

ГЕНЕРАТИВНІ АЛГОРИТМИ ТРИВИМІРНОГО МОДЕЛЮВАННЯ: ПРИНЦИПИ НАВЧАННЯ ТА ІНТЕЛЕКТУАЛЬНИЙ АСИСТЕНТ

Христенко Андрій Володимирович

студент групи МППм-1-25-1.4-д

Київський столичний університет імені Бориса Грінченка, м. Київ

науковий керівник – д.т.н., проф. Бондарчук А.П.

ВСТУП

Актуальність дослідження зумовлена поширенням генеративного штучного інтелекту в комп'ютерній графіці та потребою скорочення трудомісткості створення тривимірного контенту. Сучасні алгоритми здатні генерувати геометрію, матеріали й візуальні представлення сцен, однак отримані результати не завжди мають коректну топологію та готові до безпосереднього редагування. Додатковою проблемою є обмежена інтеграція генеративних засобів у робочий процес 3D-редактора: користувачеві часто доводиться самостійно обирати інструменти, виправляти артефакти й виконувати повторювані операції.

Мета дослідження — узагальнити принципи функціонування та навчання генеративних алгоритмів тривимірного моделювання й визначити архітектуру інтелектуального асистента, здатного аналізувати контекст дій користувача, надавати рекомендації та підтримувати виконання типових операцій у 3D-редакторі.

У наукових працях генеративно-змагальні мережі (GAN) застосовують для навчання розподілу тривимірних форм і створення текстурованих об'єктів; зокрема, GET3D орієнтована на генерацію геометрії, придатної до подальшої обробки [1]. Дифузійні моделі навчаються відновлювати структуру даних із поступово зашумленого представлення, а DreamFusion використовує попередньо навчену 2D-дифузійну модель для оптимізації 3D-сцени за текстовим описом [2; 3]. NeRF навчає неперервне представлення сцени на основі зображень із різних ракурсів [4; 7]. Попри прогрес цих напрямів, генерація контенту та інтелектуальний супровід дій користувача залишаються різними задачами, що потребують окремої архітектури й критеріїв оцінювання.

КОРОТКИЙ ОПИС ДОСЛІДЖЕННЯ, ЙОГО МЕТОДІВ І ЗАСОБІВ

Застосовано аналіз наукових публікацій, порівняння принципів навчання генеративних моделей і системний підхід до проектування програмного асистента. Розглянуто три групи підходів: GAN для генерації форм, дифузійні моделі для синтезу за умовою та нейронні поля для реконструкції сцени. Для майбутнього прототипу як середовище інтеграції обрано Blender, оскільки його Python API дає змогу отримувати дані про сцену й автоматизувати окремі операції [5]. Поточний етап роботи має концептуально-проектний характер: архітектура асистента та план його навчання визначені, однак повноцінне навчання власної моделі й експериментальна перевірка ще мають бути виконані.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Навчання генеративної моделі починається з підготовки набору даних і визначення формату цільового представлення. Для полігональних моделей потрібно узгодити масштаб, орієнтацію, геометрію та, за потреби, матеріали. Для моделей, що працюють із зображеннями, важливими є різноманітність ракурсів і відповідність зображень об'єктам. Дані розділяють на навчальну, валідаційну й тестову частини, щоб оцінювання проводилося на прикладах, які не використовувалися для оптимізації параметрів.

У GAN генератор формує синтетичний приклад із латентного вектора, а дискримінатор оцінює його відповідність навчальним даним. Параметри мереж оновлюються по чергово: генератор навчається створювати правдоподібніші приклади, а дискримінатор — краще розрізняти реальні та згенеровані дані [1]. У дифузійній моделі до навчального прикладу поступово додається шум, після чого нейронна мережа навчається прогнозувати шум або пов'язану з ним величину. Під час генерації виконується зворотний процес, у якому шум послідовно перетворюється на структурований результат [2]. У NeRF мережа оптимізує параметри поля кольору й щільності, мінімізуючи різницю між синтезованими та реальними зображеннями сцени [4].

Ці методи відрізняються типом результату й критеріями якості. GAN і дифузійні моделі можуть застосовуватися для синтезу нових об'єктів, тоді як NeRF переважно орієнтований на відтворення сцени з нових ракурсів. Тому оцінювання 3D-генерації має враховувати не лише візуальну правдоподібність, а й точність геометрії, якість топології, наявність артефактів, відповідність вхідному опису та обсяг ручного доопрацювання.

Таблиця 1

Порівняння основних підходів до генерації та реконструкції 3D-контенту

Підхід	Принцип навчання	Типовий результат	Основне обмеження
GAN / GET3D	Змагальна оптимізація генератора й дискримінатора	Тривимірна форма та текстурована геометрія	Можлива нестабільність навчання; потрібно перевіряти топологію
Дифузійні моделі	Прогнозування шуму та зворотне відновлення даних	Синтез за текстом або іншою умовою	Ітеративна генерація може бути ресурсомісткою
NeRF	Оптимізація поля сцени за помилкою відтворення зображень	Неперервне представлення сцени й нові ракурси	Результат не обов'язково є готовою полігональною сіткою

Для реалізації асистента запропоновано модульну структуру. Модуль збору контексту отримує стан сцени, активний об'єкт, режим редагування та послідовність останніх операцій. Модуль попередньої обробки перетворює події на узгоджений набір ознак. Модуль визначення наміру оцінює ймовірну задачу користувача, наприклад оптимізацію сітки або застосування матеріалу.

Далі модуль політики рішень зіставляє визначену задачу з перевіреними сценаріями та перевіряє дозволи й передумови виконання. Результатом є рекомендація або контрольований виклик функції редактора.

На першому етапі асистент не потребує навчання великої мовної моделі. Базову версію можна реалізувати на правилах і невеликій базі сценаріїв, а зібрані приклади використовувати для навчання класифікатора намірів. Один навчальний приклад має містити ознаки контексту, клас задачі, запропоновану дію та результат взаємодії: прийняття, відхилення чи скасування. Після накопичення даних їх потрібно перевірити, анотувати й розділити на навчальну та тестову вибірки. Модель оцінюється за точністю визначення задачі, частотою хибних рекомендацій і стабільністю на невідомих сценаріях.

Автоматичне виконання доцільно дозволяти лише для обмежених операцій із передбачуваним результатом і можливістю скасування. Якщо впевненість моделі низька, контекст неоднозначний або операція може змінити геометрію небажаним чином, асистент має запитувати підтвердження. Зворотний зв'язок не слід трактувати як безумовний сигнал для автоматичного перенавчання: перед оновленням модель необхідно перевіряти на контрольному наборі, щоб уникнути погіршення раніше реалізованих функцій.

РОБОЧИЙ ЦИКЛ
Дії користувача та стан сцени → Збір контексту → Визначення наміру
↓
Перевірка сценарію й дозволів → Рекомендація або безпечне виконання
↓
Зворотний зв'язок: прийнято / відхилено / скасовано
НАВЧАЛЬНИЙ ЦИКЛ
Відібрані приклади → Анотування → Поділ даних → Навчання класифікатора
↓
Тестування → Аналіз помилок → Контрольоване оновлення моделі

Рис.1. Схема роботи та вдосконалення інтелектуального асистента

План реалізації передбачає: 1) визначення типових сценаріїв; 2) створення інтеграції з Blender та отримання контексту сцени; 3) реалізацію правил і бази рекомендацій; 4) збирання й маркування прикладів; 5) навчання та оцінювання класифікатора намірів; 6) порівняльне тестування роботи з асистентом і без нього. Ефективність потрібно перевіряти за часом виконання завдань, точністю рекомендацій, частотою скасувань і якістю кінцевих моделей. До отримання експериментальних даних ці показники є планом оцінювання, а не заявленими результатами.

ВИСНОВКИ

Розглянуто принципи навчання GAN, дифузійних моделей і NeRF та визначено відмінності між генерацією нових 3D-об'єктів і реконструкцією сцен за зображеннями. Практичну якість результату слід оцінювати за геометричною коректністю, топологією, наявністю артефактів і потребою в ручному доопрацюванні. Запропоновано модульну архітектуру інтелектуального асистента й план його навчання на маркованих прикладах взаємодії з 3D-редактором. Подальша робота передбачає створення прототипу, формування навчальної вибірки та експериментальну перевірку впливу асистента на швидкість і якість виконання типових операцій.

ДЖЕРЕЛА

1. Gao J., Shen T., Wang Z. та ін. GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. Advances in Neural Information Processing Systems. 2022. URL: <https://arxiv.org/abs/2209.11163>
2. Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models. 2020. URL: <https://arxiv.org/abs/2006.11239>
3. Poole B., Jain A., Barron J. T., Mildenhall B. DreamFusion: Text-to-3D using 2D Diffusion. 2022. URL: <https://arxiv.org/abs/2209.14988>
4. Mildenhall B., Srinivasan P. P., Tancik M. та ін. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. European Conference on Computer Vision (ECCV). 2020. URL: <https://arxiv.org/abs/2003.08934>
5. Бондарчук, А. П., Мельник, І. Ю., Суханевич, Є. І., & Абрамов, В. О. (2025). Підвищення ефективності систем відеоспостереження за рахунок гібридного методу відбору ключових кадрів і інтерпретації рішень. Телекомунікаційні та інформаційні технології, (3), 101-105.
6. Blender Foundation. Blender Python API Documentation. URL: <https://docs.blender.org/api/current/>
7. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., Hlushak, O., Zhyltsov, O., Ilich, L., Kobets, N., Kovaliuk, T., Kuchakovska, H., Lytvyn, O., Lytvyn, P., Mashkina, I., Morze, N., Nosenko, T., Proshkin, V., Radchenko, S., & Yaskevych, V. (2021). Theoretical and practical aspects of the use of mathematical methods and information technology in education and science. URL: <https://doi.org/10.28925/9720213284km>