювання розрахунків

Нехай є обчислювальна задача, що передбачає обробку деякого масиву з N однотипних елементів (вихідних даних, об'єктів, що моделюються, випадкових подій), по одному i тому ж алгоритму, що включає m операцій. Тоді блок-схема матиме загальний вигляд, показаний на рис. 3.1.1. З цієї блок-схеми видно, що можливо три наступних підходи до розпаралелювання розрахунків.

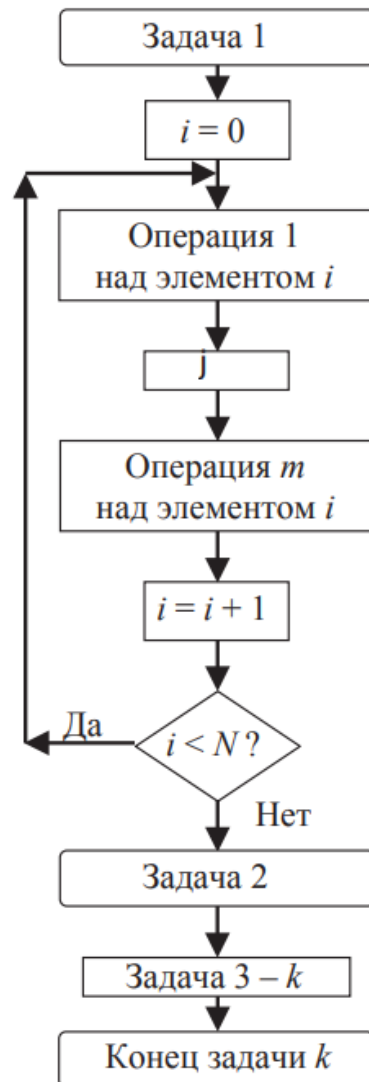


Рис. 3.1.1

3.1.1. Розпаралелювання по задачам

Розпаралелювання за задачам можливо, якщо задачі незалежні одна від одної. Таке розпаралелювання актуально для мережевих серверів і інших обчислювальних систем, що виконують одночасно кілька функцій або обслуговуючих багатьох користувачів. Воно може бути затребуване і в фізичному моделюванні: наприклад, для дослідження однієї і тієї ж системи при різних початкових і зовнішніх умовах. Один графічний процесор, сумісний з CUDA, зможе моделювати одну і ту ж систему одночасно в декількох станах.

3.1.2. Розпаралелювання за інструкціями

Може виявитися, що деякі інструкції з набору операцій незалежні один від одного. При наявності декількох обчислювальних блоків ці інструкції можуть бути виконані паралельно. Таке розпаралелювання апаратно реалізовано в сучасних центральних процесорах загального призначення, оскільки воно ефективно при виконанні програм, що інтенсивно обмінюються різномірною інформацією з іншими програмами і з користувачем комп'ютера.

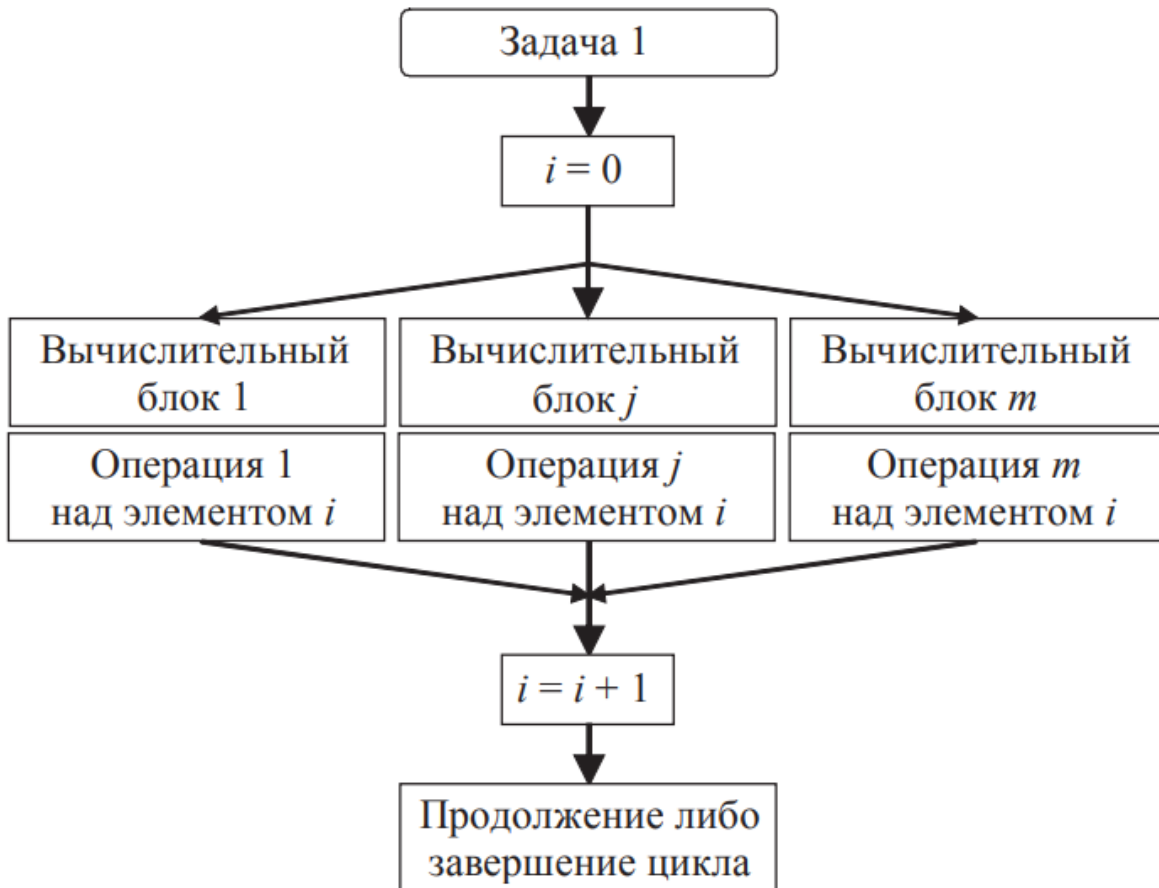


Рис. 3.1.2.1

3.1.3. Розпаралелювання за даними

Якщо операції над кожним i -м елементом взаємозалежні, то ці елементи можна обробляти паралельно. При цьому розрахунки розподіляються між декількома обчислювальними блоками (процесори, конвеєри, машини). Саме можливість розпаралелювання за даними особливо характерна для задач фізико-математичного моделювання.

Процессор 1	:	Операции $1-t$ над элементами $1, M+1, j, N-M+1$
Процессор i	:	Операции $1-t$ над элементами $i, M+i, j, N-M+i$
Процессор M	:	Операции $1-t$ над элементами $M, 2M, j, N$

Рис. 3.1.3.1

При розпаралелювання за даними всі елементи множини нерідко обробляються одним і тим же алгоритмом. В такому випадку обчислення можуть бути організовані за принципом SIMD. Цей принцип, як раз і полягає в незалежній потокової обробки даних по одній програмі, ідеально відповідає архітектурі графічних процесорів, що представляють собою системи з багатьох «обчислювачів», паралельно виконують одне обчислювальне ядро. Передача потокового обробка даних особливо ефективна для алгоритмів, що володіють такими властивостями, характерними для задач фізичного і математичного моделювання:

- Велика щільність обчислень - велике число арифметичних операцій, що припадають на одну операцію введення-виведення (наприклад, звернення до пам'яті). У багатьох сучасних додатках обробки сигналів вона досягає 50: 1, причому зі складністю алгоритмів збільшується;
- Локальність даних по часу - кожен елемент завантажується і обробляється за час, мале по відношенню до загального часу обробки, після чого він більше не потрібен

В результаті в пам'яті потокового процесора для кожного «обчислювача» можна зберігати лише дані, необхідні для обробки одного елемента, на відміну від центральних процесорів з моделлю доволно залежних даних.

3.2. Переваги графічних процесорів при паралельних розрахунках

З часу своєї появи на початку 1980-х років, персональні комп'ютери розвивалися в основному як машини для виконання програм, складних за внутрішньою структурою, що містять велику кількість розгалужень, інтенсивно взаємодіють з користувачем, але рідко пов'язаних з потоковою обробкою великої кількості однотипних даних. Центральні процесори ПК оптимізувалися для вирішення саме таких задач, тому мають наступні характеристики:

- Великою кількістю блоків для управління виконанням програми (кешування даних, проорокування розгалужень і тому подібне) і порівняно малою кількістю блоків для обчислень;
- Архітектурою, оптимальної для програм зі складним потоком управління (обробка різнорідних команд і даних, організація взаємодії програм між собою і з користувачем);
- Пам'яттю з максимальною швидкістю довільного доступу до даних.

Збільшення продуктивності CPU в основному було пов'язано зі збільшенням тактової частоти і розмірів високошвидкісний кеш-пам'ять (пам'ять, розташована прямо на процесорі). Програмування CPU для ресурсномістких наукових обчислень має на увазі ретельне структурування даних і порядку інструкцій для ефективного використання всіх рівнів кеш пам'яті. Ядра сучасних центральних процесорів є супер-скалярними, підтримуючи векторну обробку, Самі ж CPU зазвичай містять кілька ядер. Таким чином, в сукупності центральні процесори можуть реалізовувати десятки паралельних обчислювальних потоків. Однак графічні процесори включають в себе тисячі паралельних «обчислювачів». Крім того, при поточно-паралельних розрахунках графічні процесори мають перевагу завдяки таким особливостям архітектури:

- пам'ять GPU оптимізована на максимальну пропускну здатність (а не на швидкість довільного доступу, як у CPU), що прискорює завантаження потоку даних;

- велика частина транзисторів графічного процесора призначена для обчислень, а не для управління виконанням програми;
- при запитах до пам'яті, за рахунок конвеєрної обробки даних, не відбувається припинення обчислень.

Однак обробка розгалужень (виконання операцій умовного переходу) на GPU менш ефективна, оскільки кожен керівник блок обслуговує не один, а кілька обчислювачів. Таким чином, продуктивність одного GPU при добре розпаралелених обчисленнях аналогічна кластеру з сотень звичайних обчислювальних машин, причому графічні процесори зараз підтримують практично всі операції, які використовуються в алгоритмах загального призначення:

- поширені математичні операції і функції речового аргументу. В рамках SM 4.0 підтримуються цілі числа і логічні операції, а в SM 4.1 і CUDA - також і речові числа подвійний (64-бітної) точності;
- організацію циклів. В SM 3 довжина циклів обмежена 255 ітераціями, в SM 4 довжина циклів не обмежена;
- операції умовного переходу (які виконуються порівняно повільно, оскільки в складі GPU блоків управління менше, ніж обчислювальних блоків).

Графічного процесора доступна відеопам'ять - спеціалізована пам'ять, зазвичай розташована на відеокарті, яка за обсягом близька до оперативної пам'яті персонального комп'ютера. Можливо послідовне і довільне читання даних з відеопам'яті, а також послідовний запис результатів в відеопам'ять. На платформі CUDA можлива і довільний запис в відеопам'ять. Графічний процесор може використовувати регістри - комірки пам'яті, розташовані прямо на процесорі і характеризуються дуже малою латентністю (швидкий доступ до даних):

- є читання з регістрів для констант, які можуть зберігати постійні величини, які не змінюються в ході обробки всіх даних;

- є читання і запис в тимчасові регістри, дані в яких не зберігаються при переході до наступних елементів потоку даних.

Довжина програми для GPU в шейдерній моделі 3.0 обмежена 65 536 інструкціями, чого для більшості обчислювальних алгоритмів достатньо. В SM 4 і на CUDA програми вже можуть мати будь-яку довжину. Обмеженням графічних процесорів в даний час є відсутність динамічного розміщення структур даних (наприклад, зміни розмірів масивів в ході виконання програми). Це обмеження може призводити до неефективного використання пам'яті, проте не перешкоджає виконанню алгоритмів.

3.3. Принцип програмування SIMD на прикладі піксельного шейдера

Оскільки шейдерні моделі програмування SM 3 і SM 4 орієнтовані на обробку графіки, алгоритми для них зручно розглядати в формі завдання тривимірної візуалізації (наприклад, піксельного шейдера). При цьому вихідні дані передаються графічного процесора в формі текстури - двомірного масиву векторів, подібного масиву значень кольору елементів зображення. Таку ж форму мають і масиви результатів розрахунку. Уявити алгоритм загального призначення як піксельний шейдер можна відповідно до таких принципів. Вихідні дані, призначені для потокової обробки, розміщені в масивах з чотиривимірних речових векторів (текстури). При роботі з квітами текселів три компоненти елементів текстури містять інтенсивності червоного, зеленого і синього кольорів, а четверта компонента - ступінь прозорості. Їх значення лежать в діапазоні від 0 до 1, однак для вихідних даних загального призначення потрапляння на відрізок $[0, 1]$ необов'язкове.

Якщо для користувача програмі всі чотири компоненти вхідних векторів не потрібні, то «зайві» компоненти можна залишити нульовими. Масивів описаної структури із вхідними параметрами може бути кілька. До кожного з елементів вхідного масиву при потокової обробці повинен застосовуватися один і той же алгоритм, що не залежить від результатів обробки інших елементів (обчислювальний ядро на рис. 2.1

утворено операціями $(1 - m)$). Цей алгоритм повинен бути реалізований у формі програми - шейдера (на мові HLSL). Масиви вихідних даних (текстури) передаються графічного процесора через відеопам'ять, яка доступна шейдеру. Шейдер потім завантажує і обробляє елементи текстур в поточно-паралельному режимі, як це показано на рисунку.

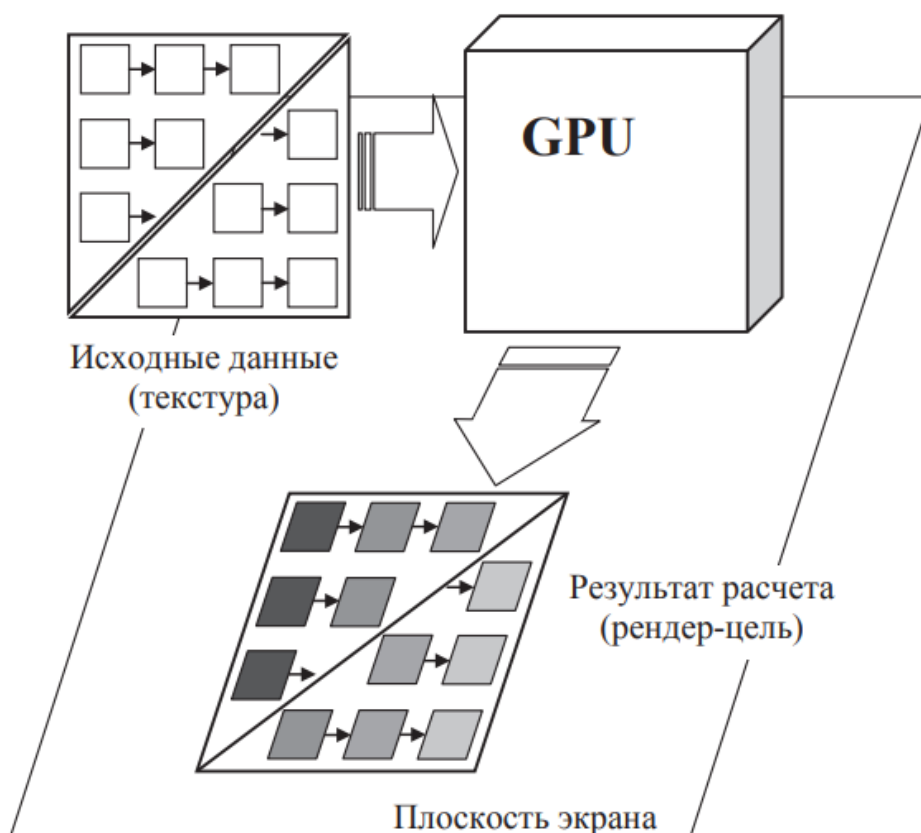


Рис. 3.3.1

Результати розрахунку в формі текстури з чотиривимірних векторів записуються в область відеопам'яті, яку в задачах візуалізації називають рендер-метою. Принцип формування масиву результатів показаний на рис. 2.5.

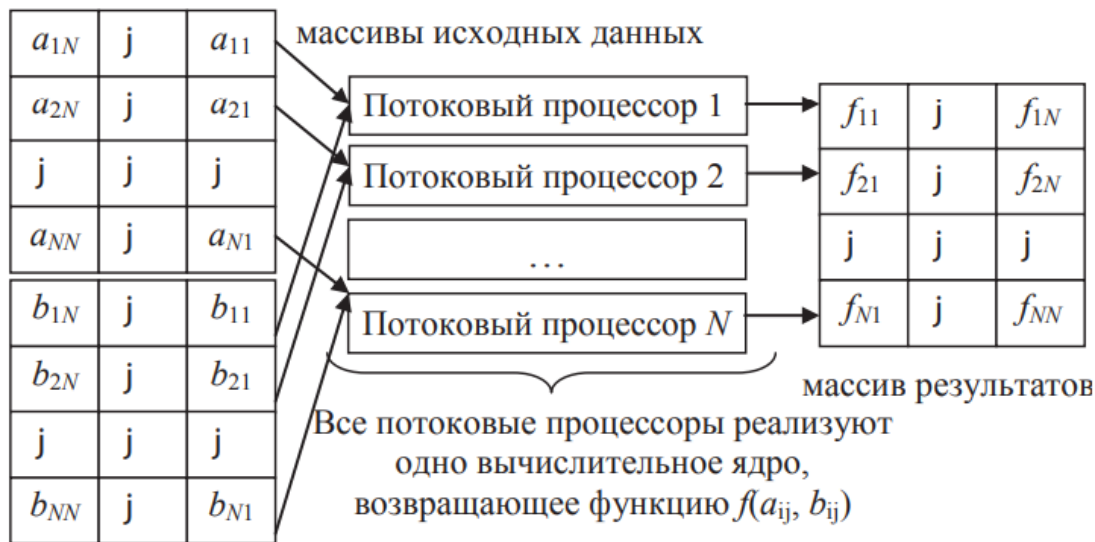


Рис. 3.3.2

Центральний процесор комп'ютера має доступ до відеопам'яті, в тому числі і до рендер-мети. З використанням програми, що виконується на центральному процесорі, рендер мету копіюють в оперативну пам'ять ПК для подальшої обробки.

3.4. Висновки до розділу

Можливо три наступних підходи до розпаралелювання розрахунків:

- 1) Розпаралелювання по задачам
- 2) Розпаралелювання за інструкціями
- 3) Розпаралелювання за даними

Центральні процесори ПК мають наступні характеристики:

- Великою кількістю блоків для управління виконанням програми (кешування даних, пророкування розгалужень і тому подібне) і порівняно малою кількістю блоків для обчислень;
- Архітектурою, оптимальної для програм зі складним потоком управління (обробка різнорідних команд і даних, організація взаємодії програм між собою і з користувачем);
- Пам'яттю з максимальною швидкістю довільного доступу до даних.

Графічні процесори мають перевагу завдяки таким особливостям архітектури:

- пам'ять GPU оптимізована на максимальну пропускну здатність (а не на швидкість довільного доступу, як у CPU), що прискорює завантаження потоку даних;
- велика частина транзисторів графічного процесора призначена для обчислень, а не для управління виконанням програми;
- при запитах до пам'яті, за рахунок конвеєрної обробки даних, не відбувається припинення обчислень.

РОЗДІЛ 4

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПЗ

4.1. Інструменти розробки ПЗ

При створенні програмного забезпечення використовувались наступні інструменти:

4.1.1. Ігровий рушій Unity

Unity – кросплатформене середовище розробки комп'ютерних ігор, розроблене американською компанією Unity Technologies. Unity дозволяє створювати додатки, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-додатки та інші. Випуск Unity відбувся в 2005 році і з того часу йде постійний розвиток. Основними перевагами Unity є наявність візуального середовища розробки, міжплатформенної підтримки та модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатокомпонентними схемами і труднощі при підключенні зовнішніх бібліотек. На Unity написані тисячі ігор, додатків, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів. При цьому Unity використовується як великими розробниками, так і незалежними студіями. Редактор Unity має простий Drag & Drop інтерфейс, а так само установкою плагінів Kali який легко налаштовувати, що складається з різних вікон, завдяки чому можна виробляти налагодження гри прямо в редакторі. Движок використовує для написання скриптів `C#`. Раніше підтримувалися також `Boo` (діалект Python, підтримку прибрали в 5-й версії) і модифікація JavaScript, відома як UnityScript (підтримка припинена у версії 2017.1). Розрахунки фізики виробляє фізичний движок PhysX від NVIDIA. Графічний API-DirectX (на даний момент DX 11, підтримується DX 12). Проект в Unity ділиться на сцени (рівні) — окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, власне, об'єкти (моделі), так і порожні ігрові об'єкти — об'єкти, які не мають моделі

(«пустушки»). Об'єкти, в свою чергу містять Набори компонентів, з якими і взаємодіють скрипти. Також у об'єктів є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь — якого об'єкта на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту і розмірів об'єкта по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель об'єкта видимою. Також Unity підтримує фізику твердих тіл і тканини, а також фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів. При імпорті текстури в Unity можна згенерувати alpha-канал, тір-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна — буде створений матеріал, якому буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання та редагування шейдерів. Редактор Unity має компонент для створення анімації, але також анімацію можна створити попередньо в 3D-редакторі і імпортувати разом з моделлю, а потім розбити на файли.

4.1.2. Мова програмування C#

C# - об'єктно орієнтована мова програмування. Розроблений в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберга і Скотта Вільтаумота як мова розробки додатків для платформи Microsoft.NET Framework. Згодом був стандартизований як ECMA - 334 та ISO/IEC 23270. C# відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (у тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи і методи,

ітератори, анонімні функції з підтримкою замикань, LINQ, винятки, коментарі у форматі XML. Перейнявши багато від своїх попередників-мов C++, Delphi, Модула, Smalltalk і, особливо, Java — C#, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, C# на відміну від C++ не підтримує множинне спадкування КЛАСІВ (тим часом допускається множинна реалізація інтерфейсів).

4.1.3. Мова програмування шейдерів CG

Мова програмування Nvidia Cg (C для графіки) є однією з декількох часто використовуваних мов рендеринга в реальному часі (інші приклади-HLSL Direct3D і GLSL OpenGL). Ці мови використовуються для програмування шейдерів (тобто більш-менш невеликих програм), які виконуються на графічному процесорі (графічному процесорі), тобто процесорі графічної системи комп'ютера, на відміну від центрального процесора (центрального процесора) комп'ютера. Графічні процесори – це масивно паралельні процесори, які надзвичайно потужні. Велика частина сучасної графіки в реальному часі в іграх та інших інтерактивних графічних додатках була б неможлива без графічних процесорів. Однак, щоб повною мірою скористатися продуктивністю графічних процесорів, необхідно запрограмувати їх безпосередньо. Це означає, що повинні бути написані невеликі програми (тобто шейдери), які можуть виконуватися графічними процесорами. Мови програмування для написання цих шейдерів-це мови затінення. Cg-один з них. Фактично, це була одна з перших мов високого рівня для графічних процесорів і була реалізована для декількох API 3D-графіки(інтерфейсів прикладного програмування), найбільш важливих OpenGL і Direct3D. Сьогодні основною причиною його популярності є його схожість з HLSL, який є мовою шейдерів Direct3D від Microsoft.

4.2. Техніка Volume Rendering

Об'ємний рендеринг – техніка, яка використовується для отримання плоского зображення (проекції) тривимірного дискретного набору даних. Під вхідним

набором даних часто мається на увазі безліч плоских зображень шарів, отримане при комп'ютерній томографії або магнітно-резонансної томографії. Зазвичай шари мають рівну товщину (наприклад, фотографується один шар на міліметр) і рівну кількість пікселів на кожен шар. Таким чином, вхідними даними є регулярна сітка вокселів, де кожному вокселу відповідає усереднене значення (температура, щільність матеріалу) в даній точці тривимірного об'єкта. Об'ємна модель може бути отримана або шляхом побудови полігональної сітки на основі вхідних даних, або прямим об'ємним рендерингом. Алгоритм *Marching cubes* є стандартним для перетворення набору вокселів в полігональну модель. Прямий об'ємний рендеринг є складним обчислювальним завданням, яке можна виконати кількома способами.

4.3. Техніка *Raymarching*

Raymarching схожий на традиційне трасування променів (*raytracing*) тим, що промінь в сцену випускається для кожного пікселя. У трасуванні променів у нас є набір рівнянь, що визначають перетин променя і рендерингу об'єктів. Завдяки цьому можна знайти об'єкти, які перетинає промінь (тобто об'єкти, які бачить камера). Також таким чином можна рендерити не полігональні об'єкти, наприклад, сфери, тому що досить тільки знати формулу сфери і променя. Однак *raytracing* дуже витратний, особливо коли в сцені є безліч об'єктів і складне освітлення. Крім того, неможливо виконувати трасування променів через об'ємні матеріали, наприклад, хмари і воду. Тому трасування променів рідко відповідає вимогам додатків реального часу.

Raymarching пропонує інший спосіб вирішення завдання перетину променя і об'єкта. *Raymarching* не намагається безпосередньо обчислити цей перетин аналітично. При ньому ми "крокуємо" точкою уздовж променя, поки не знайдемо, де точка перетинає об'єкт (Рис. 4.3.1).

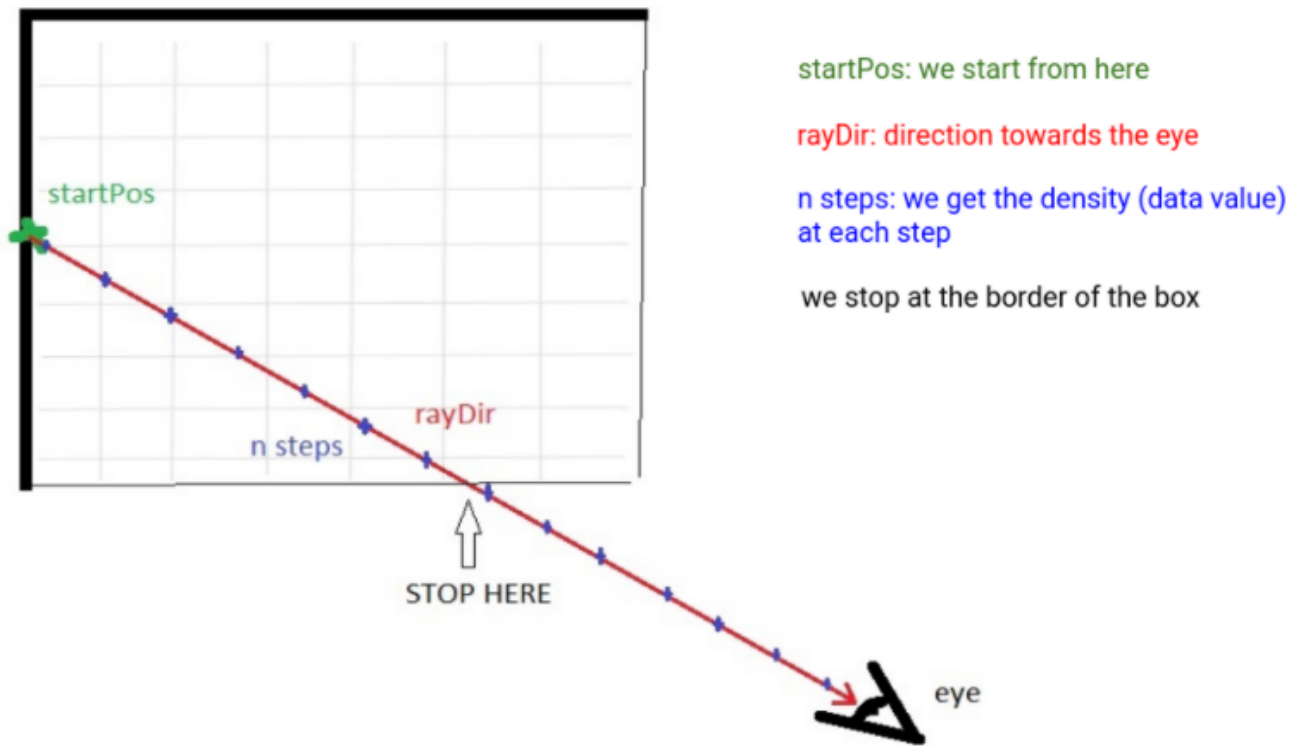


Рис. 4.3.1 Принцип роботи алгоритму Raymarching

4.4. Метод проєкції максимальної інтенсивності

У науковій візуалізації проєкція максимальної інтенсивності (MIP)-це метод для 3D-даних, який проєктує в площині візуалізації вокселі з максимальною інтенсивністю, які падають на шляху паралельних променів, простежених від точки огляду до площини проєкції. Це означає, що два MIP-рендеринга з протилежних точок зору є симетричними зображеннями, якщо вони візуалізуються з використанням ортогональної проєкції.

MIP використовується для виявлення вузликів легенів у програмах скрінінгу раку легенів, в яких використовується комп'ютерна томографія. MIP посилює тривимірну природу цих вузликів, виділяючи їх з легеневих бронхів і судинної мережі. MIP-візуалізація також регулярно використовується лікарями при

інтерпретації досліджень позитронно-емісійної томографії (ПЕТ) або магнітно-резонансної ангіографії.

Цей метод є обчислювально швидким, але результати 2D не дають хорошого уявлення про глибину вихідних даних (Рис. 4.4.1).

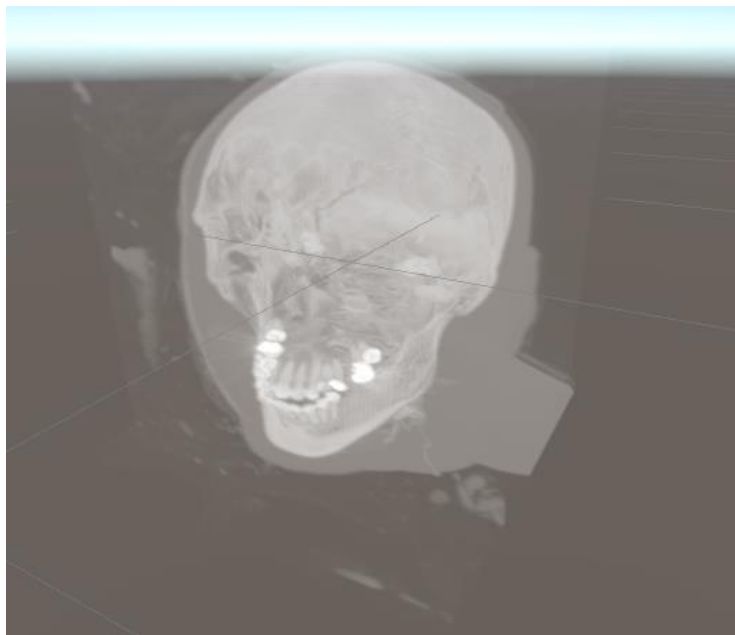


Рис. 4.4.1 Візуалізація через метод MIP

Щоб поліпшити відчуття 3D, анімація зазвичай візуалізується з декількох кадрів MIP, в яких точка зору злегка змінюється від одного до іншого, створюючи ілюзію обертання. Це допомагає сприйняттю глядача знаходити відносні 3D-положення компонентів об'єкта. Однак, оскільки проекція є ортогональною, глядач не може розрізнити, зліва чи справа, спереду або ззаду, і навіть якщо об'єкт обертається за годинниковою стрілкою або проти годинникової стрілки. Використання зважування глибини при виробництві обертових кінокадрів зображень MIP дозволяє уникнути проблеми труднощі розрізнення правого і лівого, а також обертання за годинниковою стрілкою проти годинникової стрілки.

Простим поліпшенням MIP є локальна проекція максимальної інтенсивності, тобто:

$$\boxed{\text{col.a} = \max(\text{src.a})} \quad (4.4.2)$$

У цьому методі ми беремо не глобальне максимальне значення, а перше максимальне значення, яке перевищує певний поріг. Оскільки-в цілому - ми можемо перервати промінь раніше, цей метод швидше, а також дає дещо кращі результати, оскільки він наближається до оклюзії.

4.5. Isosurface rendering

Основна відмінність між рендерингом обсягу iso-поверхні і традиційними методами рендеринга обсягу полягає в тому, що рендеринг обсягу iso-поверхні спрямований на пошук точки перетину променя з ізоперхністю між місцями повторної вибірки на цьому промені замість обчислення непрозорості в місцях повторної вибірки. Непрозорість і градієнт обчислюються в місці перетину.

Найбільш базова реалізація рендеринга isosurface: намалювати перший воксель (значення даних) з щільністю вище деякого порога. Коли ми потрапляємо в воксель з щільністю $>$ поріг, ми зупиняємося і використовуємо цю щільність (Рис 4.5.1).

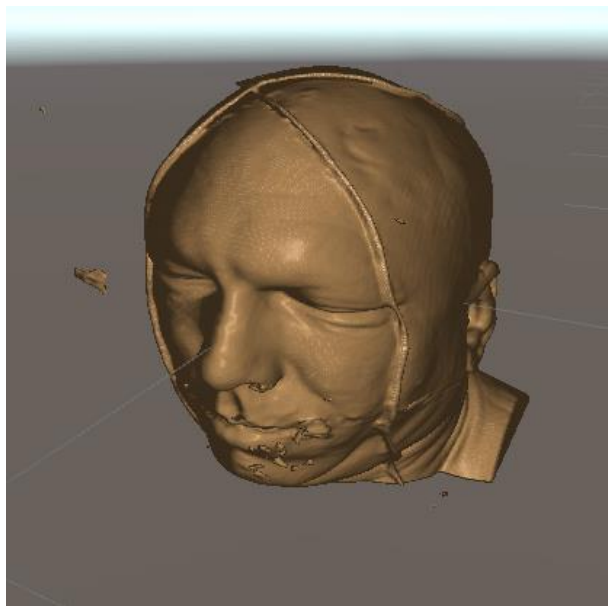


Рис. 4.5.1 Візуалізація через метод Isosurface rendering

4.6. Direct Volume Rendering

Об'ємний рендеринг відрізняється від презентацій томографії з тонкими зрізами, а також зазвичай відрізняється від проекцій 3D-моделей, включаючи проекцію максимальної інтенсивності. Тим не менш, технічно всі візуалізації обсягу стають проекціями при перегляді на 2-мірному дисплеї, що робить відмінність між проекціями і візуалізацією обсягу трохи розпливчастим. Проте, втілення моделей об'ємного рендеринга містять поєднання, наприклад, забарвлення і затінення для створення реалістичних і / або спостережуваних уявлень.

Основна ідея полягає в наступному: на кожному кроці поєднувати поточний колір вокселя з накопиченим значенням кольору з попередніх кроків. Ми лінійно інтерполюємо значення RGB, тобто: $\text{newRGB} = \text{lerp}(\text{oldRGB}, \text{currRGB}, \text{currAlpha})$. А потім додайте нову альфу до старої Альфи, помноженої на (1-Нова альфа):

$$\begin{aligned} \text{col.rgb} &= \text{src.a} * \text{src.rgb} + (1.0f - \text{src.a}) * \text{col.rgb}; \\ \text{col.a} &= \text{src.a} + (1.0f - \text{src.a}) * \text{col.a}; \end{aligned} \tag{4.6.1}$$

Приклад візуалізації черепа за допомогою алгоритма зображень на рисунку 4.6.2

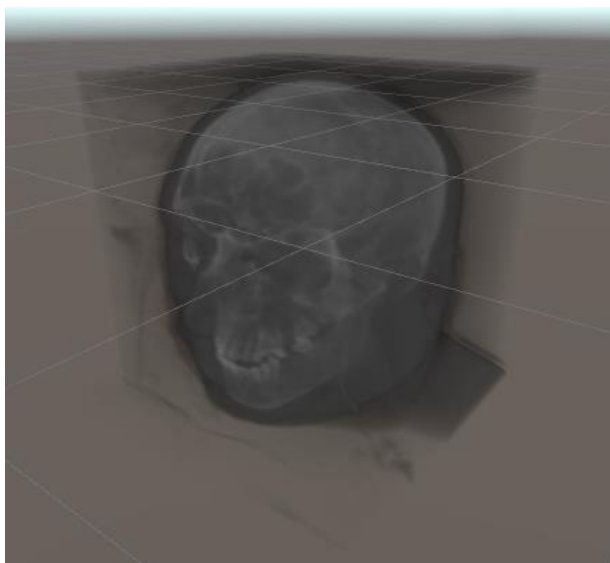


Рис. 4.6.2 Візуалізація через метод *Isosurface rendering*

4.7. Висновки до розділу

При створенні програмного забезпечення використовувались наступні інструменти:

- 1) Unity – кросплатформене середовище розробки комп'ютерних ігор, розроблене американською компанією Unity Technologies.
- 2) C# - об'єктно орієнтована мова програмування
- 3) мова програмування Nvidia Cg

Volume Rendering – набір технік, які використовують для відображення двовимірної проекції тривимірного набору даних. Реалізовано три техніки: Maximum Intensity Projection, Direct volume rendering, Isosurface rendering. В основі кожної з вищесказаних технік використовується алгоритм raymarching.

Maximum Intensity Projection – це метод для візуалізації 3D-даних, який проектує в площині візуалізації вокселі з максимальною інтенсивністю. Вихідний колір буде білим, з Альфа = максимальною щільністю.

Direct Volume Rendering – це метод для візуалізації 3D-даних, основна ідея якого полягає в наступному: на кожному кроці поєднувати поточний колір вокселя з накопиченим значенням кольору з попередніх кроків. Ми лінійно інтерполюємо значення RGB, тобто: $newRGB = lerp(oldRGB, currRGB, currAlpha)$. А потім додайте нову альфу до старої Альфи, помноженої на (1-Нова альфа).

Isosurface Rendering – це метод для візуалізації 3D-даних, найбільш базова реалізація якого полягає в наступному: намалювати перший воксель (значення даних) з щільністю вище деякого порога. Коли ми потрапляємо в воксель з щільністю > поріг, ми зупиняємося і використовуємо цю щільність.

ВИСНОВКИ

Медична візуалізація - це метод і процес створення візуальних уявлень внутрішньої частини тіла для клінічного аналізу та медичного втручання, а також візуального представлення функцій деяких органів або тканин (фізіологія). Медична візуалізація спрямована на виявлення внутрішніх структур, прихованих під шкірою і кістками, а також для діагностики та лікування захворювань.

Графічний процесор являє собою обчислювальний пристрій, що працює окремо від центрального процесора, паралельно з ним. GPU являє собою систему з паралельних обчислювальних пристроїв, кожне з яких застосовує задану, єдину для всіх пристроїв, програму (обчислювальне ядро, англ. kernel) до різних елементів вхідних масивів даних, розташованих в загальній пам'яті. Концепція програмування, яка полягає в потоковій обробці даних, відома під аббревіатурою SIMD. Процесор, що працює за принципом SIMD, перетворює потік даних в потік результатів, використовуючи програму як функцію перетворення.

Можливо три наступних підходи до розпаралелювання розрахунків:

- 4) Розпаралелювання по задачам
- 5) Розпаралелювання за інструкціями
- 6) Розпаралелювання за даними

Графічні процесори мають перевагу завдяки таким особливостям архітектури:

- пам'ять GPU оптимізована на максимальну пропускну здатність (а не на швидкість довільного доступу, як у CPU), що прискорює завантаження потоку даних;
- велика частина транзисторів графічного процесора призначена для обчислень, а не для управління виконанням програми;

при запитах до пам'яті, за рахунок конвеєрної обробки даних, не відбувається припинення обчислень.

При створенні програмного забезпечення використовувались наступні інструменти:

- 4) Unity – кросплатформене середовище розробки комп'ютерних ігор, розроблене американською компанією Unity Technologies.
- 5) C# - об'єктно орієнтована мова програмування
- 6) Мова програмування Nvidia Cg

Volume Rendering – набір технік, які використовують для відображення двовимірної проекції тривимірного набору даних. Реалізовано три техніки: Maximum Intensity Projection, Direct volume rendering, Isosurface rendering. В основі кожної з вищесказаних технік використовується алгоритм raymarching.

Maximum Intensity Projection – це метод для візуалізації 3D-даних, який проектує в площині візуалізації вокселі з максимальною інтенсивністю. Вихідний колір буде білим, з Альфа = максимальною щільністю.

Direct Volume Rendering – це метод для візуалізації 3D-даних, основна ідея якого полягає в наступному: на кожному кроці поєднувати поточний колір вокселя з накопиченим значенням кольору з попередніх кроків. Ми лінійно інтерполюємо значення RGB, тобто: $newRGB = lerp(oldRGB, currRGB, currAlpha)$. А потім додайте нову альфу до старої Альфи, помноженої на (1-Нова альфа).

Isosurface Rendering – це метод для візуалізації 3D-даних, найбільш базова реалізація якого полягає в наступному: намалювати перший воксель (значення даних) з щільністю вище деякого порога. Коли ми потрапляємо в воксель з щільністю > поріг, ми зупиняємося і використовуємо цю щільність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M.J. Bentum (1995) Interactive Visualization of Volume Data. Ph.D. thesis, University of Twente, ISBN 90- 9008788-5.
2. Kaufman, A. (Ed.): Volume Visualization. IEEE Computer Society Press, Los Alamitos, CA, 1991.
3. Gerig, G., Martin, J., Kikinis, R., Kubler, O., Shenton, M., Jolesz, F. A.: Automating segmentation of dual-echo MR head data. In Colchester, A. C. F., Hawkes, D. (Eds.): Information Processing in Medical Imaging, Proc. IPMI '91, Lecture Notes in Computer Science 511, Springer-Verlag, Berlin, 1991, 175-187.
4. Levoy, M.: A hybrid ray tracer for rendering polygon and volume data. IEEE Comput. Graphics Appl. 10, 2 (1990), 33-40
5. Haring, S., Viergever, M. A., Kok, J. N.: A multiscale approach to image segmentation using Kohonen networks. In Barrett, H. H., Gmitro, A. F. (Eds.): Information Processing in Medical Imaging, Proc. IPMI '93, Lecture Notes in Computer Science 687, Springer-Verlag, Berlin, 199 , 212-224.
6. H. Gouraud (1971) Continuous Shading of Curved Surfaces. IEEE Transactions on Computers, 20(6), pp. 623-629.
7. NVIDIA Corp. NVIDIA CUDA Compute Unified Device Architecture. Programming Guide Version 1.0 [Електронний ресурс] — Режим доступу: <http://www.nvidia.com>
8. Reference for HLSL [Електронний ресурс] — Режим доступу: [https://msdn.microsoft.com/ru-ru/library/windows/desktop/bb509638 \(v=vs.85\).aspx](https://msdn.microsoft.com/ru-ru/library/windows/desktop/bb509638 (v=vs.85).aspx)
9. Медицинская визуализация [Електронний ресурс] — Режим доступу: https://star-wiki.ru/wiki/Medical_imaging

ДОДАТКИ

Додаток А

Software Architecture Documentation

VOLUME VISUALIZATION IN MEDICINE

Zhuravlyov Andriy

Version 1.0

30/05/2021

Revision History

Version	Description of Versions / Changes	Responsible Party	Date
1.0	Initial version	Zhuravlyov Andriy	30/05/2021

Approval Block

Version	Comments	Responsible Party	Date

Software Architecture Documentation

Abstract

Humanity receives a huge amount of information thanks to Information Technologies. Therefore, such a huge amount of information requires proper filtering by basic necessities. Through imaging, medical communities are currently recording many breakthroughs in diagnosis and radiation therapy. Computer visualization of information has been recognized as a very useful and accurate tool for translating abstract data into images that can be easily verified and analyzed. DVR is a visualization method that aims to transmit the entire set of 3D data in a 2D image without an intermediate representation. This article reviews a number of previous work on three-dimensional imaging, detailing the challenges, benefits, and limitations of some methods that will help further research in the field of medical volume imaging and possibly scientific imaging in general. Keywords: visualization; direct three-dimensional visualization; medical three-dimensional visualization; radiation therapy; Scientific Visualization

Introduction

The purpose of this document is to provide a detailed architectural design for a new medical imaging project, focusing on four key quality attributes: usability, accessibility, maintainability, and testability. These attributes were selected based on their importance in the development and creation of the application. This document will discuss the prerequisites for this project and architecturally significant functional requirements. Each of the above quality attributes will be described using a complete set of scenarios, followed by an architecture overview including a bird's eye view and a full description of the patterns and tactics that will be used to address the main quality attributes. This will be followed by viewing multiple views in the system. Finally, thank you, links and attachments will complete the document. The purpose of this document is to help the development team determine how the system will be structured at the highest level. It is also designed for project sponsors to sign a high-level structure agreement before the team moves on to detailed design. Finally, the project coach can use this document to confirm that the development team meets the agreed requirements when evaluating the team's efforts.

Volume visualization is a sub-field of scientific visualization that extracts meaningful information from volumetric data using interactive graphics and imaging. It is concerned with volume data representation, modeling, manipulation, and rendering. Volume data are 3D (possibly time-varying) entities that may have information inside them, might not consist of tangible surfaces and edges, or might be too voluminous to be represented geometrically. Hospitals usually rely on radiology departments as their main source of data when it comes to the decision on the treatment of most diseases, hence clinical discipline of radiology is seen to be providing the greater percentage of all diagnoses for patient's treatments. Volumetric data can be obtained using sampling, simulation, or modeling techniques. 2D slice is obtained through different image modalities such as Magnetic Resonance Imaging (MRI), Computed Tomography

(CT), functional MRI (fMRI), or Positron Emission Tomography (PET). In the medical sciences, this is visualized for either diagnostic purpose, planning of treatment or surgery exercises. There are many works already published in medical image visualization, in this paper, we review related works on such visualization techniques and the use of DVR in medical volume visualization. Subsequent sections present basic rendering algorithms, their challenges and proposed approaches to tackle them.

Background

Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the volume visualization project. It presents a number of different architectural views to depict the different aspects of the system.

In order to depict the software as accurately as possible, the structure of this document is based on Philippe Kruchten's "4+1" model view of architecture.

The "4+1" View Model allows various stakeholders to find what they need in the software architecture.

Scope

The purpose of this paper is to explain the architecture of the volume visualization project in medicine.

The project team and developers are working hard to make this project a fully functional and viable technology system for the public market. The vital component was developed some time ago and is in a fairly mature state, but the project as a product is just finishing its incubation state to begin its journey to becoming a startup. As any project turns into a product, it naturally lacks several components necessary for success in a competitive market, such as documentation or project management and scheduling. In addition, only a representative selection of use cases and tests is presented, and flowcharts are not presented due to the lack of preliminary documents, such as the entity model, and insufficient formal and in-depth collection of requirements and discussion of expectations from the system. In addition, the author did not have the opportunity to talk to another developer to discuss general information about the source component of the system and any decisions made. Therefore, at least one future revision of the document is strongly recommended to correct inaccuracies, expand it, and include additional feedback and decisions that have been made in the past. With these considerations in mind, this document was made as accurate as possible and like any artifact created by the developer.

Non-functional requirements

1. High performance: The system must be able to receive a big number of video streams and be able to process the head count and store it on the database each second.
2. User friendly: the final users will be retail companies' managers, and it should not be assumed that are technology experts at all.

3. Security: all the components must be totally secured, in order to prevent leaks and intrusions that could imply physical security issues inside the stores and data manipulation to harm the company.
4. Failure tolerance: the system should be fail proof and be able to recover and keep working in a matter of seconds.
5. Human errors: humans are the #1 source of involuntary (or voluntary) cause of problems in the systems. The system should always check the user input and, in general, any instruction.
6. Maturity: the system has to run tests every time that a change is made, and new tests have to be built for new features. Both unit and inter-module tests should be done.
7. Changeability: everything is very likely to change, so the system must be able to handle any kind of changes in features

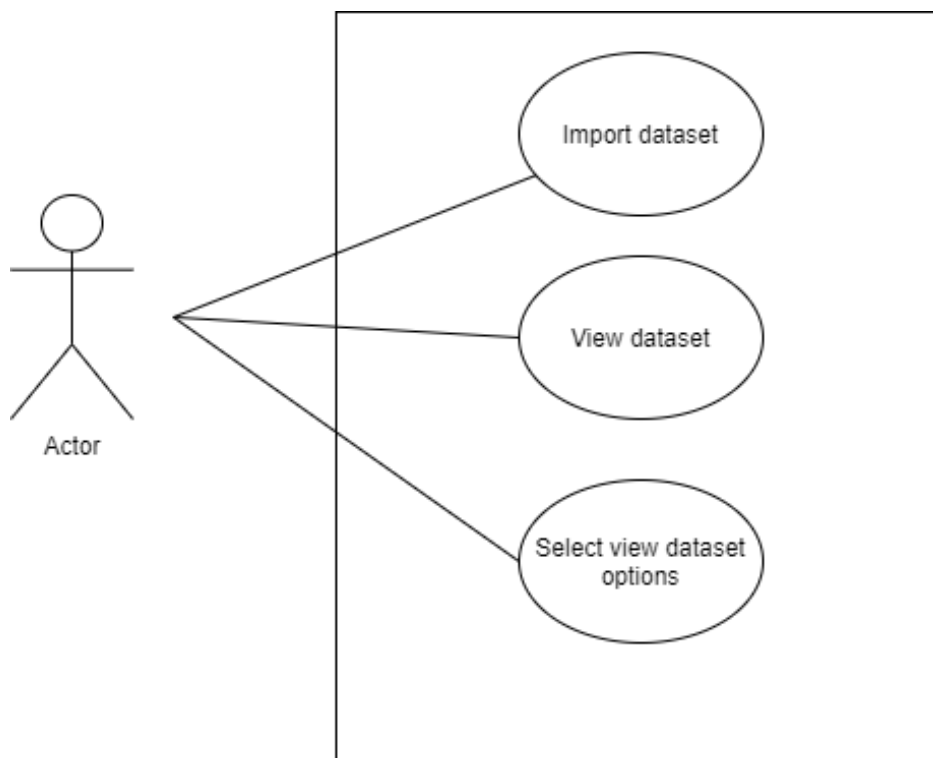
Use-Case View

The purpose of presenting use cases is to provide additional context related to the use of the system and the interaction between its components. For the purposes of this document, each component is treated as a precedent. Lists the current participants and provides a brief description of each of them in the general context of using the system. The most common use cases are described and illustrated using UML use case diagrams and sequence diagrams to clarify the interaction between components.

Actors

User

The user will drive all operation of the software. No distinction is made in regards to type of user. The user interacts with all available interfaces to initiate and monitor all application operations.



Class Diagram View

The purpose of a class diagram is to model a static representation of an application. Class diagrams are the only diagrams that can be directly mapped to object-oriented languages and are therefore widely used in construction.

UML diagrams, such as an action diagram or sequence diagram, can only create a program sequence flow, but the class diagram is slightly different. This is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

