

УДК 004.94

Авсиевич Владимир Викторович, к. т.н., доцент кафедры "инженерии
искусственного интеллекта"

*Поволжского государственного университета телекоммуникаций и
информатики
443000, Россия, Самара*

Чижов Данила Андреевич, магистрант кафедры "инженерии
искусственного интеллекта"

*Поволжского государственного университета телекоммуникаций и
информатики
443000, Россия, Самара*

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ СОЗДАНИЯ СИНТЕТИЧЕСКИХ ДАННЫХ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Аннотация. В статье проводится сравнительный анализ четырех основных методов создания 3D-моделей для создания синтетических данных: параметрического, полигонального, цифровой скульптуры и генеративного ИИ-моделирования. Данные методы будут сравниваться в качестве и скорости создания 3D-моделей.

Ключевые слова: синтетические данные, 3D-моделирование, нейронные сети, Houdini, Blender, Zbrush, генеративное ИИ-моделирование.

Abstract. This article provides a comparative analysis of four main methods for creating 3D-models for generating synthetic data: parametric, polygonal, digital sculpting, and generative AI modeling. These methods will be compared in terms of the quality and speed of 3D-model creation.

Keywords: synthetic data, 3D-modeling, neural networks, Houdini, Blender, Zbrush, generative AI-modeling.

ВВЕДЕНИЕ

Сегодня нейронные сети уже практически стали стандартом в разных сферах деятельности. Например, машинное зрение помогает врачам находить скрытые патологии на ранних стадиях, замечая детали, которые сложно уловить человеческому глазу. Эта технология повышает точность анализа медицинских снимков (МРТ, КТ), уменьшая риск диагностических ошибок. Похожие технологии отвечают за безопасность, узнавая лица в толпе или номера машин на дорогах. Тот же принцип распознавания объектов позволяет в супермаркетах внедрять поиск товаров по фото, а разработчикам беспилотников — создавать «цифровое зрение», без которого автопилоты и роботы не смогли бы ориентироваться в пространстве.

Однако для нормального обучения таких систем нужны огромные массивы данных, которых в открытом доступе всегда не хватает. Чтобы решить эту проблему, разработчики начали создавать эти данные искусственно — так появилась технология синтетических данных [1]. В этой работе будут рассмотрены четыре способа моделирования таких данных и приведена таблица сравнения их между собой по качеству и скорости создания.

ПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

В отличие от классического моделирования, параметрический подход строится на алгоритмах и конкретных вводных данных — математических значениях, корреляциях или зависимостях. К примеру, в Houdini [2] это реализовано через систему нод (узлов), где каждый блок хранит свою часть информации: геометрию, координаты или атрибуты цвета. Все ноды связаны в единую цепь, поэтому можно в любой момент вернуться в самое начало, подкрутить один параметр, и вся модель автоматически перестроится под новые условия.

Конечно, у такой гибкости есть и обратная сторона: если неосторожно изменить базовую ноду, вся последующая геометрия может «поплыть» или непредсказуемо исказиться, так что структура сети должна быть продуманной. Зато в плане производительности Houdini гораздо лучше обычного софта. Там, где классический редактор не сможет вручную расставить тысячи объектов, Houdini мгновенно раскидает их через процедурные алгоритмы (рис. 1). Именно поэтому софт считается стандартом для сложных симуляций и спецэффектов, где нужно ворочать миллионами полигонов без вылетов и тормозов.

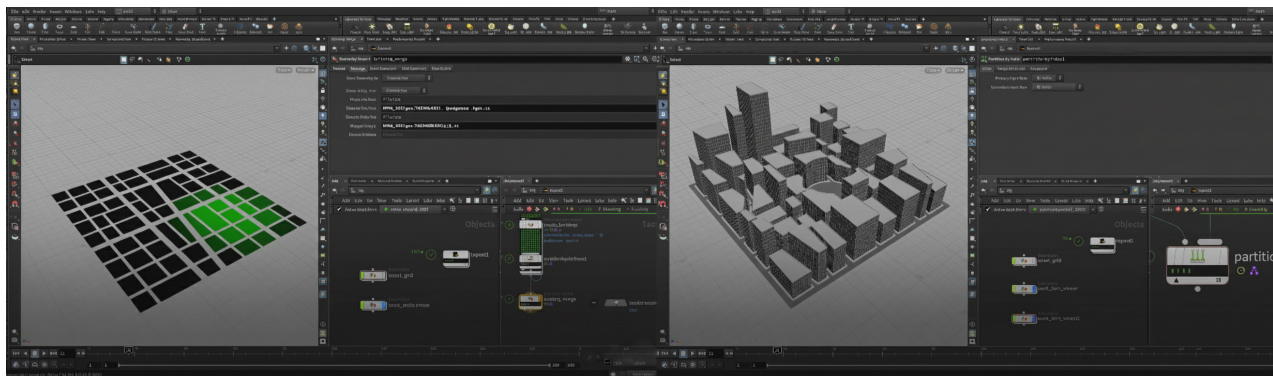


Рис. 1 – Параметрическое моделирование в Houdini

ПОЛИГОНАЛЬНОЕ МОДЕЛИРОВАНИЕ

Полигональное моделирование — это, по сути, «ручное моделирование» 3D-объектов. Весь процесс строится на работе с базовыми элементами: вершинами (vertices), ребрами (edges) и гранями (polygons).

В отличие от того же параметрического моделирования (где всё задается цифрами и формулами), здесь мы меняем геометрию напрямую. В Blender или других редакторах для этого есть стандартный набор инструментов:

- Extrude (выдавливание) — «вытягивание» новой геометрии из грани;
- Loop Cut (петля) — добавление новых разрезов для уточнения формы;
- Bevel (фаска) — смягчение острых углов;
- Knife (нож) — произвольная нарезка полигонов.

Метод очень гибкий и наглядный — сразу видно результат и можно быстро набросать сложную форму (рис. 2). Но есть и большой минус: нелинейность. Если внесено множество правок и закрыт доступ к изменениям, то вернуться к исходным параметрам или легко поменять базовые размеры уже не получится. В большинстве случаев приходится переделывать часть работы заново.

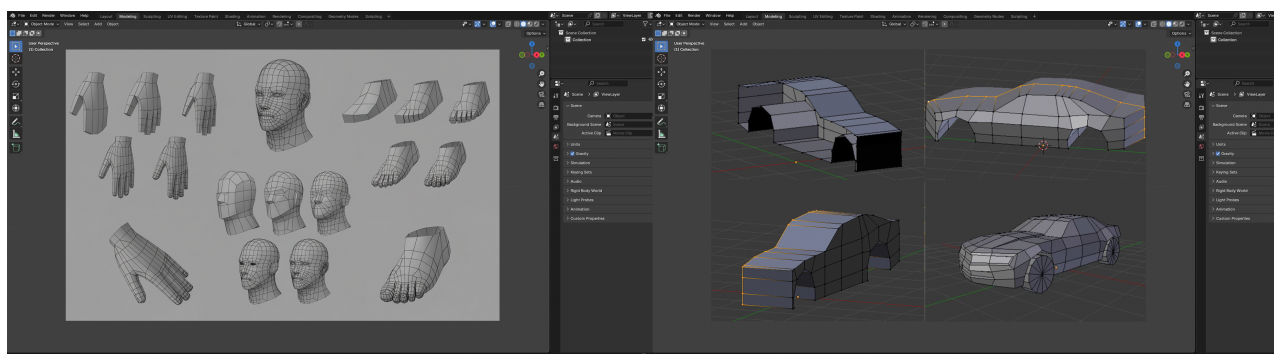


Рис. 2 – Полигональное моделирование в Blender

ЦИФРОВАЯ СКУЛЬПТУРА

Цифровая скульптура, реализованная в таких программах, как ZBrush, по своей логике ближе к ручной лепке из глины, чем к черчению [3]. В отличие от полигонального моделирования, где приходится работать с каждой точкой и ребром отдельно, здесь используются кисти, позволяющие свободно наращивать и изменять массу объекта. Это дает значительное преимущество в скорости при создании сложной органики — лиц, анатомии персонажей или складок ткани, — которые собирать вручную по полигонам было бы слишком долго. Скульптинг позволяет оперировать миллионами полигонов для проработки микродеталей, таких как поры кожи или мелкие трещины, предоставляя полную творческую свободу без необходимости следить за технической чистотой сетки на ранних этапах.

Однако за интуитивность приходится платить отсутствием инженерной точности, из-за чего метод не подходит для задач промышленного дизайна, где важны строгие размеры. Кроме того, высокая детализация делает объекты крайне тяжелыми, что перегружает железо и замедляет рендеринг. Главным

же ограничением является хаотичная структура полученной сетки: она непригодна для анимации, так как при движении будут возникать визуальные дефекты. По этой причине скульптинг нельзя считать финальным этапом. Он служит эффективным инструментом для создания визуального образа и детализации, после чего модель обязательно проходит через ретопологию — процесс создания правильной и легкой сетки, адаптированной для дальнейшего использования в симуляциях или дата сетках (рис. 3).

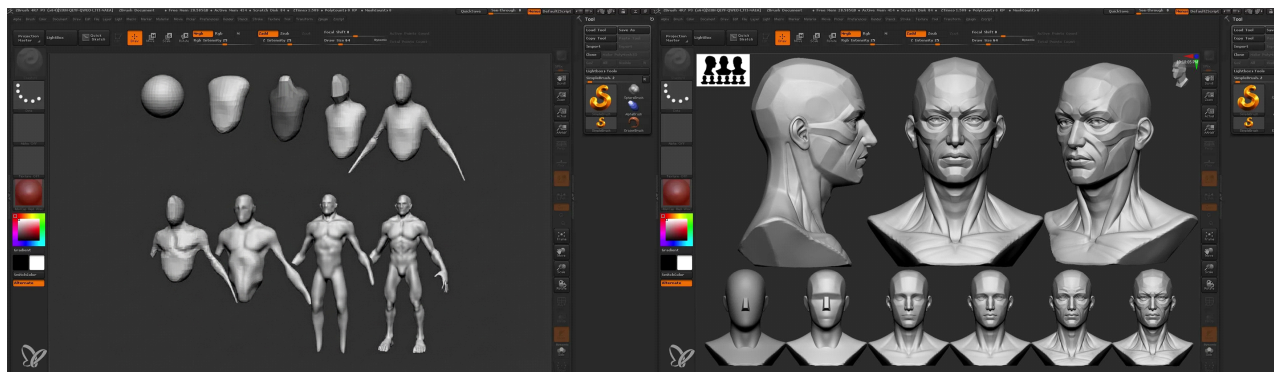


Рис. 3 – Цифровое скульптурирование в Zbrush

ГЕНЕРАТИВНОЕ ИИ-МОДЕЛИРОВАНИЕ

Генеративное ИИ моделирование — это технология создания объемных объектов на основе анализа закономерностей в больших массивах данных. Вместо ручной расстановки полигонов алгоритм обучается восстанавливать форму и структуру поверхности по текстовому описанию или плоскому изображению, фактически «достраивая» невидимые части объекта.

На практике, как в случае с системой Hunyuan3D, этот процесс делится на два этапа [4]. Сначала модель выстраивает геометрию («скелет») с помощью диффузионного трансформера. На этой стадии важно получить правильные пропорции и детальную полигональную сетку. Затем в работу вступает модуль текстурирования. Его задача — не просто раскрасить модель, а создать полноценные PBR-материалы (карты нормалей и шероховатости). При этом система выполняет «делайтинг»: убирает с исходного фото тени и блики, чтобы объект корректно смотрелся при любом освещении в 3D-редакторе [5].

Такой подход позволяет за 20–30 секунд получить цифровой актив с разрешением до 1536^3 вокселей. Модели на выходе имеют оптимизированную топологию, что позволяет сразу использовать их в игровых движках, или же в различных веб-проектах без внесения долгих правок в топологию моделей.

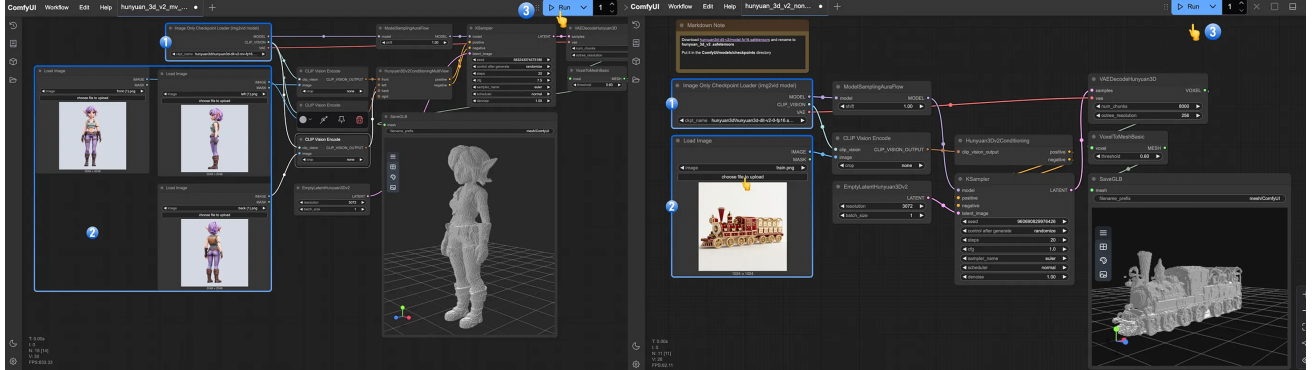


Рис. 4 – Генеративное ИИ-моделирование в Hunyuan3D

СРАВНЕНИЕ

После ознакомления со всеми методами генерации объектов, можем провести сравнительный анализ со всеми их преимуществами и недостатками, дополнительно описав, где лучше всего их можно использовать (табл. 1).

Таблица 1 – Сравнительный анализ методов создания синтетических данных

Способ моделирования	Преимущества	Недостатки	Возможные способы использования
Параметрическое моделирование	<ul style="list-style-type: none"> Высокая степень контроля над параметрами, распределениями и 	<ul style="list-style-type: none"> Большие нод-графы сложно читать и отлаживать. Изменение в ранних нодах может 	<ul style="list-style-type: none"> Генерация структурированных синтетических данных; Создание сложных сцен, VFX;

	<p>корреляция ми;</p> <ul style="list-style-type: none"> • Возможность вернуться к любому шагу и изменить параметр с автоматическим перестроением всей модели; • Удобно для утверждения изменений в коллективе. 	<p>непредсказуемо изменить весь объект.</p> <ul style="list-style-type: none"> • Требуется программистского мышления: нужно понимание логики и процедурного подхода. 	<ul style="list-style-type: none"> • Процедурные ландшафты и архитектура.
<p>Полигональное моделирование</p>	<ul style="list-style-type: none"> • Интуитивное и прямое манипулирование геометрией (вершины, ребра, грани); 	<ul style="list-style-type: none"> • Сложно вернуться к ранним этапам без потерь; • Создание сложных органических форм требует 	<ul style="list-style-type: none"> • Создание финальных, чистых 3D-активов для игр, кино, анимации; • Моделирование техники, архитектуры, персонажей с

	<ul style="list-style-type: none"> • Позволяет создавать модели с четкими инженерными допусками и идеальной геометрией; • Результат пригоден для анимации, игр и эффективного рендеринга; • Подходит для любых объектов — от органики до техники. 	<p>много времени и высокого мастерства;</p> <ul style="list-style-type: none"> • Ручное создание большого количества уникальных объектов неэффективно. 	<p>анимационной топологией; этап ретопологии в пайплайне.</p>
Цифровая скульптура	<ul style="list-style-type: none"> • Быстрое прототипирование и блокинг сложных органически 	<ul style="list-style-type: none"> • Создание объектов с точными инженерными размерами затруднено; 	<ul style="list-style-type: none"> • Концеп-арт, скульптинг высокодетализированных статичных моделей,

	<p>х форм, интерфейс, приближенный к лепке;</p> <ul style="list-style-type: none"> • Работа с десятками миллионов полигонов, создание микрорельефа (поры, складки, текстуры); • Лучший выбор для персонажей, существ, скульптур. 	<ul style="list-style-type: none"> • Автогенерируемые сетки (DynaMesh) не пригодны для анимации без ретопологии. • Высокополигональные модели требуют много ресурсов. • Финальные этапы (ретопология, UV-развертка) очень трудоемки и отнимают много времени. 	<p>создание детализированных карт нормалей и смещений для последующего использования на низкополигональных моделях, этап детализации в производстве персонажей.</p>
<p>Генеративное ИИ-моделирование</p>	<ul style="list-style-type: none"> • Создание 3D-прототипа из текста за минуты, а не часы/дни; 	<ul style="list-style-type: none"> • Отсутствие контроля над топологией, точными размерами, чистотой сетки; 	<ul style="list-style-type: none"> • Быстрое прототипирование и мозговой штурм, создание фоновых или простых объектов для сцен, генерация идей и

	<ul style="list-style-type: none"> • Позволяет не-специалистам генерировать 3D-контент; • Возможность быстро генерировать множество вариаций по одному промту. 	<ul style="list-style-type: none"> • Артефакты в геометрии (плавающие детали, "размножение" элементов); • Процесс генерации одной модели требует больших мощностей GPU и времени (10-40 минут); • Стохастичность результатов, требуется ручная доработка. 	<p>референсов, области, где допустима низкая детализация и нестрогая геометрия.</p>
--	--	--	---

ЗАКЛЮЧЕНИЕ

Важность создания синтетических дата сетов обусловлена необходимостью их скоростью и качеством, для лучшего обучения на них нейронных сетей. Методы, рассмотренные в этой статье, показали отличные друг от друга результаты качественного использования для текущей проблемы. Параметрическое моделирование предоставляет возможность точного с инженерной точки зрения создания 3D-моделей, однако для этого необходим опыт работы с математическими формулами и затрачивается

огромное количество времени и энергии у работника. Хотя методы полигонального моделирования и цифровой скульптуры так же требуют значительных временных затрат, они предоставляют художникам гораздо больше свободы в создании сложных и уникальных форм по сравнению с параметрическим моделированием. Метод генеративного ИИ – моделирования отличается возможностью создания огромного количества вариаций финального объекта за короткое время. Но и недостаткам свойственным нейросетям – возможности образования артефактов на объектах и некорректного результата из-за несодержательного промта (запроса). Из всего вышесказанного следует, что использование каждого метода определяется вопросом определенного качества создания дата сетов для нейронных сетей. Если необходимы более качественные и детализированные материалы для обучения, то лучше применить три первых метода. В случае критичности временного ресурса и более низкими критериями к качеству объектов, метод генеративного ИИ-моделирования подойдет как лучшая альтернатива. При этом активное развитие нейронных сетей создает предпосылки для того, что инструменты генеративного ИИ - моделирования станут основной альтернативой классическим методам, превзойдя их по точности и эффективности.

Список источников

1. Д.А. Шлёнских, М.Л. Белокопытов, Д.В. Анохин, И.Г. Иванов. Метод синтеза данных для повышения эффективности обучения нейронных сетей. // ЖУРНАЛ РАДИОЭЛЕКТРОНИКИ. 2024. С. 2. URL: <http://jre.cplire.ru/jre/mar24/8/text.pdf> (дата обращения: 21.11.2025).
2. С.Д. Клинский. Обзор на Houdini. Логика и интерфейс. // Постулат. 2021. С. 2. URL: <https://e-postulat.ru/index.php/Postulat/article/view/3764> (дата обращения: 23.11.2025).
3. Е. Келлер. Представляем ZBrush 4. // Google Книги. 2011. С. 15. URL: <https://books.google.ru/books?hl=ru&lr=&id=DQuMCSPKFpYC&oi=fnd&pg=PP9&dq=zbrush&ots=NPxA7OmrGZ&sig=smds-Xv7vTy->

[2CbsmBkZvcohOd4&redir_esc=y#v=onepage&q&f=false](https://arxiv.org/abs/2411.02293) (дата обращения: 24.11.2025).

4. Хуанчжу Янг, Хуивэн Ши, Бовэн Чжанг. Hunyuan3D 1.0: Единая платформа для преобразования текста в 3D и изображений в 3D. // arXiv. 2024. URL: <https://arxiv.org/pdf/2411.02293> (дата обращения: 29.11.2025).

5. Б. Мильденхолл, П.П. Сринивасан, М. Танкик, Д.Т. Баррон, Р. Рамамоотхи. NeRF: Представление сцен в виде нейронных полей излучения для синтеза изображения. // arXiv. 2020. С. 3. URL: <https://arxiv.org/abs/2003.08934> (дата обращения: 29.11.2025).