

Анализ применения в области стриминга пикселей традиционных и основанных на машинном обучении технологий сжатия видео

Киселёв А.В.
Аспирант кафедры ВМиК,
Институт информатики, математики и
робототехники
Уфимский университет науки и
технологий
Уфа, Российская Федерация
e-mail: andrey.kise15@yandex.ru

Сметанина О.Н.
Профессор кафедры ВМиК, доктор
технических наук, доцент
Институт информатики, математики и
робототехники
Уфимский университет науки и
технологий
Уфа, Российская Федерация
e-mail: smoljushka@mail.ru

Аннотация¹

Проблемой, рассматриваемой в данной статье, является применение традиционных технологий и технологий машинного обучения в процессе потоковой передачи (стриминга) пикселей.

В статье рассматриваются этапы процесса стриминга пикселей. Далее анализируется возможность усовершенствования технологии с помощью добавления технологий машинного обучения в определенные сегменты технологического конвейера. В конце статьи приводится анализ и сравнение традиционных технологий сжатия видео и технологий с использованием машинного обучения и предлагается вывод на основе проведенного сравнения.

Ключевые слова: стриминг пикселей, сжатие видео, кодек, H.265, VP9, AV1, машинное обучение, CNN, DAE, RNN.

1. Введение

Обычно процесс рендера сцены компьютерного приложения запускается на том же самом устройстве, что и выводит результат на экран, будь то desktop-приложение, приложение для мобильной ОС или консоль. Особенно это критично, в случае многопользовательских приложений, так как рендеринг каждого экземпляра приложения происходит отдельно на локальном устройстве пользователя, что отрицательно влияет на опыт взаимодействия в приложении.

Одним из способов перенести расчетную логику рендеринга на удаленное устройство является

технология стриминга пикселей (PixelStreaming). Она непрерывно кодирует полученные выходные данные в медиапоток, который проходит через легкий стек веб-сервисов. Затем пользователи могут просматривать этот широковещательный поток в стандартных веб-браузерах, работающих на других компьютерах и мобильных устройствах.

Потоковая передача пикселей предоставляет конечным пользователям множество преимуществ: гибкость запуска сложных приложений практически на любом типе устройств; возможность передать тяжелую работу по обработке данных серверу, который с ней справится; экономия средств за счет отсутствия необходимости обновлять компьютеры дома или в офисе; улучшенная производительность за счет более быстрой загрузки и более плавного взаимодействия.

В данной статье рассматривается применение технологий машинного обучения (МО) в процессе стриминга пикселей [1].

2. Анализ технологии стриминга пикселей

Потоковая передача пикселей — это процесс передачи данных о цвете и положении каждого пикселя на экране в режиме реального времени. Этот процесс используется для создания видеоизображений в реальном времени, например, в играх, видеоконференциях или других приложениях, требующих высокой частоты кадров.

Технология потоковой передачи пикселей довольно проста для понимания: пользователь передает входные данные на удаленный сервер или в облачную систему, например, печатает или нажимает на экран, а сервер, в свою очередь,

Труды X Международной научной конференции
"Информационные технологии
интеллектуальной поддержки принятия

решений", 12-14 ноября, Уфа-Баку-Чандигарх,
2024

X Международная научная конференция "Информационные технологии интеллектуальной поддержки принятия решений"
Уфа-Баку-Чандигарх, 2024

получает эти входные данные и использует их для рендеринга необходимой графики или видеокадров. Эти кадры сжимаются и кодируются перед передачей на устройство пользователя. Когда устройство пользователя получает поток данных, компьютер декодирует полученную информацию.

Несжатые данные отображаются на устройстве пользователя. Весь этот процесс очень быстрый и обеспечивает быстрый интерактивный опыт [2]. Процесс потоковой передачи пикселей включает в себя ряд этапов (рис. 1).

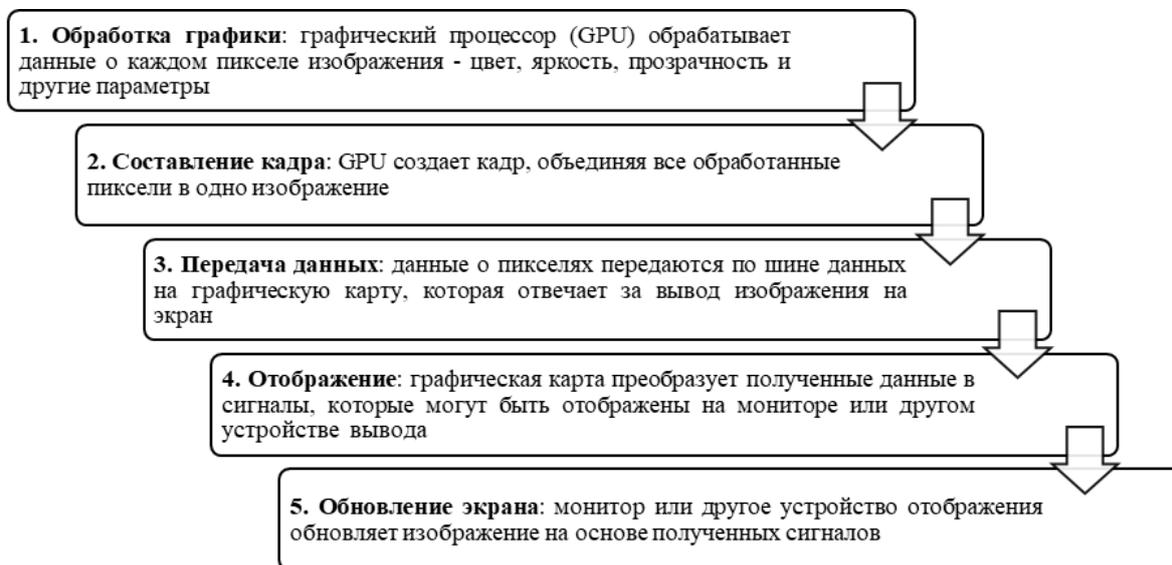


Рис. 1. Этапы процесса потоковой передачи пикселей

Процесс повторяется с частотой, которая необходима для того, чтобы видеопоток был плавным и непрерывным. Процесс также может включать в себя дополнительные этапы, например сглаживание или фильтрация, а также иные методы улучшения качества изображения.

Кроме частоты также следует учитывать ограничения шины данных и возможности графической карты, чтобы обеспечить плавную работу приложения.

Еще одним пунктом, который необходимо учитывать для плавности работы приложения, является правильный выбор основных компонентов системы потоковой передачи пикселей:

- Сетевое и интернет-подключение: необходимо для доставки входных данных и получения визуализированных данных.
- Устройство ввода на стороне пользователя: обеспечивает ввод в систему, например, клавиатура или мышь.
- Клиентское приложение: используется для доставки данных. Это может быть видеоигра, программа виртуальной реальности или

программное обеспечение для совместной работы в реальном времени.

- Механизм рендеринга: принимает входные данные клиента и преобразовывает их в нужную графику, видео или другой тип данных в реальном времени.
- Система кодирования: осуществляет сжатия и кодирование данных перед отправкой конечному пользователю.
- Сервер потоковой передачи: доставляет обработанные данные на устройство конечного пользователя.

Некоторые этапы процесса стриминга пикселей можно улучшить с помощью интеграции МО (рис. 2). Это поможет улучшить качество изображения или повысить производительность системы.

В данной работе будет рассмотрена интеграция МО в процесс сжатия видео. Сжатие видео является очень важным аспектом в процессе потоковой передачи пикселей, так как оно определяет как размер отправляемого видеопотока, так и скорость его создания, что затрагивает практически все этапы жизненного цикла системы.

Оптимизация графики	<ul style="list-style-type: none"> Анализ данных о производительности системы и выявления эффективных графических настроек для желаемого качества изображения при заданной частоте кадров
Сглаживание и фильтрация	<ul style="list-style-type: none"> Разработка эффективных алгоритмов сглаживания и фильтрации с целью учета характеристик каждого кадра и улучшения его качества без потери детализации
Распознавание объектов	<ul style="list-style-type: none"> Распознавание объектов на экране и автоматической настройки графических характеристик в зависимости от значимости объекта в кадре
Анализ данных о производительности	<ul style="list-style-type: none"> Анализ больших объемов данных о производительности системы с выявлением закономерностей для оптимизации потоковой передачи
Прогнозирование нагрузки	<ul style="list-style-type: none"> Прогнозирование нагрузки системы в реальном времени и адаптация графических настроек для поддержания стабильной частоты кадров
Процесс сжатия видео	<ul style="list-style-type: none"> Уменьшение размера видеофайла без существенной потери качества изображения, например, при потоковой передаче видеоплеера или других приложений, требующих высокой частоты кадров и большого объема данных для передачи

Рис. 2. Этапы процесса стриминга пикселей с интеграцией МО

Изначально технология потоковой передачи пикселей создавалась для движка Unreal Engine с использованием формата сжатия H.265, который используется и по сей день. Однако, быстро развивающиеся технологии машинного обучения показали отличные результаты сжатия выходного видео при сохранении исходного качества, что позволило конкурировать с традиционными технологиями сжатия видео. Для того чтобы выяснить, какая технология сжатия видео лучше всего подходит под требования потоковой передачи пикселей, необходимо проанализировать существующие технологии сжатия видео.

3. Анализ технологии сжатия видео

Видео — это последовательный набор изображений, который создан путем проецирования реальной сцены на двухмерную плоскость с помощью видеодатчика или создания последовательности искусственно созданных изображений, например анимации. Каждое изображение (кадр или изображение) отображается с определенной частотой, которая обычно измеряется в кадрах в секунду (кадров в секунду) или герцах (Гц). Частота кадров может варьироваться от 24 до 30 кадров в секунду, при этом 24 кадра в секунду являются наиболее часто используемой частотой кадров в киноиндустрии [3].

Системы кодирования видео состоят из двух основных компонентов: кодера и декодера. Кодер отвечает за создание сжатого битового потока из исходного видеофайла. Отношение между скоростью передачи сжатого битового потока и исходного видеофайла называется коэффициентом

сжатия. Задача декодера состоит в том, чтобы принять сжатый поток битов в качестве входных данных и создать выходной видеофайл, пригодный для отображения. Поскольку поток битов, создаваемый кодером, должен однозначно интерпретироваться декодером, который часто располагается на отдельном устройстве, точная совместимость между двумя системами имеет решающее значение. Например, в сервисе потокового видео кодирование происходит на серверах данных, а декодирование на принимающем устройстве, которым может быть телевизор, ПК или сотовый телефон [4].

Для эффективного сжатия видео требуются видеокодеки, обеспечивающие хранение и передачу видеоданных с минимальными потерями. Эти кодеки используют сложные алгоритмы для кодирования и сжатия видеоданных до более удобного размера без ущерба для качества видео. Можно выполнить сжатие практически без потери качества, либо устранить избыточность видео- или графических данных посредством сжатия с минимальным заметным ухудшением качества. Сжатие с потерями является необратимым, поскольку кодеки обращают процесс распаковки для аппроксимации входных данных [5].

4. Сравнение характеристик кодеков

Разберем некоторые популярные кодеки для кодирования видео. Анализ публикаций [6-8] позволил выявить их наиболее значимые для рассматриваемой задачи характеристики и функции (табл. 1).

Таблица 1. – Сравнение традиционных технологий сжатия видео

Характеристика	H.265	VP9	AV1
Разработчик	JCT-VC	Google	Alliance for Open Media
Распространенность	Поддерживается большинством устройств	Используется в некоторых крупных компаниях	Используется в некоторых проектах сообщества
Эффективность сжатия	Высокая	Средняя	Очень высокая
Потребление ресурсов	Среднее	Низкое	Высокое
Качество видео	Высокое	Хорошее	Отличное
Применение	Видео высокой чёткости, стриминг, архивирование	Стриминг, видеоконференции, облачные сервисы	Видео высокой чёткости, стриминг, архивирование, облачные сервисы
Разрешение	8К	8К	8К
Поддержка HDR	Есть	Есть	Есть
Особенности алгоритма сжатия	Многоуровневое предсказание Энтропийное кодирование Адаптивное квантование Использование больших блоков Компенсация движения	Энтропийное кодирование Компенсация движения Адаптивное квантование Многоуровневое предсказание	Контекстно-адаптивное кодирование Энтропийно-иерархическое кодирование Адаптивное квантование Многоуровневое предсказание Векторное предсказание движения Оптимизация для работы с различными типами видеоконтента

Среди существующих на данный момент кодеков можно выделить три наиболее популярных: H.265, VP9 и AV1.

Первый из них является наиболее распространенным. На сегодняшний день он поддерживается практически каждым девайсом, что есть на рынке устройств. Рассматривая H.265, нужно также вспомнить о его предшественнике – H.264. По сравнению с ним, H.265 сделал внушительный рывок в эффективности сжатия (экономия битрейта примерно в полтора раза).

Разработка Google также показывает хорошие результаты в эффективности сжатия при соответствующем качестве видео, хоть и отстает от H.265 (экономия битрейта примерно на треть больше, чем у H.264). Однако, VP9 обладает достаточно низким потреблением ресурсов, что также позволяет найти применение во множестве проектов (в основном связанных с онлайн-трансляциями).

Наименее распространенный кодек – AV1. Он достигает наибольшей эффективности сжатия среди представленных кодеков (экономия битрейта на треть больше, чем у H.265). Данный кодек является еще довольно молодым, однако уже может использоваться на множестве устройств благодаря отличной масштабируемости. Это делает его многообещающим вариантом для будущих видеоприложений.

Теперь рассмотрим применение технологий нейронных сетей в сжатии видео. Проведенный анализ [10-12] позволил выявить технологии нейронных сетей, которые играют важную роль в разработке кодеков сжатия видео на данный момент (табл. 2).

В статье рассмотрены три технологии нейронных сетей, применение которых значительно улучшает технологию сжатия видео – CNN, DAL и RNN.

Таблица 2. – Сравнение технологий сжатия видео с использованием машинного обучения

Характеристика	CNN	DAE	RNN
Тип сети	Сверточная нейронная сеть	Автоэнкодер	Рекуррентная нейронная сеть
Основная задача	Распознавание образов и классификация изображений	Обучение без учителя, сжатие данных и шумоподавление	Обработка последовательностей данных
Архитектура	Сверточные слои, которые обрабатывают изображения с помощью применения фильтров к небольшим участкам входного изображения	Энкодер и декодер, которые сжимают и восстанавливают данные	Рекуррентные блоки, которые позволяют обрабатывать последовательности данных
Преимущества	Высокая эффективность в задачах компьютерного зрения, инвариантность к сдвигу	Хорошая способность к обучению без учителя, эффективное сжатие данных	Хорошо подходит для обработки последовательностей данных (текст, временные ряды)
Недостатки	Требует большого количества данных для обучения	Может быть менее эффективным в задачах классификации	Может страдать от проблемы затухания градиента
Распространенность	Широко распространена	Используется в большом количестве проектов	Используется в ограниченном количестве проектов
Эффективность сжатия	Высокая	Средняя	Высокая
Потребление ресурсов	Большое	Большое	Среднее
Качество видео	Высокое	Отличное	Хорошее

Наиболее распространена в среде сжатия видео – CNN так как она предназначена для распознавания и обучения на основе визуальных шаблонов, таких как изображения, видео и т. д. Оказывает значительное влияние на эффективность сжатия, а также позволяет достигнуть высокой скорости кодирования/декодирования.

Одним из конкурентов CNN является автокодировщик DAE. DAE позволяет получить отличное качество выходного видеоматериала. Однако такая эффективность требует значительного количества обучающих данных и вычислительных ресурсов.

Еще одним кодеком на основе нейронных сетей является RNN. RNN позволяет добиться наибольшей эффективности сжатия среди представленных в статье технологий. При этом затраченные ресурсы зависят от входных параметров.

В своей статье [13] автор представила сравнение описанных выше кодеков на основе двух характеристик: пикового отношения сигнал/шум (PSNR) и индекса структурного сходства (SSIM). Результаты показали, что все кодеки имели одинаковые значения PSNR и SSIM, но VP9 показал немного меньшую производительность при том же качестве видео. Также было обнаружено, что модели CNN и RNN превосходят традиционные кодеки по высокой эффективности сжатия и качеству изображения. Однако модель RNN показала лучшие результаты по степени сжатия и качеству изображения по сравнению с моделью CNN. Метод DAE также продемонстрировал высокую эффективность сжатия и отличное качество изображения, но требует больше обучающих данных и вычислительных ресурсов.

Таким образом, все методы показали хорошие результаты с точки зрения эффективности сжатия и качества изображения. Однако метод RNN

превзошел методы CNN и DAE, а также традиционные методы сжатия.

5. Заключение

Машинное обучение в настоящее время имеет широкий спектр областей применения. Одной из таких областей является стриминг пикселей, который, в свою очередь, может быть использован в медицине, например, при проведении медицинской операции (требуется четкая и оперативная картинка из устройства внутри пациента или необходимо выделять какие-то фрагменты в организме во время работы). Применение машинного обучения на ряде этапов данного процесса может позволить значительно повысить эффективность. Например, учитывать характеристики каждого кадра и повышать качество изображения без потери детализации, на основе анализа больших объемов данных о производительности системы выявлять закономерности для оптимизации потоковой передачи и пр.

Для рассматриваемого в статье этапа сжатия видео могут быть использованы традиционные технологии, а также технологии с применением машинного обучения, которые позволяют уменьшить размер видеофайла без существенной потери качества изображения.

Традиционные технологии связаны с применением кодеков, сравнение которых позволяет выявить достоинства и недостатки каждого (H.265, VP9, AV1) и выбрать их для стриминга пикселей в зависимости от ситуации.

- AV1 предлагает значительно более высокую эффективность сжатия по сравнению с остальными кодеками.
- VP9 требует меньшей вычислительной мощности для кодирования и декодирования по сравнению с другими кодеками.
- H.265 поддерживается большинством устройств.

Наряду с традиционными технологиями приведенный в статье анализ технологий с методами машинного обучения позволяет сделать вывод о том, что методы CNN и RNN особенно эффективны для хранения пространственной и временной информации, DAE лучше всего подходит для использования в технологии потоковой передачи пикселей. Эффективность сжатия этой технологии немного хуже, чем у RNN, но качество выходного изображения кажется лучшим.

Благодарности

Исследование поддержано грантом РФФ № 22-19-00471.

Список используемых источников

1. Pixel Streaming 2 Overview; URL: <https://dev.epicgames.com/documentation/en->

[us/unreal-engine/pixel-streaming-2-overview-in-unreal-engine](https://dev.epicgames.com/documentation/en-us/unreal-engine/pixel-streaming-2-overview-in-unreal-engine) (дата обращения: 28.10.2024).

2. Mike Leins. What is Pixel Streaming? Beginner's Complete Guide to Use Cases & Setup Guide; URL: <https://vagon.io/blog/what-is-pixel-streaming> (дата обращения: 28.10.2024).
3. Brown, A.J., Baburin, A.S. System and Method for Digital Video Management—USA, U.S. patent, 2010. – 32 с.
4. Ameres, E.L.; Bankoski, J.; Grange, A.W.; Murphy, T.; Wilkins, P.G.; Xu, Y. Video Compression and Encoding Method Management—USA, U.S. patent, 2009. – 5 с.
5. Wiseman, Y. Video Compression Prototype for Autonomous Vehicles // Smart Cities 2024, Vol. 7, P. 758–771.
6. Grois, D.; Nguyen, T.; Marpe, D. Coding efficiency comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC encoders // Proc. of the 2016 Picture Coding Symposium (PCS). Germany, Nuremberg, 2016. P. 1–5.
7. Mukherjee, D.; Bankoski, J.; Grange, A.; Han, J.; Koleszar, J.; Wilkins, P.; Xu, Y.; Bultje, R. The latest open-source video codec VP9—An overview and preliminary results // Proc. of the 2013 Picture Coding Symposium (PCS). USA, CA, San Jose, 2013. P. 390–393.
8. Yasin, H.M.; Abdulazeez, A.M. Image Compression Based on Deep Learning: A Review // Asian J. Res. Comput. Sci. 2021. Vol. 8. P. 62–76.
9. Nandi, U. Fractal image compression with adaptive quadtree partitioning and non-linear affine map // Multimed. Tools Appl. 2020. Vol. 79. P. 26345–26368.
10. Albahar, M. A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities // Agriculture. 2023. Vol. 13. P. 540.
11. Hu, Y.; Yang, W.; Xia, S.; Cheng, W.H.; Liu, J. Enhanced intra prediction with recurrent neural network in video coding // Proc. of the 2018 Data Compression Conference. USA, UT, Snowbird, 2018. P. 413.
12. Toderici, G.; O'Malley, S.M.; Hwang, S.J.; Vincent, D.; Minnen, D.; Baluja, S.; Covell, M.; Sukthankar, R. Variable Rate Image Compression with Recurrent Neural Networks // Proc. Of ICLR 2016, USA, CA, Mountain View, 2015.
13. Mochurad L. A Comparison of Machine Learning-Based and Conventional Technologies for Video Compression // Technologies, 2024.