

ИССЛЕДОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ АНИМАЦИЙ ПЕРСОНАЖЕЙ НА ОСНОВЕ ДАННЫХ ДВИЖЕНИЯ.

Гришель Д.Г. Лицкевич В.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Рябычина О.П. – канд. техн. наук, доцент

В данной работе рассматривается метод end-to-end обучения для генерации анимации VR-персонажей по данным с датчиков. Комбинируя motion matching и нейросети, подход использует codebook для компактного представления движений и предсказания их последовательностей. Это обеспечивает плавную анимацию (ходьба, прыжки) без задержек, превосходя традиционные методы по отзывчивости и реализму.

Создание реалистичной анимации для кино, игр и VR требует больших ресурсов, но ИИ и машинное обучение упрощают эту задачу. Методы на основе motion capture ускоряют процесс и улучшают качество движений. Однако анимация в VR остается сложной из-за необходимости высокой точности и отзывчивости в реальном времени. В этой работе исследуются методы машинного обучения, позволяющие генерировать плавную и адаптивную анимацию даже при ограниченных входных данных.

Создание виртуальных аватаров, точно отражающих движения пользователей, критически важно для иммерсивного опыта в AR/VR. В отличие от традиционных видеоигр, AR/VR требует точного воспроизведения движений на основе разреженных данных, таких как трекинг головы и контроллеров. Однако эти данные содержат мало информации о нижней части тела, что создает сложности для реалистичной анимации. Существующие методы, такие как motion matching [1] или нейронные сети, не справляются с этой задачей из-за ограничений в обработке произвольных движений и отсутствия информации о будущих действиях.

Однако существует новый подход к созданию контроллера, который сочетает качество motion matching и гибкость нейронных сетей. Контроллер — это алгоритм или модель, которая преобразует входные данные (например, сигналы от датчиков или управляющие команды) в движения виртуального персонажа. Он использует автоэнкодер [2] для сжатия данных движений в компактное представление и обучается end-to-end, предсказывая распределение вероятностей для будущих движений. Это позволяет эффективно синтезировать разнообразные движения, такие как ходьба, прыжки, танцы и спорт, на основе как трекинга, так и традиционных контроллеров, сохраняя контекст движений пользователя.

Для работы с большими наборами данных motion capture разработаны модели, которые изучают структуру движений. Motion graphs — это структура данных, которая представляет движения как узлы, соединенные переходами между похожими позами [3]. Это позволяет создавать плавные анимации, переклюкаясь между разными движениями. Однако такие модели требуют значительных ресурсов.

Генеративные модели (GPLVM, Gaussian Processes, VAE, диффузионные) создают анимации через выборку из шума, обеспечивая разнообразие и реалистичные физические реакции на воздействия. Данные модели применяются в физически реалистичных контроллерах, где пользователь может воздействовать на персонажа, вызывая реалистичные реакции.

Несмотря на способность генеративных моделей создавать новые движения, в индустрии сохраняются опасения по поводу сглаживания артефактов при смешивании движений. Для сохранения четкости движений используются методы, такие как VQVAE, которые обучают дискретное латентное пространство. Эти методы применяются для синтеза мимики, движений тела и генерации движений на основе текста или речи.

Для управления персонажами на основе моделей движений применяются методы, такие как model predictive control и reinforcement learning (RL). Однако двухэтапное обучение, при котором сначала изучается модель движений, а затем управление, может привести к неоптимальным результатам, таким как медленный отклик или неестественные переходы. Альтернативой являются методы с обучением с учителем, такие как motion matching и PFNN, которые проще в обучении и могут работать end-to-end.

Reinforcement Learning (RL, обучение с подкреплением) — это метод, при котором программа (агент) учится управлять персонажем методом проб и ошибок [4]. Она получает "награды" за правильные действия (например, плавную ходьбу) и штрафы за ошибки (падение). Со временем алгоритм находит оптимальную стратегию, но обучение может быть сложным и требовать много вычислений.

Обучение с учителем (motion matching, PFNN) — это более простые методы, где модель учится на готовых примерах движений (например, записи реальной анимации).

На рисунке 1 изображена архитектура системы под названием "Codebook Matching", которая предназначена для обучения многообразия движений (motion manifold) и выборки из него в режиме end-to-end. Система обеспечивает сходство между двумя категориальными распределениями вероятностей, заменяя одно другим, и создает латентное пространство, информированное управляющими сигналами.

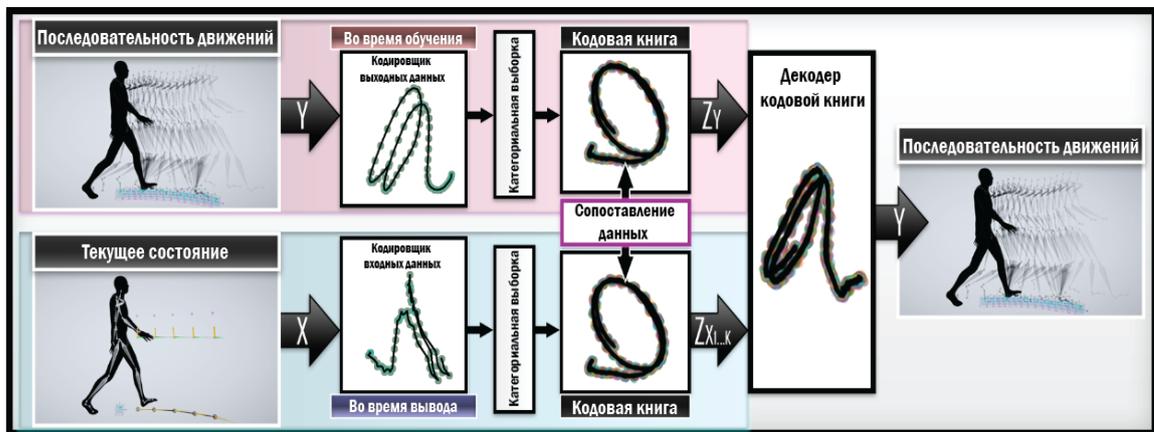


Рисунок 1 – Архитектура системы "Codebook Matching".

Основными элементами архитектуры являются Input Encoder, Output Encoder, Codebook, Category.

Input Encoder кодирует текущие управляющие сигналы (например, данные с датчиков) и выполняет целенаправленную выборку (Targeted Sampling) из кодбука (Codebook).

Output Encoder кодирует будущую последовательность движений (Future Motion Sequence) и также выполняет целенаправленную выборку из кодбука.

Codebook хранит категории движений, которые используются для сопоставления и выборки.

Category – это категория движений, которая используется для организации данных в латентном пространстве.

Задача во время вывода (Task during inference) заключается в том, чтобы на основе текущих управляющих сигналов предсказать будущую последовательность движений, используя обученное латентное пространство. Это позволяет системе эффективно генерировать реалистичные и отзывчивые движения для интерактивных приложений, таких как VR или видеоигры.

Кодбук (Codebook) — структура данных для хранения дискретных элементов (категорий, кластеров) в сжатом виде. Каждый элемент соответствует определённому паттерну (например, позе в анимации). При обучении система сопоставляет входные данные с ближайшими элементами кодбука, обеспечивая эффективное кодирование.

End-to-end обучение — подход, при котором модель обучается решать задачу целиком, без разделения на этапы. Это позволяет оптимизировать систему как единое целое, упрощая обучение и улучшая результат. В анимации end-to-end подход одновременно обрабатывает движения и управляющие сигналы, повышая точность и отзывчивость.

Метод Codebook Matching превосходит традиционные подходы (motion matching, нейронные сети, motion graphs и RL) за счет сочетания гибкости генеративных моделей и точности дискретного латентного пространства. В отличие от motion matching, который ограничен жесткими переходами между анимациями, и RL, требующего сложного обучения, Codebook Matching обеспечивает плавную и адаптивную анимацию даже при разреженных входных данных (например, только трекинг головы и рук в VR). Использование кодбука позволяет эффективно кодировать разнообразные движения и быстро выбирать подходящие паттерны, избегая артефактов сглаживания, характерных для VAE и диффузионных моделей. Кроме того, end-to-end обучение упрощает интеграцию управляющих сигналов, делая систему более отзывчивой в реальном времени, что критично для иммерсивного VR. В итоге, этот подход сохраняет реализм движений, снижает вычислительные затраты и подходит для интерактивных приложений, где важны и точность, и производительность.

Таким образом, в рамках данного исследования проведён сравнительный анализ методов генерации анимаций и доказаны преимущества подхода Codebook Matching перед альтернативными решениями.

Список использованных источников:

1. Phase-Functioned Neural Networks for Character Control / D. Holden et al. // ACM Transactions on Graphics, 2017. - P. 1-13.
2. Learning Motion Controllers with Adaptive Depth Perception / K. Lee et al. // Computer Graphics Forum, 2021. - P. 1-12.
3. Neural State Machine for Character-Scene Interactions / J. Zhang et al. // ACM Transactions on Graphics, 2019. - P. 1-14.
4. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills / X.B. Peng et al. // ACM Transactions on Graphics, 2018. - P. 1-14.