



Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри \_\_\_\_\_

(підпис)

“ 10 ” листопада 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Сизченку Микиті Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Математичні моделі та методи розв'язування обернених  
задач 3D візуалізації за допомогою нейронних мереж

затверджена наказом по університету від 10 листопада 2025 р. № 1028 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 18 грудня 2025 р.

3. Вихідні дані до роботи математична модель нейронної мережи  
та набори даних

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

1. Актуальність теми роботи \_\_\_\_\_

2. Постановка задачі \_\_\_\_\_

3. Аналіз предметної області \_\_\_\_\_

4. Метод чисельного аналізу \_\_\_\_\_

5. Результати обчислювального експерименту \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	10 – 16 листопада 2025 р.	виконано
2	Вибір та обґрунтування методу	17 – 23 листопада 2025 р.	виконано
3	Розробка алгоритму і програми	24 – 30 листопада 2025 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	01 – 07 грудня 2025 р.	виконано
5	Робота над текстом пояснювальної записки	08 – 17 грудня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	18 грудня 2025 р.	виконано

Дата видачі завдання 10 листопада 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Валентин ЄСІЛЕВСЬКИЙ  
(підпис) (посада, Власне ім'я, ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка: 58 с., 13 рис., 1 дод., 31 джерел.

РЕНДЕРИНГ, ГРАДІЄНТНИЙ СПУСК, ОБЕРНЕНА ЗАДАЧА, ФУНКЦІЯ ВТРАТ, 3D-ВІЗУАЛІЗАЦІЯ, КОМП'ЮТЕРНИЙ ЗІР, СИНТЕЗ ЗОБРАЖЕННЯ, ЗБІЖНІСТЬ, КОРЕЛЯЦІЯ.

Об'єкт дослідження – процес розв'язання оберненої задачі локалізації камери в 3D-сцені за допомогою нейронних мереж.

Мета роботи – дослідження математичних моделей та методів розв'язання обернених задач 3D візуалізації, зокрема задачі визначення позиції камери, на основі нейронних представлень сцени.

Методи дослідження – методи інверсії нейронних полів випромінювання в рамках парадигми «аналізу через синтез», засновані на мінімізації фотометричної похибки за допомогою диференційованого волюметричного рендерингу та градієнтної оптимізації.

В даній кваліфікаційній роботі було розглянуто проблему розв'язання обернених задач 3D-візуалізації, зокрема визначення просторового положення (пози) камери. Проведено аналіз існуючих математичних моделей та обґрунтовано вибір методу iNeRF, що базується на інверсії попередньо навченої нейронної мережі (NeRF). Результати обчислювального експерименту продемонстрували високу ефективність обраного підходу: досягнуто значення метрики якості на рівні, що свідчить про точне відновлення пози та візуальну відповідність рендерингу. Отримані результати можуть бути використані в системах робототехніки, автономної навігації та додатках доповненої реальності для точного позиціонування у складних сценах. Значимість роботи полягає у демонстрації переваг неявних нейронних представлень над класичними методами комп'ютерного зору. Доцільним є продовження досліджень у напрямку оптимізації швидкодії методу та його адаптації для роботи з динамічними сценами.

## ABSTRACT

Introductory note: 58 pages, 13 figures, 1 appendix, 31 sources.

RENDERING, GRADIENT DESCENT, INVERSE PROBLEM, LOSS FUNCTION, 3D VISUALIZATION, COMPUTER VISION, IMAGE SYNTHESIS, CONVERGENCE, CORRELATION.

Object of research – the process of solving the inverse problem of camera localization in a 3D scene using neural networks.

Purpose of work – research and software implementation of the method of inverting neural radiance fields (iNeRF) for determining camera pose from a single 2D image.

Methods of research – methods for inversion of neural emission fields within the framework of the "analysis through synthesis" paradigm, based on minimization of photometric error using differentiated volumetric rendering and gradient optimization.

In this qualification work, the problem of solving inverse 3D visualization problems was considered, in particular, the determination of the spatial position (pose) of the camera. An analysis of existing mathematical models was conducted, and the choice of the iNeRF method, which is based on the inversion of a pre-trained neural network (NeRF), was substantiated. The results of the computational experiment demonstrated the high efficiency of the chosen approach: the quality metric value reached a level indicating accurate pose recovery and visual correspondence of the rendering. The obtained results can be used in robotics systems, autonomous navigation, and augmented reality applications for precise positioning in complex scenes. The significance of the work lies in demonstrating the advantages of implicit neural representations over classical computer vision methods. It is advisable to continue research in the direction of optimizing the method's speed and adapting it for work with dynamic scenes.

## ЗМІСТ

	С.
Перелік скорочень, умовних познач, одиниць і термінів .....	8
Вступ .....	9
1 Аналіз предметної області та постановка задач дослідження .....	11
1.1 Огляд математичних моделей обернених задач 3D-візуалізації.....	11
1.2 Аналіз нейронних підходів до розв'язування обернених задач у комп'ютерному зорі .....	13
1.3 Змістовна та формальна постановка задачі .....	15
1.4 Постановка задач дослідження .....	16
2 Вибір та обґрунтування методу розв'язання .....	18
2.1 Теоретичні основи методів вирішення обернених задач 3D-візуалізації	18
2.2 Методи оптимізації в задачах інверсії нейронних моделей .....	20
2.2.1 Стохастичний градієнтний спуск .....	20
2.2.2 Стохастичний градієнтний спуск з моментом .....	22
2.2.3 RMSprop (Root Mean Square Propagation) .....	22
2.2.4 Adam (Adaptive Moment Estimation) .....	23
2.3 Метрики оцінки збіжності алгоритму .....	25
2.4 Метод розв'язання оберненої задачі на основі iNeRF .....	27
Висновки за розділом 2 .....	29
3 Програмна реалізація .....	30
3.1 Фреймворк глибокого навчання PyTorch .....	30
3.2 Алгоритм розв'язання задачі оберненої задачі .....	31
3.3 Опис програми .....	33
Висновки за розділом 3 .....	35
4 Результати обчислювального експерименту та їх аналіз .....	37
4.1 Навчання моделі .....	37
4.2 Оцінка якості моделі .....	38
Висновки за розділом 4 .....	39

	7
Висновки .....	41
Перелік джерел посилання .....	43
Додаток А Лістинг програми .....	46

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ**

iNeRF – Inverting Neural Radiance Fields;

SfM – Structure from Motion;

MVS – Multi-View Stereo;

SGD – Stochastic Gradient Descent;

Adam – Adaptive Moment Estimation;

MSE – Mean Squared Error;

PSNR – Peak Signal-to-Noise Ratio;

6-DOF – Six Degrees of Freedom;

GPU – Graphics Processing Unit;

CUDA – Compute Unified Device Architecture.

## ВСТУП

**Актуальність теми.** У сучасному світі стрімкий розвиток робототехніки, навігаційних систем та технологій доповненої реальності породжує гостру необхідність у точному визначенні просторового положення камери або об'єктів за їх двовимірними зображеннями. Часто класичні методи комп'ютерного зору, що базуються на геометричному зіставленні ключових точок, виявляються неефективними або неможливими для застосування у складних умовах – наприклад, при роботі з прозорими, дзеркальними поверхнями або об'єктами без вираженої текстури. Крім того, традиційні алгоритми нерідко потребують значних обчислювальних ресурсів для побудови щільних 3D-карт і є чутливими до змін освітлення. Зважаючи на це, дослідження новітніх підходів на основі нейронних полів, які дозволяють розв'язувати обернені задачі візуалізації шляхом диференційованої оптимізації (аналіз через синтез), є актуальною темою сьогодення.

**Мета і завдання кваліфікаційної роботи.** Метою кваліфікаційної роботи є дослідження математичних моделей та методів розв'язання обернених задач 3D візуалізації, зокрема задачі визначення позиції камери, на основі нейронних представлень сцени. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану обернених задач 3D візуалізації на основі нейронних представлень сцени;
- дослідити математичні моделі та архітектури диференційованого об'ємного рендерингу;
- розробити нейромережвий підхід для інвертування неявного представлення сцени з метою визначення позиції камери;
- оцінити ефективність та чутливість розробленого методу на основі числових експериментів.

*Об'єктом дослідження є процес розв'язання оберненої задачі локалізації камери в 3D-сцені за допомогою нейронних мереж.*

*Предметом дослідження є математичні методи та нейромережеві підходи до оцінки поз камери в обернених задачах 3D візуалізації.*

**Методи дослідження.** У роботі використовуються методи інверсії нейронних полів випромінювання в рамках парадигми «аналізу через синтез», засновані на мінімізації фотометричної похибки за допомогою диференційованого волюметричного рендерингу та градієнтної оптимізації.

**Публікації.** Результати, отримані у кваліфікаційній роботі, було представлено на XXVIII Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (м. Харків, 16-18 квітня 2024 р.).

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Огляд математичних моделей обернених задач 3D-візуалізації

У галузі комп'ютерного зору та графіки 3D-візуалізація фундаментально поділяється на два типи задач: прямі та обернені. Розуміння прямої задачі є критичним для побудови ефективних методів розв'язання обернених задач, тому розглянемо її першою.

Пряма задача, широко відома як рендеринг, є чітко визначеним детермінованим процесом. Маючи повний опис 3D-сцени, такі як: геометрія, матеріали, освітлення та параметри камери, пряма задача математично симулює фізику світла для генерації 2D-зображення.

Сучасним стандартом математичного моделювання прямої задачі стали нейронні поля випромінювання (NeRF) [1]. Цей метод пропонує представляти сцену не у вигляді дискретних полігональних сіток, а як неперервну 5D-функцію, параметризовану глибокою нейронною мережею. Мережа діє як функція, що запам'ятовує об'ємну щільність та колір у кожній точці 3D-простору (рис. 1.1).

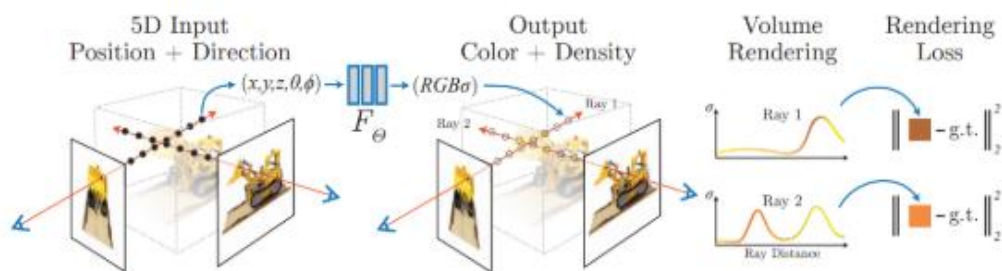


Рисунок 1.1 – Процес синтезу даних за допомогою нейронних полів випромінювання

Для синтезу зображення використовується алгоритм волюметричного рендерингу [2], який пускає промені крізь цю неперервну модель та інтегрує на-

копичену інформацію вздовж них для отримання кольору кожного пікселя (рис. 1.2).

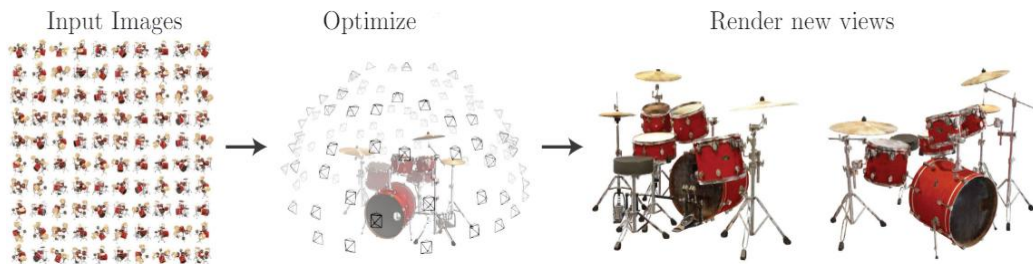


Рисунок 1.2 – Процес реконструкції

Ключовою властивістю цього підходу є повна диференційованість усього процесу рендерингу, що дозволяє використовувати градієнтні методи оптимізації і стало фундаментом для новітніх методів розв'язання обернених задач.

Розглянемо класичні моделі обернених задач. Обернена задача, яка є предметом цього дослідження, полягає у відновленні прихованих параметрів 3D-сцени (геометрії або пози камери) на основі спостережуваних 2D-зображень. Математично ці задачі є некоректними, оскільки характеризуються неоднозначністю результатів та високою чутливістю до зашумлених даних. Історично сформувалися два основних класи математичних моделей для їх розв'язання: геометричні та фотометричні.

Геометричні підходи покладаються на принципи проєктивної геометрії та пошук відповідностей між кадрами. Найбільш фундаментальною моделлю в цій категорії є Structure from Motion [4]. Вона розв'язує задачу одночасного відновлення розрідженої хмари 3D-точок та траєкторії руху камери на основі колекції зображень. Після етапу SfM часто застосовується метод Multi-View Stereo [5], який дозволяє відновити щільну геометрію сцени, базуючись на принципі фотометричної узгодженості пікселів.

Фотометричні моделі аналізують взаємодію світла з поверхнею об'єкта. Класичним прикладом є задача Shape from Shading, що полягає у відновленні 3D-форми з одного зображення шляхом аналізу градієнтів затінення (залеж-

ності інтенсивності пікселя від кута падіння світла). Для подолання неоднозначності цього методу використовується підхід Photometric Stereo, який залучає декілька зображень об'єкта з фіксованої позиції камери, але при різних умовах освітлення.

Незважаючи на математичну обґрунтованість, класичні моделі мають суттєві обмеження: геометричні методи (SfM, MVS) часто зазнають невдач на текстурно-бідних або дзеркальних поверхнях, а фотометричні (SfS, PS) вимагають суворо контрольованих умов освітлення. Ці обмеження стали катализатором для розробки нових підходів на основі нейронних мереж та диференційованого рендерингу.

## 1.2 Аналіз нейронних підходів до розв'язування обернених задач у комп'ютерному зорі

Обмеження класичних математичних моделей, стимулювали пошук нових методів, здатних оперувати зі складними, нелінійними залежностями та невизначеністю, що притаманні оберненим задачам. З розвитком глибокого навчання, нейронні мережі стали потужним інструментом для розв'язання таких задач, оскільки вони здатні вивчати складні апріорні знання безпосередньо з великих обсягів даних.

Ранні нейромережеві підходи до обернених задач 3D-візуалізації часто намагалися розв'язати проблему "в лоб". Наприклад, згорткові нейронні мережі використовувалися для прямої регресії 3D-представлення з одного 2D-зображення. Модель навчалася відображати вхідне зображення у дискретне 3D-представлення, таке як воксельна сітка, тривимірний масив, де кожна комірка позначає наявність матеріалу. Хоча цей підхід продемонстрував життєздатність концепції, він мав суттєві недоліки. Через кубічну складність пам'яті, роздільна здатність воксельних сіток була вкрай обмеженою, що не дозволяло відтворювати дрібні деталі.

Більш досконалі підходи почали використовувати генеративні моделі, такі як генеративно-змагальні мережі або варіаційні автокодувальники. Ці моделі навчаються не просто прямому відображенню, а вивчають сам прихований простір ознак можливих 3D-форм. Навчившись на тисячах прикладів, генеративна модель формує потужне апріорне знання про те, як виглядає правдоподібний 3D-об'єкт. Це дозволяє їй генерувати реалістичні 3D-форми навіть з неповної або зашумленої 2D-інформації.

Ключовий прорив у розв'язанні обернених задач 3D-візуалізації відбувся з популяризацією парадигми "аналізу через синтез", яка поєднала класичну оптимізацію з потужністю нейронних мереж. Ідея полягає в тому, щоб не намагатися безпосередньо регресувати 3D-форму, а натомість побудувати диференційований рендерер. Це математична функція, часто у вигляді нейронної мережі, яка симулює пряму задачу, але зберігає можливість обчислення градієнтів. Маючи такий інструмент, обернена задача розв'язується шляхом оптимізації. Ми вгадуємо невідомі параметри сцени (наприклад, форму чи позу), виконуємо прямий рендеринг (рис. 1.3), обчислюємо різницю між згенерованим та реальним зображенням, і потім, за допомогою градієнтного спуску, коригуємо наші параметри, доки синтезоване зображення не співпаде з цільовим.

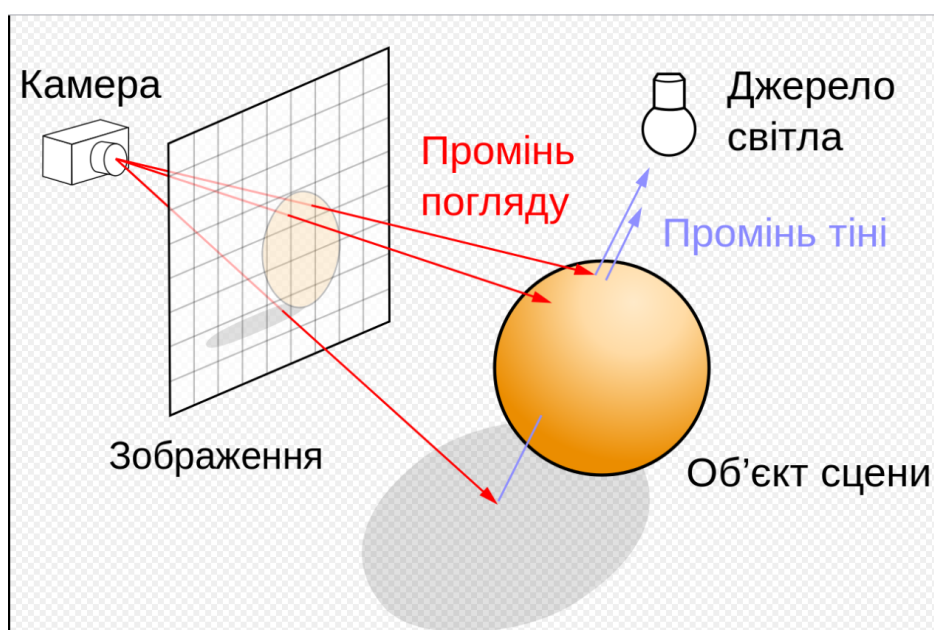


Рисунок 1.3 – Схема роботи рендерингу

Останнім еволюційним кроком у цьому напрямку стали неявні нейронні представлення. Замість явного зберігання 3D-форми у вигляді сітки чи вокселів, ці методи пропонують моделювати сцену як неперервну функцію, параметризовану вагами простої нейронної мережі, зазвичай багат шарового перцептрона. Наприклад, така мережа може приймати 3D-координати  $(x, y, z)$  як вхід і видавати значення щільності або відстань до найближчої поверхні. Таке представлення є неперервним, не залежить від роздільної здатності та є надзвичайно компактним. Саме поєднання неявних нейронних представлень з принципом "аналізу через синтез" за допомогою диференційованого рендерингу призвело до появи найсучасніших методів реконструкції та синтезу зображень, які будуть детально розглянуті в наступних розділах.

### 1.3 Змістовна та формальна постановка задачі

Задача, що розглядається в даній роботі, належить до класу обернених задач 3D-візуалізації, а саме – задачі локалізації камери. Припускається, що ми маємо високоточну, заздалегідь побудовану 3D-модель сцени. Ця модель представлена не у вигляді класичної полігональної сітки, а як навчене нейронне поле випромінювання – неперервна функція, що параметризована вагами нейронної мережі та здатна синтезувати фотореалістичні 2D-зображення з будь-якої точки огляду.

Маючи цю "цифрову копію" сцени, ми отримуємо нове цільове 2D-зображення, зняте реальною камерою. При цьому параметри пози камери – її точне 6-DOF положення в просторі  $(x, y, z)$  та орієнтація – у момент зйомки є невідомими.

Задача полягає у тому, щоб знайти ці невідомі параметри пози, спираючись лише на цільове зображення та навчену нейронну модель сцени. Для її розв'язання використовується підхід "аналізу через синтез".

Формальна постановка задачі може бути представлена наступним чином.

Задана модель сцени у вигляді неперервного відображення

$$F_{\Theta}(x, d) = (c, \sigma), \quad (1.1)$$

де  $x$  – параметри пози камери, що оптимізуються;

$d$  – напрямок огляду;

$c$  – вихідний колір випромінювання в даній точці;

$\sigma$  – вихідна об'ємна щільність;

$\Theta$  – фіксовані ваги нейронної мережі.

Треба знайти оптимальні параметри пози  $P^*$ , які мінімізують суму квадратів фотометричної помилки на наборі променів  $\beta$ :

$$P^* = \arg \min_{P \in SE(3)} \sum_{r \in \beta} \|R(r, P, F_{\theta}) - I_{gt}(r)\|_2^2, \quad (1.2)$$

де  $P \in SE(3)$  – параметри пози камери, що оптимізуються;

$I_{gt}(r)$  – колір цільового 2D-зображення на промені  $r$ ;

$F_{\theta}$  – фіксовані ваги навченої моделі;

$R(r, P, F_{\theta})$  – прогнозований колір променя  $r$ ;

$\beta$  – випадковий набір променів з цільового зображення.

#### 1.4 Постановка задач дослідження

Отже, метою кваліфікаційної роботи є дослідження математичних моделей та методів розв'язання обернених задач 3D візуалізації, зокрема задачі визначення позиції камери, на основі нейронних представлень сцени. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану обернених задач 3D візуалізації на основі нейронних представлень сцени;
- дослідити математичні моделі та архітектури диференційованого об'ємного рендерингу;
- розробити нейромережвий підхід для інвертування неявного представлення сцени з метою визначення позиції камери;
- оцінити ефективність та чутливість розробленого методу на основі числових експериментів.

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

### 2.1 Теоретичні основи методів вирішення обернених задач 3D-візуалізації

Як було встановлено попередньо, обернені задачі 3D-візуалізації є за своєю суттю некоректно поставленими. Для їх розв'язання в рамках обчислювальних методів історично сформувалися два фундаментальні теоретичні підходи: пряме виведення та аналіз через синтез.

Підхід прямого виведення полягає у спробі побудувати складну, високопараметризовану модель (наприклад, глибоку згорткову нейронну мережу), яка навчається безпосередньо відображати 2D-зображення у 3D-представлення. У цьому випадку нейронна мережа  $F$  навчається апроксимувати обернену функцію  $R^{-1}$ , тобто  $3D = F(2D)$ . Модель вивчає кореляції між 2D-ознаками та 3D-структурами на основі величезного навчального набору даних. Хоча цей підхід є дуже швидким на етапі виконання, він має значні теоретичні недоліки. По-перше, він вимагає наявності гігантських, різноманітних та ідеально вирівняних наборів парних даних ("2D-зображення" - "3D-модель"), які складно або неможливо отримати для реальних, складних сцен. По-друге, такі моделі часто погано узагальнюються і зазнають невдачі, коли стикаються з об'єктами або ракурсами, які суттєво відрізняються від навчальних даних.

На противагу цьому, парадигма аналізу через синтез (рис. 2.1) є значно більш гнучкою та потужною, хоча й обчислювально більш витратною на етапі виконання. Вона не намагається вивчити складну обернену функцію. Замість цього, вона використовує знання про пряму задачу (рендеринг), яка є чітко визначеним та відомим математичним процесом. Цей підхід формулює обернену задачу як проблему оптимізації.

Суть методу полягає у тому, щоб ітеративно шукати такі невідомі параметри 3D-сцени, які при проходженні через відому функцію прямого рендерингу генерують 2D-зображення, максимально схоже на цільове зображення.

Математично, це виражається як пошук параметрів, що мінімізують

функцію втрат між синтезованим та цільовим зображеннями:

$$P^* = \arg \min_P \mathcal{L}(R(P), I_{gt}), \quad (2.1)$$

де  $P$  – поточні параметри 3D-сцени, що оптимізуються;

$I_{gt}$  – цільове 2D-зображення;

$\mathcal{L}$  – функція втрат, що вимірює різницю між двома зображеннями;

$R(P)$  – синтезоване 2D-зображення.

Процес оптимізації ітеративно оновлює параметри, щоб мінімізувати цю помилку.

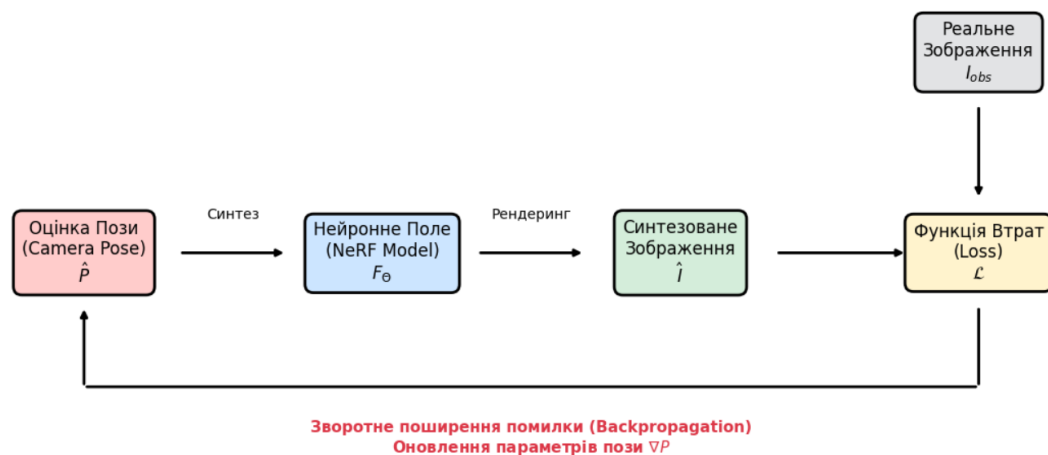


Рисунок 2.1 – Схема методу аналіз через синтез (iNeRF)

Ключовою теоретичною вимогою для реалізації цього підходу є здатність обчислювати градієнт функції втрат відносно параметрів. Цей градієнт вказує, як саме потрібно змінити 3D-параметри, щоб зменшити 2D-помилку на зображенні. Проблема в тому, що градієнт повинен "пройти" крізь процес рендерингу. Традиційні графічні конвеєри містять недиференційовані операції, такі як растеризація, перетворення векторів у пікселі, та  $z$ -буферизація, визначення видимості оклюзії, які унеможливають обчислення градієнта.

Саме тому теоретичною основою сучасних методів вирішення обернених

задач став диференційований рендеринг. Це клас алгоритмів рендерингу, які або точно, або шляхом неперервних апроксимацій дозволяють обчислювати градієнти 2D-зображення відносно вхідних 3D-параметрів. Поява неявних нейронних представлень, зокрема нейронних полів випромінювання (NeRF), забезпечила найбільш елегантне рішення цієї проблеми. NeRF моделює і саму 3D-сцену і процес волюметричного рендерингу, як неперервні, повністю диференційовані функції. Це дозволяє градієнтам з простору 2D-зображення безперешкодно "протікати" аж до 3D-параметрів.

Таким чином, парадигма "аналізу через синтез", реалізована за допомогою диференційованого волюметричного рендерингу нейронних полів, є теоретичною основою, для розв'язання даної оберненої.

## 2.2 Методи оптимізації в задачах інверсії нейронних моделей

### 2.2.1 Стохастичний градієнтний спуск

Як було встановлено у попередньому пункті, теоретична основа нашого методу зводить обернену задачу до проблеми мінімізації функції втрат. Оскільки ми використовуємо диференційовані нейронні представлення, ця мінімізація може бути ефективно розв'язана за допомогою ітеративних методів, що базуються на градієнтах.

Основним методом для оптимізації параметрів у глибокому навчанні є градієнтний спуск. Його ідея полягає в ітеративному коригуванні параметрів, що оптимізуються, у напрямку, протилежному вектору градієнта (рис. 2.2).

У задач інверсії нейронних моделей, обчислення градієнта по всьому зображенню на кожній ітерації є обчислювально надто дорогим. Тому на практиці застосовується стохастичний градієнтний спуск або його варіант з міні-пакетами.

Основне рекурентне співвідношення для оновлення параметрів у методі

SGD має вигляд:

$$P_{t+1} = P_t - \alpha \cdot \nabla_p \mathcal{L}(P_t), \quad (2.2)$$

де  $P_{t+1}$  – оновлені параметри пози на кроці  $t + 1$ ;

$\alpha$  – швидкість навчання, гіперпараметр, що контролює розмір кроку;

$\nabla_p \mathcal{L}(P_t)$  – градієнт функції втрат.

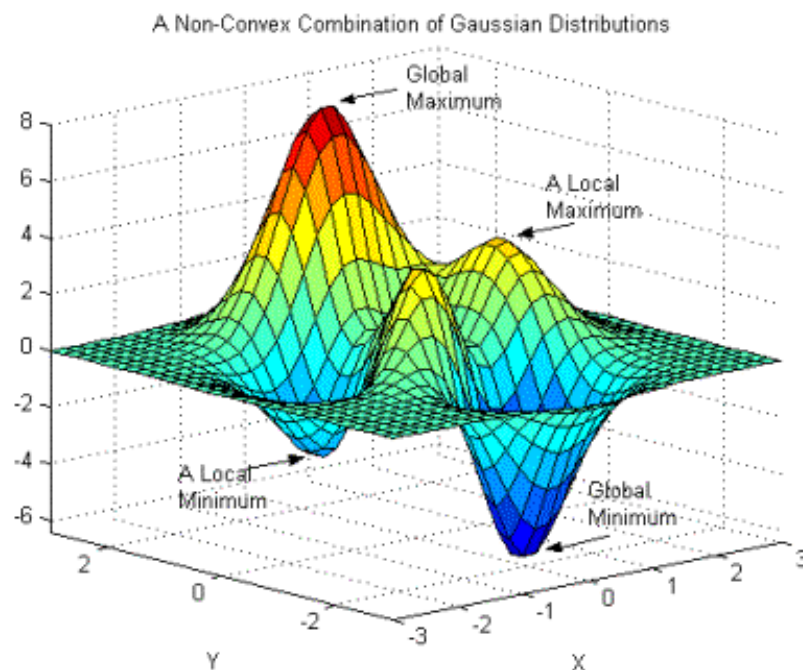


Рисунок 2.2 – Стохастичний градієнтний спуск

Простий стохастичний градієнтний спуск має суттєві недоліки, зокрема складність вибору швидкості навчання та чутливість до складного ландшафту втрат. Для подолання цих проблем були розроблені адаптивні методи оптимізації.

### 2.2.2 Стохастичний градієнтний спуск з моментом

Для подолання нестабільності стохастичного градієнтного спуску був запропонований метод стохастичний градієнтний спуск з моментом. Цей метод вводить поняття "імпульсу" або "моменту", накопичуючи "інерцію" з попередніх кроків. Це дозволяє оптимізатору "прокочуватися" через невеликі локальні мінімуми та плато, де градієнт може бути малим або коливатися, а також прискорює рух вздовж вузьких "ярів" ландшафту втрат. Суть полягає в тому, що вектор оновлення параметрів стає не просто градієнтом, а зваженою сумою поточного градієнта та попереднього вектора оновлення.

$$v_{t+1} = \beta \cdot v_t + (1 - \beta) \cdot \nabla_p \mathcal{L}(P_t), \quad (2.3)$$

$$P_{t+1} = P_t - \alpha \cdot v_{t+1}, \quad (2.4)$$

де  $v_t$  – вектор моменту (ковзне середнє градієнтів) на кроці  $t$ ;

$\beta$  – коефіцієнт збереження моменту.

### 2.2.3 RMSprop (Root Mean Square Propagation)

Метод RMSprop адаптує швидкість навчання для кожного параметра окремо. Він обчислює ковзне середнє квадратів градієнтів, щоб нормалізувати поточний градієнт. У складних нейронних мережах, де різні параметри можуть мати градієнти різної величини та важливості, така єдина швидкість навчання є неефективною. RMSprop адаптує швидкість навчання для кожного параметра окремо, дозволяючи моделі навчатися швидше у потрібних напрямках (рис. 2.3).

$$s_{t+1} = \gamma \cdot s_t + (1 - \gamma) \cdot [\nabla_p \mathcal{L}(P_t)]^2, \quad (2.5)$$

$$P_{t+1} = P_t - \frac{\alpha}{\sqrt{s_{t+1} + e}} \cdot \nabla_p \mathcal{L}(P_t), \quad (2.6)$$

де  $s_t$  – ковзне середнє квадратів градієнтів;

$\gamma$  – коефіцієнт згасання для другого моменту;

$e$  – мала константа для уникнення ділення на нуль.

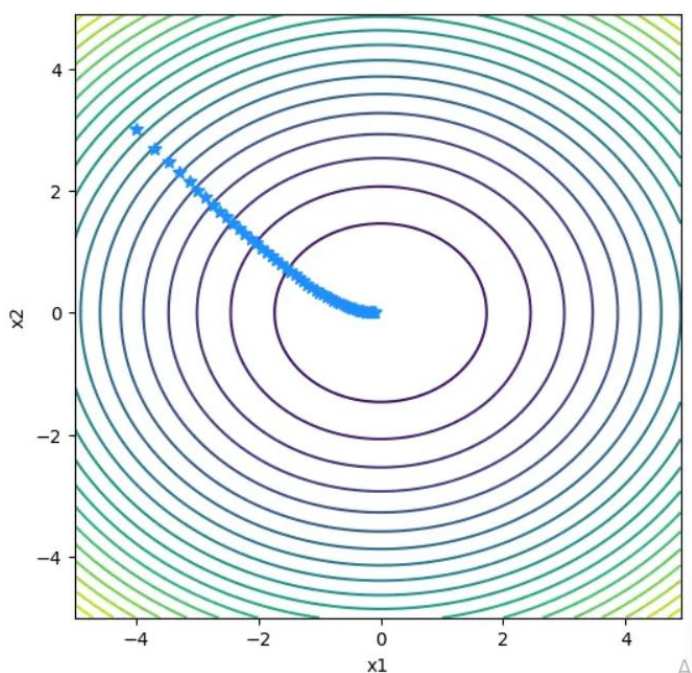


Рисунок 2.3 – Приклад роботи адаптивного методу RMSprop

#### 2.2.4 Adam (Adaptive Moment Estimation)

Найбільш поширений у сучасних дослідженнях, пов'язаних з нейронними полями, метод Adam фактично поєднує обидві ідеї: він використовує і перший момент (моментум, як першому методі), і другий момент (адаптивну швидкість, як у RMSprop).

Спочатку обчислюються ковзні середні для першого та другого моментів:

$$m_{t+1} = \beta_1 \cdot m_t + (1 - \beta_1) \cdot \nabla_P \mathcal{L}(P_t), \quad (2.7)$$

$$v_{t+1} = \beta_2 \cdot v_t + (1 - \beta_2) \cdot [\nabla_P \mathcal{L}(P_t)]^2, \quad (2.8)$$

Потім ці моменти коригуються для усунення зміщення, що виникає на початкових кроках, і використовуються для оновлення параметрів (рис. 2.4):

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}}, \quad \hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^{t+1}}, \quad (2.9)$$

$$P_{t+1} = P_t - \frac{\alpha}{\sqrt{\hat{v}_{t+1} + e}} \cdot \hat{m}_{t+1}, \quad (2.10)$$

де  $m_t$  – ковзне середнє градієнтів;

$\beta_1, \beta_2$  – гіперпараметри, коефіцієнти згасання;

$\hat{m}, \hat{v}$  – кориговані оцінки моментів.

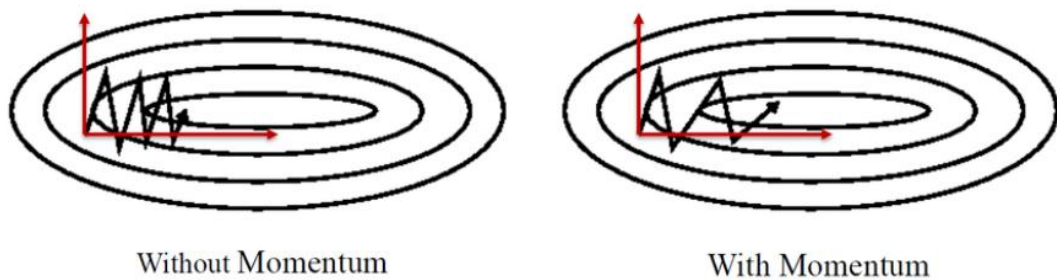


Рисунок 2.4 – Ефективність збіжності алгоритму Adam в умовах поверхні похибки

Завдяки своїй здатності ефективно орієнтуватися у складних ландшафтах втрат, Adam часто демонструє значно швидшу збіжність порівняно з іншими методами і є менш чутливим до початкового вибору гіперпараметрів. Саме тому адаптивні оптимізатори, і зокрема Adam, є де-факто стандартом для розв'я-

зання оптимізаційних задач інверсії, подібних до тієї, що поставлена в нашій роботі.

### 2.3 Метрики оцінки збіжності алгоритму

Для кількісної оцінки якості розв'язання оберненої задачі та моніторингу процесу оптимізації пози камери критично важливим є вибір відповідних метрик. У даному дослідженні використовуються дві основні метрики: середньоквадратична похибка (MSE) та пікове співвідношення сигналу до шуму (PSNR).

Середньоквадратична похибка (MSE) є ключовою функцією втрат, мінімізація якої лежить в основі методу «аналізу через синтез». Вона визначає усереднену квадратичну різницю між значеннями кольорів пікселів синтезованого зображення та відповідними пікселями реального цільового зображення. Математично для набору з  $N$  променів (пікселів) у пакеті  $\mathcal{B}$  вона розраховується як:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{r \in \mathcal{B}} \|C_{render}(r) - C_{gt}(r)\|_2^2, \quad (2.11)$$

де  $C_{render}(r)$  – колір пікселя, отриманий в результаті волюметричного рендерингу (прогнозоване значення);

$C_{gt}(r)$  – колір пікселя на цільовому зображенні (істинне значення).

Саме градієнт цієї функції  $\nabla \mathcal{L}_{MSE}$  використовується оптимізатором для оновлення параметрів пози камери на кожній ітерації (рис. 2.5).

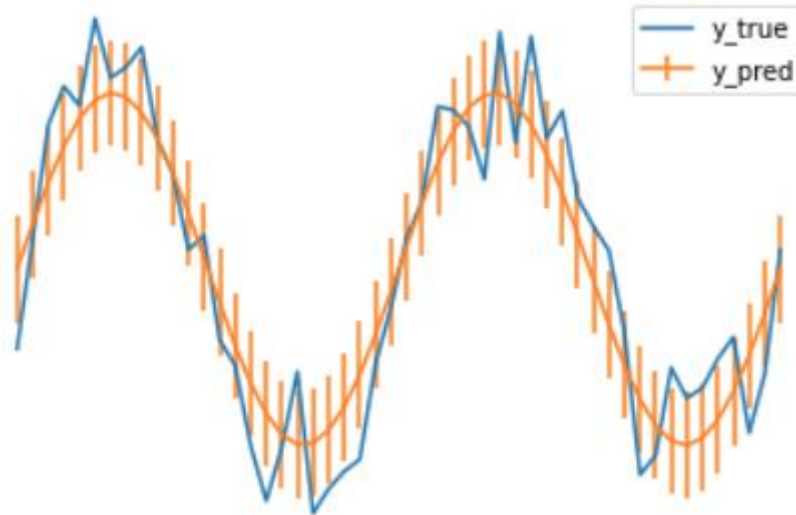


Рисунок 2.5 – Графік зміни функції втрат (MSE) під час навчання

Пікове співвідношення сигналу до шуму (PSNR) є метрикою, що широко застосовується в задачах обробки зображень та комп'ютерного зору для оцінки якості реконструкції. Вона вимірюється в логарифмічній шкалі децибелах (dB) і дозволяє оцінити, наскільки синтезоване зображення близьке до еталонного, ігноруючи незначні шумові відхилення. PSNR обчислюється на основі MSE:

$$PSNR = -10 \cdot \log_{10}(MSE), \quad (2.12)$$

У цій формулі передбачається, що значення інтенсивності пікселів нормалізовані в діапазоні  $[0,1]$ . На відміну від MSE, де менше значення означає кращий результат, для PSNR вищі значення свідчать про вищу якість збігу зображень. Метрика PSNR використовується не лише як показник якості, але і як критерій зупинки алгоритму, тобто оптимізація припиняється, коли досягається певний пороговий рівень PSNR, що свідчить про те, що знайдена поза забезпечує достатньо точний візуальний збіг з оригіналом.

## 2.4 Метод розв'язання оберненої задачі на основі iNeRF

Загальний алгоритм, описаний у попередньому підрозділі, є основою для багатьох методів аналізу через синтез. У даній роботі, для розв'язання поставленої задачі локалізації камери, в якості конкретного методу був обраний iNeRF (Inverting Neural Radiance Fields). Цей метод безпосередньо реалізує описаний алгоритм, використовуючи як модель сцени  $F_{\ominus}$  та рендерер  $R$  архітектуру нейронних полів випромінювання (NeRF).

У методі iNeRF, модель сцени – це багатошаровий перцептрон, навчений представляти 3D-сцену як неперервну 5D-функцію. Ваги цієї мережі є заздалегідь навченими на наборі зображень сцени з відомих поз і є фіксованими під час розв'язання оберненої задачі. Ця функція приймає 5D-координати (3D-позицію  $x$  та 2D-напрямок огляду  $d$ ) і видає об'ємну щільність  $\sigma$  та колір  $c$  у цій точці:

$$F_{\ominus} : (x, d) \rightarrow (c, \sigma). \quad (2.11)$$

Диференційований рендерер  $R$  у методі iNeRF є процедурою волюметричного рендерингу, запозиченою з оригінального NeRF. Ця процедура є повністю диференційованою і симулює, як промінь камери  $r(t) = o + td$ , що виходить з центру камери  $o$  у напрямку  $d$  накопичує колір, проходячи крізь сцену. Очікуваний колір  $C(r)$  променя  $r$ , що проходить від ближньої межі  $t_n$  до дальньої  $t_f$ , обчислюється за інтегралом:

$$C(r) = \int_{t_n}^{t_f} T(t) \cdot \sigma(x(t)) \cdot c(x(t), d) dt, \quad (2.12)$$

де  $T(t)$  – пропускна здатність вздовж променя від  $t_n$  до  $t$ ;

$\sigma(x(t))$  – об'ємна щільність у точці  $x(t)$  на промені, отримана з  $F_{\ominus}$ ;

$c(x(t), d)$  – колір у точці  $x(t)$ , отриманий з  $F_{\ominus}$ .

Пропускна здатність  $T(t)$  сама визначається як ймовірність того, що промінь пройде від  $t_n$  до  $t$ , не зіткнувшись з частинками:

$$T(t) = \exp -\int_{t_n}^t \sigma(x(s)) ds , \quad (2.13)$$

Для практичної реалізації алгоритму, ці неперервні інтеграли апроксимуються дискретними сумами. Промінь розбивається на  $M$  відрізків. Колір пікселя  $\hat{C}(r)$  обчислюється як:

$$\hat{C}(r) = \sum_{i=1}^M T_i \cdot \alpha_i \cdot c_i , \quad (2.14)$$

де  $c_i$  – колір, отримані з  $F_{\ominus}$  для  $i$ -го відрізка;

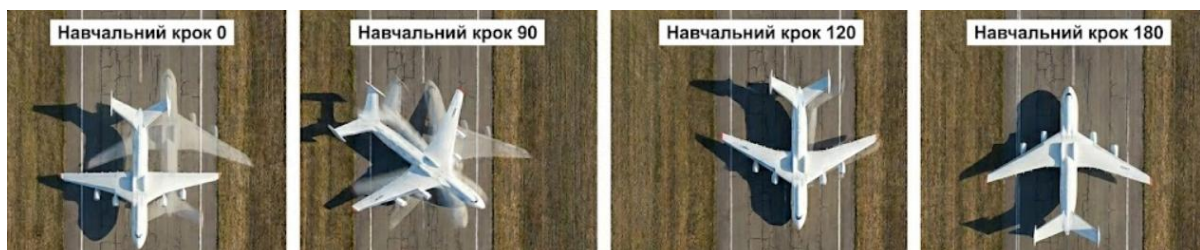
$\sigma_i$  – щільність, отримані з  $F_{\ominus}$  для  $i$ -го відрізка;

$\alpha_i = (1 - \exp(-\sigma_i \cdot \delta_i))$  – альфа-значення (непрозорість)  $i$ -го відрізка;

$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \cdot \delta_j\right)$  – дискретна пропускна здатність до  $i$ -го відрізка;

$\delta_i$  – довжина  $i$ -го відрізка.

Процес ітеративної оптимізації, який реалізує iNeRF, візуалізовано на рисунку 2.6.



Рисунку 2.6 – Візуалізація процесу інверсії NeRF (iNeRF)

Ключова ідея методу iNeRF полягає в тому, що вся ця процедура, яка використовує модель  $F_{\Theta}$ , є повністю диференційованою не лише відносно ваг  $\Theta$ , але й відносно вхідних параметрів променя  $o$  та  $d$ . Оскільки  $o$  та  $d$  безпосередньо визначаються позою камери, ми можемо обчислити градієнт фотометричної помилки  $\mathcal{L}$  відносно  $P$ .

Використовуючи метод iNeRF в якості конкретизації загального алгоритму та адаптивний оптимізатор (Adam), ми отримуємо повний, готовий до реалізації метод для розв'язання поставленої задачі локалізації камери.

## Висновки за розділом 2

В даному розділі було виконано:

- огляд теоретичних основ та методів вирішення обернених задач 3D-візуалізації, включаючи підхід «аналізу через синтез» та градієнтні методи оптимізації (SGD, Adam тощо);
- дослідження обраного підходу до вирішення задачі, а саме методу iNeRF, що базується на інверсії нейронних полів випромінювання за допомогою диференційованого волюметричного рендерингу.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Фреймворк глибокого навчання PyTorch

PyTorch – це передовий фреймворк для глибокого навчання з відкритим вихідним кодом, який на сьогодні став де-факто стандартом у науковій спільноті, що займається комп'ютерним зором та нейронним рендерингом. Ця бібліотека поєднує в собі гнучкість динамічних обчислювальних графів із високою продуктивністю, що робить її ідеальним інструментом для реалізації складних, нестандартних архітектур, таких як Neural Radiance Fields (NeRF).

Однією з ключових особливостей PyTorch, яка визначила його вибір для даної роботи, є парадигма "Define-by-Run". На відміну від статичних графів, де структура обчислень фіксується заздалегідь, PyTorch будує граф динамічно під час виконання коду. Це критично важливо для реалізації ітеративних алгоритмів оптимізації, подібних до iNeRF, де логіка виконання може змінюватися в залежності від умов. Такий підхід не лише забезпечує необхідну гнучкість, але й значно спрощує процес налагодження та експериментування з новими ідеями.

Фундаментальним компонентом фреймворку є модуль автоматичного диференціювання Autograd, який дозволяє автоматично обчислювати градієнти для довільних скалярних функцій втрат. У контексті обернених задач 3D-візуалізації ця можливість набуває особливого значення, оскільки дозволяє знаходити похідні не лише відносно ваг нейронної мережі, але й відносно вхідних параметрів, таких як поза камери. Завдяки цьому стає можливим реалізувати підхід "аналізу через синтез", пропускаючи градієнти через весь процес волюметричного рендерингу для точного налаштування просторового положення.

Крім того, PyTorch надає потужні засоби для роботи з тензорами та підтримку апаратного прискорення на графічних процесорах (GPU) через CUDA. Оскільки волюметричний рендеринг вимагає виконання мільйонів операцій семплювання та інтегрування для кожного зображення, можливість ефек-

тивного розпаралелювання обчислень є необхідною умовою для досягнення прийнятної швидкості роботи. Широка підтримка спільноти та наявність еталонних реалізацій компонентів NeRF роблять PyTorch оптимальним середовищем для розробки системи інверсії нейронних полів.

### 3.2 Алгоритм розв'язання оберненої задачі

Алгоритм є ітеративним оптимізаційним процесом. Його фундаментальна мета – здійснити пошук у 6-вимірному просторі можливих поз камери для знаходження такого набору параметрів, який мінімізує функцію фотометричних втрат. Ця функція кількісно оцінює розбіжність між зображенням, синтезованим з поточної пози, та еталонним цільовим зображенням.

Для коректної роботи алгоритму необхідні такі вхідні дані.

Нейронна модель сцени. Повністю навчена нейронна мережа, ваги якої є фіксованими. Вона виступає в ролі точної, неперервної та диференційованої "карти" 3D-сцени.

Цільове 2D-зображення. Еталонне зображення, для якого необхідно знайти невідому позу камери.

Початкова оцінка пози. Початкове 6-DOF значення, з якого починається оптимізаційний пошук. Вибір може бути важливим, оскільки складний ландшафт втрат може містити локальні мінімуми; занадто далека початкова точка може призвести до збіжності до некоректного розв'язку.

Гіперпараметри оптимізації.

$N$  – загальна кількість ітерацій оптимізації (бюджет часу на пошук).

$B$  – розмір пакету променів (кількість пікселів, що обираються на кожній ітерації), що є компромісом між точністю градієнта та швидкістю однієї ітерації;

$\alpha$  – швидкість навчання для оптимізатора.

Сам алгоритм розв'язання оберненої задачі складається з наступних кроків.

Крок 1. Ініціалізація. Встановити поточну позу камери  $P_t \leftarrow P_0$ . Ініціалізувати обраний адаптивний оптимізатор для оновлення параметрів. Оптимізатор буде зберігати свій внутрішній стан для  $P_t$ .

Крок 2. Запуск циклу оптимізації. Виконувати наступні кроки в циклі для  $t$  від 0 до  $N - 1$ .

Крок 2.1. Вибірка даних. Замість обчислення втрат по всьому зображенню, що є обчислювально не вигідно, застосовується стохастичний підхід. З цільового зображення  $I_{gt}$  формується випадковий набір з  $B$  променів (координат пікселів).

Крок 2.2. Синтез. На цьому етапі виконується пряма задача. Використовуючи поточну оцінку пози камери  $P_t$  та фіксовану модель сцени  $F_{\Theta}$ , виконується процедура диференційованого рендерингу для кожного променя  $r \in \mathcal{B}$ . Це дає нам пакет синтезованих кольорів пікселів  $I_{render}(r)$ . Цей крок відповідає "синтезу" у парадигмі аналізу через синтез.

Крок 2.3. Обчислення втрат. На цьому етапі виконується аналіз. Розраховується скалярне значення функції втрат  $\mathcal{L}$ , яке кількісно оцінює, наскільки сильно синтезовані кольори відрізняються від відповідних еталонних кольорів з цільового зображення.

Крок 2.4. Обчислення градієнта. Це ключовий крок, який став можливим завдяки диференційованій природі  $R$  та  $F_{\Theta}$ . Методом зворотного поширення помилки обчислюється вектор градієнта функції втрат відносно параметрів, що оптимізуються, тобто пози  $P_t : \nabla_P \mathcal{L}(P_t)$ . Цей 6-вимірний вектор вказує, як саме потрібно змінити положення та орієнтацію камери, щоб найсильніше зменшити поточну фотометричну помилку.

Крок 2.5. Оновлення параметрів. Отриманий градієнт  $\nabla_P \mathcal{L}(P_t)$  подається до оптимізатора (Adam). Оптимізатор, враховуючи свій внутрішній стан (накопичені моменти) та гіперпараметр швидкості навчання  $\alpha$ , обчислює крок оновлення та застосовує його до поточної пози, отримуючи нову, покращену оцінку  $P_{t+1}$ .

Крок 3. Завершення. Після  $N$  ітерацій, оптимізаційний цикл примусово завершується. Кінцева поза  $P_N$  приймається як остаточний розв'язок задачі  $P^*$ , найкраща оцінка пози камери, яку вдалося знайти.

Таким чином, представлений алгоритм є потужним інструментом для розв'язання оберненої задачі локалізації, оскільки він дозволяє знаходити розв'язок для складної 6-вимірної нелінійної системи, покладаючись лише на фотометричну помилку. Однією з головних його переваг є гнучкість, оскільки він не вимагає явного зіставлення ключових точок і може працювати з будь-якою диференційованою моделлю сцени  $F_{\Theta}$ .

Проте, цей метод має і суттєві обмеження. Як і всі оптимізаційні підходи на основі градієнтів, він чутливий до локальних мінімумів, якщо початкова оцінка пози  $P_0$  є надто далекою від істинного розв'язку, алгоритм може збігтися до некоректного. Крім того, ефективність сильно залежить від вибору гіперпараметрів оптимізатора. Занадто великий крок може призвести до нестабільності, а занадто малий – до повільної збіжності. Також метод є обчислювально витратним, оскільки кожна ітерація вимагає повного прямого та зворотного проходу через глибоку нейронну мережу.

### 3.3 Опис програми

Програмна реалізація методу iNeRF виконана мовою програмування Python у середовищі інтерактивних блокнотів Jupyter Notebook, що дозволяє поєднувати код, візуалізації та описовий текст у єдиному документі. Основним інструментом для побудови та виконання обчислювальних графів обрано бібліотеку PyTorch, яка забезпечує автоматичне диференціювання та підтримку обчислень на графічному процесорі (GPU), а для роботи з масивами даних та зображеннями використовуються бібліотеки NumPy, Pillow та Matplotlib. Робота програми розпочинається з імпорту необхідних модулів та конфігурації середо-

вища, де автоматично визначається наявність CUDA-сумісного прискорювача для забезпечення необхідної швидкодії волуметричного рендерингу.

Далі відбувається підготовка даних, яка включає завантаження вхідного та цільового зображень, їх нормалізацію та перетворення у тензори. Ключовим етапом є ініціалізація нейронної моделі сцени, в якості якої використовується архітектура PixelNeRF із попередньо натренованими вагами. На відміну від класичного навчання, ваги нейронної мережі у цій програмі залишаються фіксованими, оскільки задача полягає не у зміні параметрів сцени, а у пошуку положення камери. Для цього створюється окремий тензор, що представляє матрицю пози, якому встановлюється атрибут, що дозволяє обчислювати градієнти, а початкове значення задається зі зміщенням відносно істинного для симуляції реальних умов пошуку.

```
class ConvEncoder(nn.Module):
    def __init__(self, dim_in=3, norm_layer=util.get_norm_layer("group")):
        super().__init__()
        self.dim_in = dim_in
        # Початкові параметри шарів
        first_layer_chnls = 64
        mid_layer_chnls = 128
        last_layer_chnls = 128
        n_down_layers = 3
        # Вхідний згортковий блок
        self.conv_in = nn.Sequential( nn.Conv2d(dim_in, first_layer_chnls, kernel_size=7, stride=2, bias=False),
                                     norm_layer(first_layer_chnls),
                                     nn.LeakyReLU() )
        # Створення шарів downsampling (зменшення розмірності)
        chnls = first_layer_chnls
        for i in range(0, n_down_layers):
            conv = nn.Sequential(
                nn.Conv2d(chnls, 2 * chnls, kernel_size=3, stride=2, bias=False),
                norm_layer(2 * chnls),
                nn.LeakyReLU() )
            setattr(self, "conv" + str(i), conv)
            # Шари upsampling (відновлення розмірності через skip connections)
            deconv = nn.Sequential(
                nn.ConvTranspose2d(4 * chnls, chnls, kernel_size=3, stride=2, bias=False),
                norm_layer(chnls),
                nn.LeakyReLU())
            setattr(self, "deconv" + str(i), deconv)
            chnls *= 2
        # Середній блок обробки
        self.conv_mid = nn.Sequential(nn.Conv2d(chnls, mid_layer_chnls, kernel_size=4, stride=4, bias=False),
                                     norm_layer(mid_layer_chnls),
                                     nn.LeakyReLU() )
        # Фінальний шар
        self.deconv_last = nn.ConvTranspose2d(
            first_layer_chnls, last_layer_chnls, kernel_size=3, stride=2, bias=True)
```

Рисунок 3.1 – Модель нейромережі

Основний обчислювальний процес реалізовано у вигляді циклу ітеративної оптимізації з використанням алгоритму Adam. На кожному кроці циклу виконується стохастична вибірка променів, для яких здійснюється прямий волюметричний рендеринг з використанням поточної оцінки пози та моделі сцени. Отримані значення кольорів пікселів порівнюються з еталонним зображенням за допомогою функції втрат (MSE), після чого через механізм зворотного поширення помилки обчислюються градієнти відносно параметрів пози, і оптимізатор виконує крок оновлення, наближаючи віртуальну камеру до шуканого положення. Протягом усього процесу програма динамічно розраховує метрику PSNR для контролю якості збіжності, а по завершенню генерує фінальні зображення та графіки, що демонструють траєкторію зменшення похибки.

### Висновки за розділом 3

В розділі 3 було розглянуто програмну реалізацію методу iNeRF для розв'язання оберненої задачі 3D-візуалізації, що включає обґрунтування вибору інструментарію та детальний опис розробленого програмного забезпечення. Програма написана універсальною мовою програмування Python із використанням фреймворку PyTorch, який забезпечує автоматичне диференціювання та апаратне прискорення обчислень на GPU. Реалізована структура дозволяє ефективно виконувати ітеративну оптимізацію пози камери, використовуючи допоміжні бібліотеки для обробки зображень та візуалізації результатів роботи алгоритму.



Рисунок 3.2 – Модель нейромережі

## 4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

### 4.1 Навчання моделі

Вхідні дані та гіперпараметри для проведення обчислювального експерименту з оптимізації пози (iNeRF) були обрані наступним чином: розмір вхідного та цільового зображень становить  $128 \times 128$  пікселів, фокусна відстань віртуальної камери – 131.25, а радіус розміщення камери відносно центру сцени дорівнює 1.3. Межі глибини рендерингу (near/far) встановлено в інтервалі  $[0.8; 1.8]$ . Для оптимізації використовується алгоритм Adam зі швидкістю навчання (learning rate) 0.01, при цьому кількість променів у пакеті (batch size) для однієї ітерації складає 1024. Умовою зупинки процесу обрано досягнення цільового порогу якості у 28.0 dB.

Процес навчання, який у даному контексті є ітеративною оптимізацією параметрів пози камери при фіксованих вагах нейронної мережі, розпочався з початкової ініціалізації. На старті модель мала початкову здогадку про позу, яка суттєво відрізнялася від істинної, що характеризувалося високим значенням функції втрат ( $MSE\ Loss = 0.02080$ ) та низьким показником PSNR на рівні 16.82 dB. Протягом перших 90 епох відбувалася фаза активної збіжності, під час якої параметри камери найбільш суттєво наближалися до цільових значень: похибка зменшилася майже в 5 разів (до 0.00426), а якість реконструкції зросла до 23.71 dB.

У подальшому, після 100-ї епохи, алгоритм перейшов до точного налаштування, мінімізуючи залишкові відхилення, і вже на 120-й ітерації втрати склали 0.00201, а PSNR наблизився до 27 dB. Процес оптимізації було автоматично зупинено на 145-й епосі після виконання умови завершення. Наприкінці експерименту модель досягла значення функції втрат 0.00155 та цільового значення PSNR 28.10 dB, що свідчить про успішну локалізацію камери.

## 4.2 Оцінка якості моделі

По закінченню процесу оптимізації було проведено комплексний аналіз якості роботи моделі шляхом порівняння отриманих результатів із еталонними даними. Оцінка виконувалася у двох основних аспектах: візуальна якість синтезованого зображення та геометрична точність знайденої пози камери.

Першим елементом аналізу є динаміка метрик збіжності. На рисунку 4.1 наведено графіки зміни функції втрат MSE та метрики PSNR протягом процесу навчання. Як видно з графіків, алгоритм демонструє стабільну збіжність без значних коливань. Значення MSE стрімко падає на перших 30 ітераціях, що свідчить про швидку корекцію грубої похибки початкової ініціалізації. Фінальне значення PSNR досягло 28.10 dB, що перевищує встановлений поріг успішності та свідчить про високу візуальну відповідність результату.

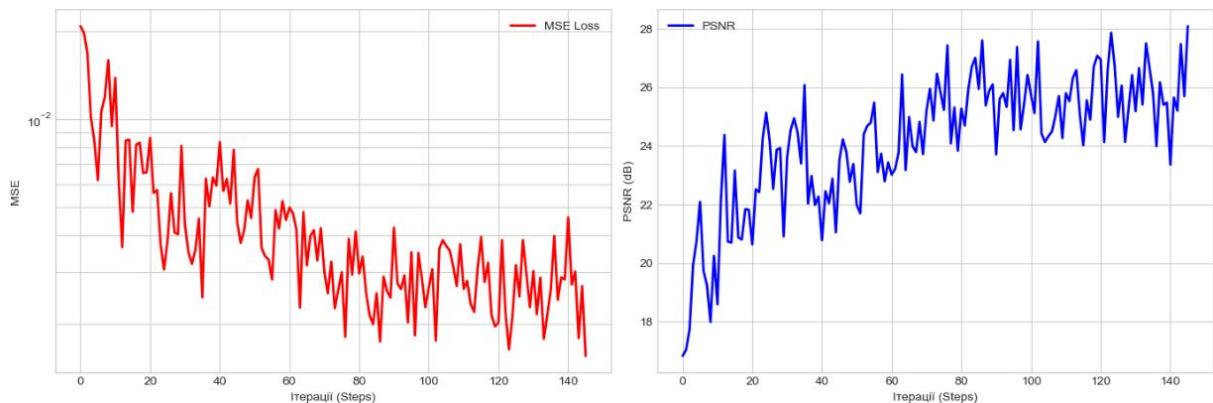


Рисунок 4.1 – Динаміка функції втрат (MSE) та якості (PSNR) під час оптимізації

Другим, і найбільш наочним елементом оцінки, є візуальне порівняння зображень. На рисунку 4.2 представлена еволюція синтезованого вигляду сцени в процесі роботи алгоритму. На початковому етапі зображення є розмитим і зміщеним, що відповідає невірній початковій позі. Однак вже до 90-ї ітерації контури об'єкта (автомобіля) стають чіткими і займають правильне положення в кадрі. Карта похибок (у другому рядку рисунка 4.2) демонструє, як абсолютна

різниця між пікселями синтезованого та цільового зображень зменшується від яскраво виражених зон розбіжності до майже повної відсутності помилок (темний фон) на фінальному кроці.

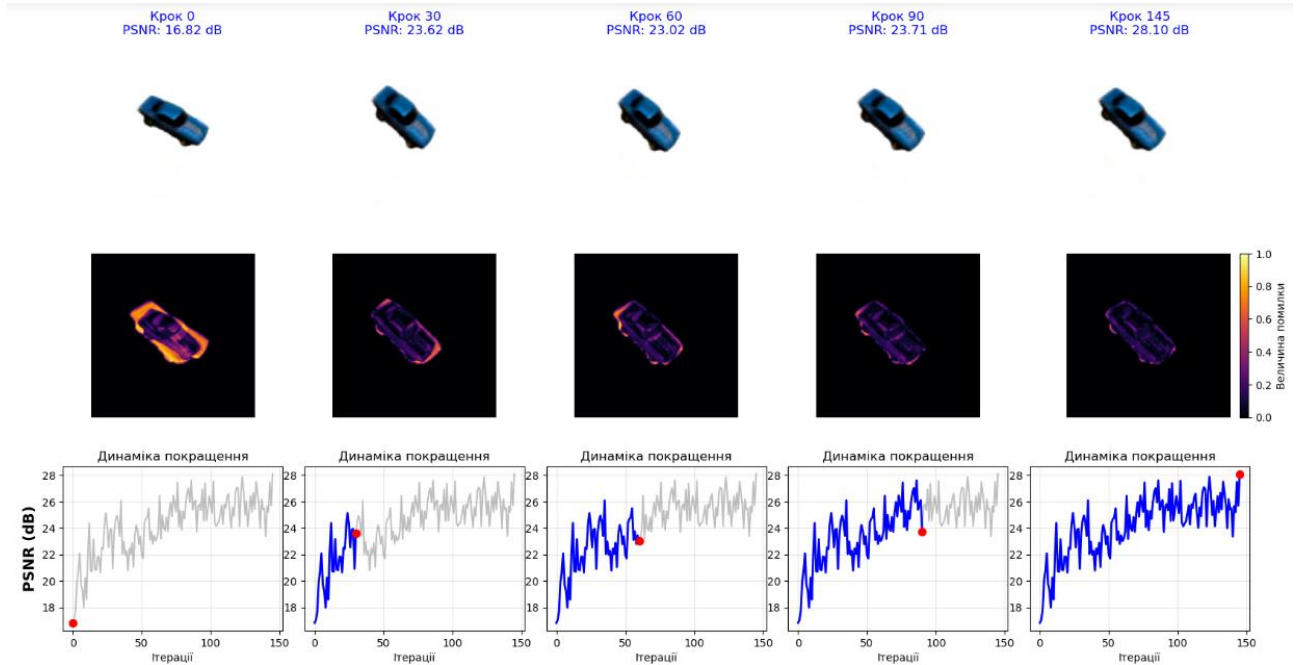


Рисунок 4.2 – Еволюція візуальної якості: рендер (верхній ряд), карта похибок (середній ряд) та траєкторія PSNR (нижній ряд)

Третім аспектом є точність оцінки пози. Хоча задача формулювалася як мінімізація фотометричної похибки на зображенні, кінцевою метою було відновлення геометричних параметрів камери. Порівняння траєкторії оптимізації з істинною позою показує, що метод iNeRF здатен ефективно "стягувати" камеру до правильного положення, використовуючи лише градієнти від різниці пікселів, навіть при значному початковому зміщенні. Це підтверджує стійкість методу до похибок ініціалізації в межах дослідженого діапазону.

#### Висновки за розділом 4

В даному розділі було проведено аналіз результатів обчислювального експерименту з відновлення пози камери. Із отриманих графіків метрик та

візуалізації видно, що алгоритм демонструє стабільну динаміку збіжності, а синтезоване зображення на фінальних ітераціях має високу візуальну схожість із еталонним.

З цих результатів можна зробити висновок, що метод iNeRF та парадигма «аналізу через синтез» заслуговують на увагу при вирішенні обернених задач 3D-візуалізації, особливо у випадках, коли доступна лише неявна нейронна презентація сцени без точної геометричної моделі. Сама по собі технологія дозволяє ефективно знаходити просторові параметри камери, хоча недоліком є чутливість методу до початкової ініціалізації (ризик потрапляння в локальні мінімуми) та значна обчислювальна ємність процесу диференційованого рендерингу.

## ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-прикладну задачу розв'язання обернених задач 3D-візуалізації за допомогою нейронних мереж. Результати, отримані у ході проведення дослідження, показують, що застосування методу інверсії нейронних полів випромінювання (iNeRF) для визначення пози камери забезпечує високу точність та стабільну збіжність. Експериментально підтверджено, що використання підходу «аналізу через синтез» у поєднанні з диференційованим волюметричним рендерингом дозволяє досягти значення метрики якості PSNR на рівні, що свідчить про високу візуальну та геометричну відповідність отриманого розв'язку еталонним даним. Одержані результати відповідають сучасному рівню світових наукових знань у галузі нейронного рендерингу та комп'ютерного зору.

Розроблений програмний комплекс на базі бібліотеки PyTorch може бути впроваджений у системи, де критично важливим є точне позиціювання у просторі на основі візуальних даних. Можливими галузями використання результатів роботи є робототехніка (для навігації автономних агентів у складних середовищах), індустрія доповненої та віртуальної реальності (для коректного накладання віртуальних об'єктів на реальний світ), а також сфера 3D-сканування та фотограмметрії.

Наукова значущість роботи полягає в обґрунтуванні ефективності використання неявних нейронних представлень сцени замість класичних полігональних моделей для розв'язання геометричних задач, що дозволяє працювати зі складними сценами, де традиційні методи на основі ключових точок є неефективними. Науково-технічна значущість роботи підтверджується реалізацією стійкого до шумів алгоритму оптимізації на графічних процесорах.

Доцільність продовження досліджень за відповідною тематикою обумовлена стрімким розвитком технологій Neural Radiance Fields. Перспективними напрямками подальшої роботи є оптимізація швидкодії алгоритму для забезпечення роботи в режимі реального часу, адаптація методу для динамічних сцен,

що змінюються у часі, а також дослідження можливостей спільної оптимізації пози камери та умов освітлення.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сизченко М. Д. Використання нейронної мережі для реконструкції 3D сцени з 2D зображень. *28-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті»* : зб. матеріалів форуму. Т. 7. Харків : ХНУРЕ, 2024. С. 278–279.
2. Glassner A. S. An introduction to ray tracing. *Academic press*, 1989.
3. State-of-the-Art on Neural Rendering / A. Tewari, O. Sira, J. Thies, C. Theobalt. *Computer Graphics Forum*. 2020. Vol. 39, No. 2. P. 703–726.
4. Hadamard J. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*. 1902. Vol. 13, № 3. P. 49–52.
5. Schönberger J. L., Frahm J. M. Structure-from-Motion Revisited. *Conference on Computer Vision and Pattern Recognition*, (Las Vegas, 27-30 June 2016). 2016. P. 104–113.
6. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms / S. M. Seitz, B. Curless, J. Diebel, D. Scharstein. *Conference on Computer Vision and Pattern Recognition*, (New York, 17-22 June 2006). 2006. Vol. 1. P. 519–526.
7. Horn B. K. Obtaining shape from shading information. *The psychology of computer vision*, 1975. P. 115–155.
8. Woodham R. J. Photometric method for determining surface orientation from multiple images. *Optical Engineering*. 1980. Vol. 19, No. 1. P. 139–144.
9. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016. 800 p.
10. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction / C. B. Choy, D. Grman, J. Gwak, M. Han. *European Conference on Computer Vision*, (Amsterdam, 8-16 October 2016). 2016. Part IV. P. 628–644.
11. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling / J. Wu, C. Zhang, T. Xue, B. Freeman. *Advances in Neural Information Processing Systems*, (Barcelona, 5-10 December 2016). 2016. P. 82–90.

12. Kato H., Ushiku Y., Harada T. Neural 3D Mesh Renderer. *Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, 18-23 June 2018). 2018. P. 907–916.
13. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis / B. Mildenhall, P. P. Srinivasan, M. Tancik, R. Ramamoorthi. *European Conference on Computer Vision*, (Glasgow, 23-28 August 2020). 2020. Part I. P. 400–416.
14. iNeRF: Inverting Neural Radiance Fields for Pose Estimation / L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez. *International Conference on Intelligent Robots and Systems*, (Prague, 27 September - 1 October 2021). 2021. P. 942–949.
15. BARF: Bundle-Adjusting Neural Radiance Fields / C. Lin, W. Ma, A. Torralba, S. Lucey. *International Conference on Computer Vision*, 2021. P. 741–751.
16. Barron J. T., Mildenhall B., Tancik M. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. *International Conference on Computer Vision*. 2021. P. 855–864.
17. pixelNeRF: Neural Radiance Fields from One or Few Images / A. Yu, V. Ye, M. Tancik, A. Kanazawa. *Conference on Computer Vision and Pattern Recognition*, 2021. P. 578–587.
18. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding / T. Müller, A. Evans, C. Schied, A. Keller. *ACM Transactions on Graphics*, 2022. Vol. 41, No. 4. P. 1–15.
19. 3D Gaussian Splatting for Real-Time Radiance Field Rendering / B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis. *ACM Transactions on Graphics*, 2023. Vol. 42, No. 4. P. 1–14.
20. Martin-Brualla R., Radwan N., Sajjadi M. S. M. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *Conference on Computer Vision and Pattern Recognition*. 2021. P. 7210–7219.
21. Wang P., Liu L., Liu Y. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Advances in Neural Information Processing Systems*. 2021. Vol. 34. P. 7171–7183.

22. Chen A., Xu Z., Geiger A. TensorRF: Tensorial Radiance Fields. *European Conference on Computer Vision*. 2022. P. 333–350.
23. Tancik M., Casser V., Yan X. Block-NeRF: Scalable Large Scene Neural View Synthesis. *Conference on Computer Vision and Pattern Recognition*. 2022. P. 8248–8258.
24. Xie Y., Takikawa T., Saito S. Neural Volume Rendering: NeRF and Beyond. *Foundations and Trends in Computer Graphics and Vision*. 2022. Vol. 14, P. 1–102.
25. VolSDF: Volume Rendering of Signed Distance Fields for 3D Geometry / L. Yariv, J. Gu, Y. Kaza, N. Snavely. *Advances in Neural Information Processing Systems*. 2021. Vol. 34. P. 6009–6019.
26. Sun J., Shen Z., Wang Y. LoFTR: Detector-Free Local Feature Matching with Transformers. *Conference on Computer Vision and Pattern Recognition*. 2021. P. 8922–8931.
27. D-NeRF: Neural Radiance Fields for Dynamic Scenes / A. Pumarola, E. Corona, G. Pons-Moll, F. Moreno-Noguer. *Conference on Computer Vision and Pattern Recognition*. 2021. P. 10318–10327.
28. Chen X., Zhang H., Liu Y. Direct-PoseNet: Absolute Pose Regression with Photometric Consistency. *International Conference on 3D Vision*. 2021. P. 112–121.
29. Depth-supervised NeRF: Fewer Views and Faster Training for Free / K. Deng, A. Liu, J. Zhu, D. Ramanan. *Conference on Computer Vision and Pattern Recognition*. 2022. P. 12882–12891.
30. Barron J. T., Mildenhall B., Verbin D. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *International Conference on Computer Vision*. 2023. P. 19697–19705.
- 31 Ravi N., Reizenstein J., Novotny D. PyTorch3D: A library for 3D deep learning. *arXiv preprint arXiv:2007.08501*. 2020.