DOI: 10.55648/1998-6920-2024-18-3-115-139 УДК 004.932.1:004.032.26

Исследование границ интенсивности видеопотока при FPV-управлении БПЛА в режиме предсказания кадров. Часть I: модели и методы *

А. А. Березкин, А. А. Ченский, Р. В. Киричек

Санкт-Петербургский гос. унив. телеком. им. проф. М. А. Бонч-Бруевича (СПбГУТ)

Аннотация: В настоящее время беспилотные летательные аппараты нашли широкое применение в различных сферах народного хозяйства. FPV-управление (управление от первого лица) относится к числу способов управления беспилотными летательными аппаратами, при котором видеопоток от беспилотного воздушного судна передается на станцию внешнего пилота в реальном масштабе времени. Из-за сбоев в работе сети связи пакеты с данными видеопотока могут потеряться или задержаться и быть доставленными с опозданием. Одним из способов, позволяющих компенсировать падение FPS в случае потери или задержки кадров на станции внешнего пилота, является прогнозирование промежуточных кадров видеопотока. В настоящей работе представлена схема предсказания промежуточных кадров видеопотока и приведены результаты экспериментов по установлению реально возможного увеличения FPS в нейросетевом кодеке, в котором для сжатия видео используются различные автокодировщики типа VQ-f16, а также алгоритм сжатия латентного пространства признаков без потерь DEFLATE, при этом для предсказания используется динамическая многомасштабная нейронная сеть воксельного потока DMVFN. Разработана регрессионная модель для прогнозирования времени предсказания. Формализована задача анализа FPS при различных конфигурациях нейросетевого декодера на стороне внешнего пилота.

Ключевые слова: FPS, вариационный автокодировщик, нейронный декодер, DMVFN, нейронные сети, предсказание видеопотока.

Для цитирования: Березкин А. А., Ченский А. А., Киричек Р. В. Исследование границ интенсивности видеопотока при FPV-управлении БПЛА в режиме предсказании кадров. Часть I: модели и методы // Вестник СибГУТИ. 2024. Т. 18, № 3. С. 115–139. https://doi.org/10.55648/1998-6920-2024-18-3-115-139.

Статья поступила в редакцию 5.06.2024;
принята к публикации 20.06.2024.

1. Введение

В настоящее время беспилотные летательные аппараты (БПЛА) широко используются как в гражданской, так и в военной сферах [1]. При этом в случаях, когда полёт в автоматическом режиме невозможен или затруднителен, используется дистанционное управление БПЛА. Одной из разновидностей дистанционного управления БПЛА является управление от

* Научная статья подготовлена в рамках прикладных научных исследований СПбГУТ, регистрационный номер 1023031600087-9-2.2.4;2.2.5;2.2.6;1.2.1;2.2.3 в ЕГИСУ НИОКТР.

первого лица (first person view – FPV). FPV-управление предполагает передачу видеопотока и данных телеметрии от БПЛА на станцию внешнего пилота (СВП) по линии связи с землей (down link), а от СВП на БПЛА по линии связи с бортом (up link) – команд управления. Преимуществом такого способа управления является возможность управления БПЛА вне поля прямой видимости без вспомогательных наблюдательных средств, не установленных на БПЛА.

На текущий момент для осуществления связи между БПЛА и СВП, кроме радиолиний прямой видимости, широко используются мобильные сети связи. Тем не менее значительная часть территории Российской Федерации не покрыта мобильной связью. В перспективе обеспечить связь для управления БПЛА на всей территории Российской Федерации согласно Стратегии развития отрасли связи Российской Федерации на период до 2035 года [2] смогут гибридные орбитально-наземные сети связи (ГОНСС).

На конец 2023 года спутниковая группировка Российской Федерации насчитывала около 230 космических аппарата (КА) [3]. К 2036 году их количество планируется увеличить в 15 раз. При этом количество официально зарегистрированных на территории Российской Федерации беспилотных воздушных судов (БВС) на конец 2023 года уже исчислялось десятками тысяч (более 93 тысяч штук) и выросло с прошлого года на более чем 30 % [4]. Таким образом, существует значительная диспропорция между числом КА и БВС.

Можно предположить, что в перспективе 2035–2036 годов наличие задержек и потерь пакетов при FPV-управлении БПЛА прежде всего на отдалённых территориях Российской Федерации сохранится, т.к. только некоторая часть из 3435 КА, которые планируется вывести на орбиту к 2036 году, будет использоваться в качестве активных сетевых устройств ГОНСС, имеющей ограниченную пропускную способность.

Задержки и потери пакетов приводят к снижению частоты кадров в секунду (frame per second – FPS) видеопотока на стороне внешнего пилота (ВП), что может привести к ошибкам управления БПЛА от первого лица. В [5] предложен механизм адаптивного контроля интенсивности видеопотока при передаче FPV-трафика БВС на основе контрольных карт Шухарта и моделей временных рядов (рис. 1).

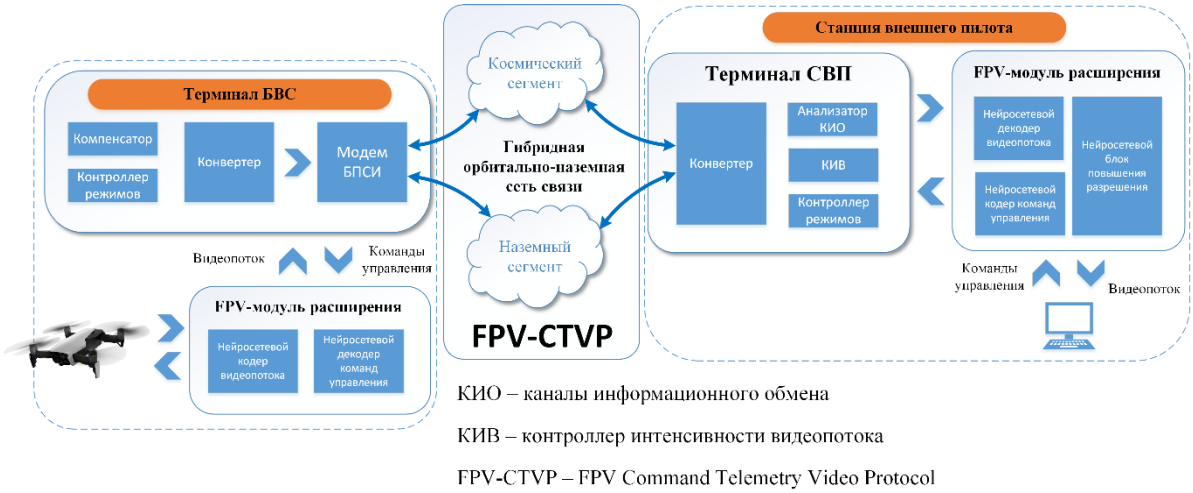

Рис. 1. Схема информационного обмена при FPV-управлении в ГОНСС

Система FPV-управления, представленная на рис.1, предназначена для предоставления услуги транспорта FPV-трафика с адаптивным контролем FPS видеопотока через различные сегменты инфраструктуры пакетных сетей связи и обеспечивает сокращение задержки передачи видеопотока и агрегацию команд управления с БВС на СВП.

На стороне СВП плавность отображения видеопотока оператору обеспечивается анализатором каналов информационного обмена (КИО), который контролирует реальный FPS на стороне оператора и отправляет команды в рамках специализированного протокола прикладного уровня FPV-CTVP (FPV-Command Telemetry Video Protocol), обеспечивающего пе-

редачу основных и служебных информационных потоков. В случае ухудшения качества канала (уменьшения ширины полосы пропускания) на стороне БВС обеспечивается выкалывание кадров видеопотока, которые не могут быть переданы вовремя, а на стороне СВП предсказание недостающих кадров модулем контроля интенсивности видеопотока (КИВ).

Ключевым элементом данной схемы является модуль КИВ, концепция и методы реализации которого рассмотрены в [6, 7], где предлагается использование нейросетевых моделей для предсказания последующих кадров видеопотока. Тем не менее проблема использования данных моделей в рамках модуля КИВ не является достаточно исследованной. Остаётся ряд вопросов, подлежащих изучению:

1. Время работы декодера t. Заданный уровень FPS определяется не только периодом поступления кадров на СВП, но и временем работы декодера. В случае использования для декодирования алгоритмов высокой сложности или нейросетевых моделей время исполнения становится слишком велико. В то же время в [6, 7] практически не рассматривается параметр времени выполнения моделей, а внимание уделяется метрикам качества восстанавливаемых кадров и уменьшению объёма видеопотока в целом.

2. Способность предсказательной модели обеспечивать увеличение FPS при некотором минимальном периоде поступления кадров $T_{\min}$. В случае ограниченности времени выполнения моделей всегда найдётся достаточно большой период $T_{\min}$, при котором модель способна обеспечить увеличение FPS.

3. Обоснованность использования различных моделей предсказания кадров. Целесообразной для использования признаётся такая модель, при которой минимальный период поступления кадров $T_{\min}$ меньше времени потери актуальности кадра $t_{\max}$. Данный параметр зависит от интенсивности смены окружения и различается для разных задач, решаемых БПЛА в рамках FPV-управления.

4. Использование механизмов интерполяции в потоке предсказания кадров. Это зависит от времени выполнения интерполяции и предсказания и от конкретных алгоритмов либо моделей машинного обучения, используемых для решения данных задач.

5. Предел возможного увеличения параметра кадров в секунду $\mathrm{FPS}_{\max}$. Так как выполнение нейросетевых моделей требует некоторого времени, FPS ограничено. Следовательно, возникает вопрос, насколько возможно максимально увеличить FPS с помощью некоторой предикторной модели.

6. Технические аспекты реализации модуля КИВ и его работы в рамках системы информационного обмена между БПЛА и СВП.

7. Определение параметра $t_{\max}$ для различных задач, возникающих в рамках FPV-управления БПЛА.

8. Выбор наилучшей предсказательной модели с точки зрения $\mathrm{FPS}_{\max}$. Определение компромиссного варианта модели с точки зрения времени выполнения и качества восстанавливаемых кадров в соответствии с требованиями обеспечения качества услуги.

В настоящей работе рассматриваются вопросы 1 – 6. Целью данной работы является исследование возможности практического применения предложенной концепции контроля интенсивности видеопотока [6, 7] в рамках нейросетевого декодера [8].

2. Экспериментальная установка

Для реализации экспериментальной установки используется прототип нейросетевого декодера (рис. 2).

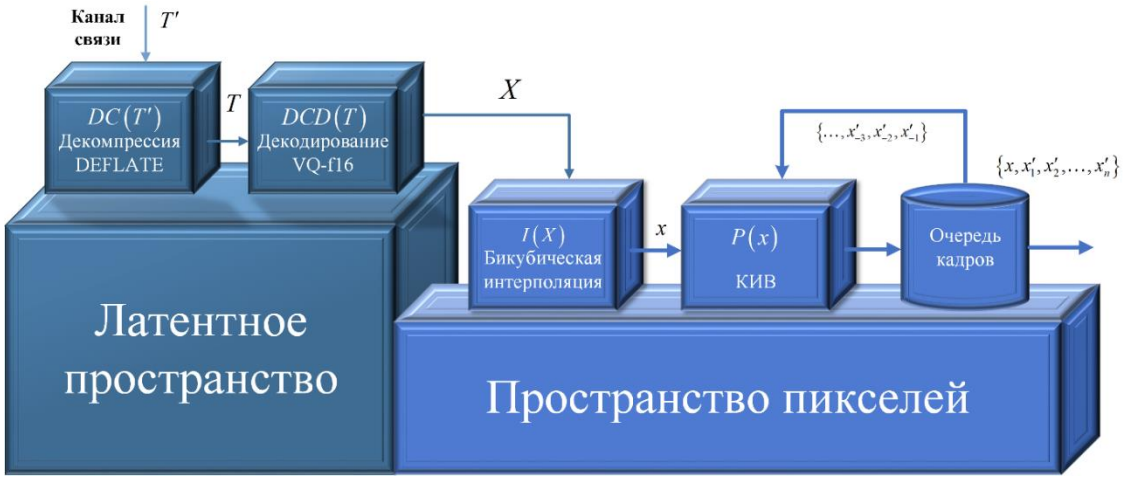

Рис. 2. Прототип нейросетевого декодера

Со стороны кодировщика, расположенного на борту БПЛА, передаются сжатые кадры видеопотока в латентном представлении T' в формате *fp16*, которые через канал связи между БПЛА и СВП поступают на блок декомпрессии *DC* нейросетевого декодера на стороне СВП. После декомпрессии кадры видеопотока в латентном пространстве T поступают на блок декодирования *VQ-f16* [9] *DCD*, на выходе которого получается пиксельное представление кадра X. Далее X поступает на блок бикубической интерполяции I для увеличения до некоторого разрешения x. На блок КИВ поступает как интерполированный кадр x, так и n ранее предсказанных кадров x_{-n}', …, x_{-3}', x_{-2}', x_{-1}'. На выход блока КИВ подаётся n+1 кадров: x и предсказанные x_1', x_2', x_3', …, x_n'. В качестве предсказательной модели блока КИВ используется нейросетевая модель DMVFN (Dynamic Multi-scale Voxel Flow Network) [10]. С выхода модуля КИВ кадры поступают в очередь и далее на оборудование ВП.

Модели нейросетевого декодера реализованы с помощью библиотеки PyTorch и выполняются на языке программирования Python версии 3.11. Для корректности все измерения времени выполнения отдельных блоков проводятся с использованием функций time_ns модуля стандартной библиотеки time и torch.cuda.syncronize(), что позволяет корректно измерять время выполнения каждого элемента блока, но может привести к небольшому увеличению общего времени работы ввиду асинхронности операций CUDA.

Входными параметрами нейросетевого декодера являются: разрешение поступающих кадров (a, x, b), число предсказываемых кадров n, результирующее разрешение (w, x, h).

Эксперименты в рамках настоящего исследования проводятся на серверной платформе Gigabyte G481-HA1. Модели нейронных сетей выполняются на видеокарте Nvidia Tesla A100 с 80 Гб видеопамяти.

3. Методика предсказания кадров видеопотока

Пусть некоторые кадры видеопотока x_i и x_{i+d} поступили в моменты времени i и (i+d). Методика предсказания кадров видеопотока заключается в нахождении кадров, которые бы поступили из канала связи в моменты времени между i и i+d (рис. 3, d = 4). Тем не менее такое определение не учитывает временные затраты на декомпрессию, декодирование и интерполяцию каждого кадра. Поэтому определим методику предсказания кадров видеопотока в контексте нейросетевого кодера другим образом.

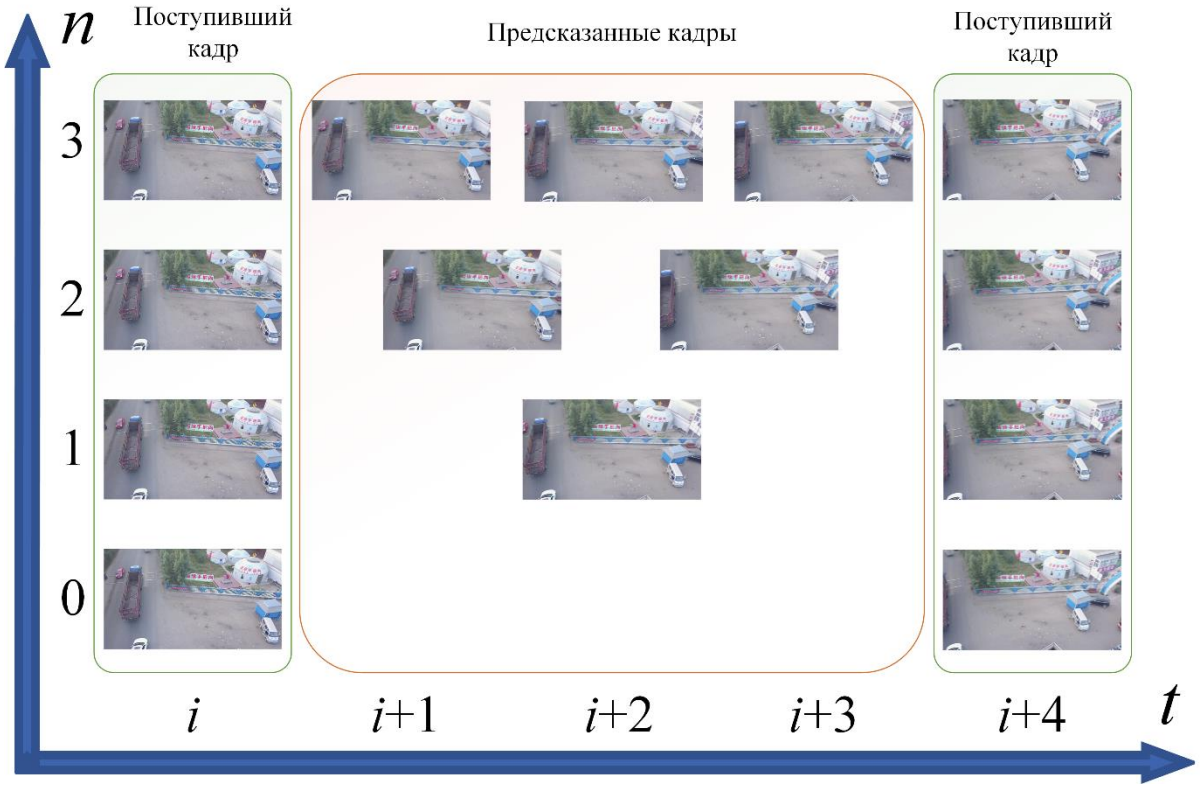

Рис. 3. Методика предсказания кадров видеопотока

Пусть z_i – суммарное время чтения из буфера, преобразования, декомпрессии, декодирования и интерполяции поступающего на нейросетевой декодер кадра, y_i – интервал времени между поступлением i и i+1 кадра, d_i – интервал времени между поступлением i-го кадра и некоторого следующего отображаемого кадра i' внешнему пилоту, t_i – время поступления i-го кадра. Поступающие кадры обрабатываются последовательно: $T_i', T_{i+1}', T_{i+2}', \ldots, T_{i+d}'$. Последний сжатый латентный кадр T_i', поступивший из канала связи, буферизируется и может быть прочитан сразу после завершения декодером работы над предыдущим. При каждом поступлении нового кадра T_{i+1}' буфер перезаписывается. Определим некоторый кадр в очереди кадров x_j* как потерявший актуальность в тот момент, когда в очередь кадров поступил новый кадр x_j. Соответственно, в один момент времени актуальный кадр x_j единственен. Далее определим J_t как множество индексов всех кадров, которые до некоторого времени t включительно были актуальными. Факт того, что все актуальные кадры поступают в очередь кадров, означает, что все они читались из буфера и обрабатывались декодером. Тогда актуальный кадр x_j в некоторый момент времени можно определить по формуле:

$$x_j : j : \left(\sum_{i=0}^{j} y_i = \left(\sum_{k \in J_t} z_k \right) - z_j \right), \tag{1}$$

а d_j – по формуле:

$$d_j = \begin{cases} 0, j \notin J_{tj} \\ \max\left(z_0, y_0\right), j \in J_{tj}, t < t_1 \\ z_j, j \in J_{tj}, t > t_1 \end{cases}.$$

Выражение (1) можно переписать в виде, где p – достаточно малая случайная величина:

$$x_j : j = \max(j) : \left(\sum_{i=0}^{j} y_i = \sum_{k \notin J_{tj+zj-p}} z_k \right), t_j \le t, p > 0, p \to 0,$$

$$x_j : j = \max(j) : \left(\sum_{i=0}^{j} y_i = t_j \right), t_j \le t .$$

Таким образом, методика предсказания кадров видеопотока заключается в предсказании n кадров на временном интервале (t_i+z_i; t_i+d_i+z_i'), таких что

$$t_{i:k} = t_i + z_i + \frac{d_i}{n} \times k,\ k = 1, 2, 3, ..., n ,$$

где $t_{i:k}$ – момент времени предсказания k-го кадра после i-го пришедшего с кодера кадра.

Процесс функционирования системы последовательной обработки кадров представлен на рис. 4.

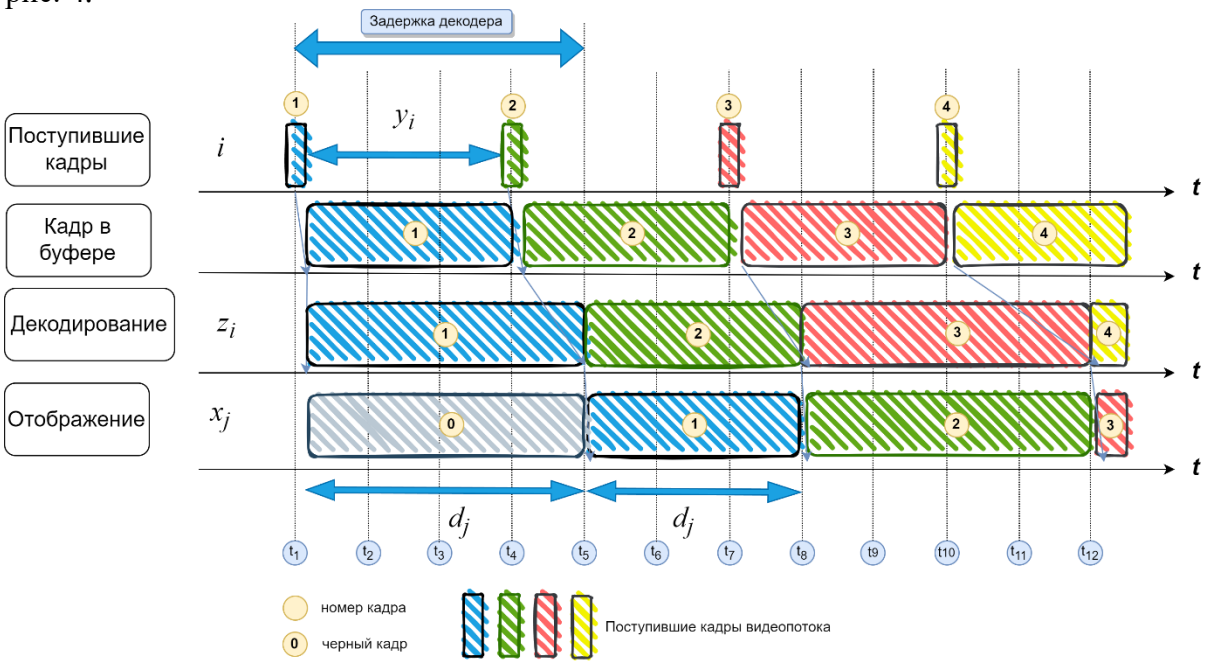

Рис. 4. Процесс последовательного отображения кадров видеопотока

Данная модель обработки кадров на стороне декодера справедлива в случае последовательного выполнения этапов декодирования на стороне СВП и не учитывает наличие предсказательной модели. В случае, когда отбор кадров для отображения на стороне ВП выполняется параллельно с заданным FPS, актуальные кадры в моменты времени j и d_i получаются в соответствии с выражениями:

$$x_j : j = \max(j) : \begin{cases} \left(\sum_{i=0}^{j} y_i = \left(\sum_{k \in J_t} z_k \right) - z_j \right) \\ t' = t - d_i \times \left\lfloor \frac{t}{d_i} \right\rfloor \end{cases},$$

$$x_j : j = \min(j) : \begin{cases} \max(j) : \left(\sum_{i=0}^{j} y_i = t_j \right), t_j \le t' \\ t' = t - d_i \times \left\lfloor \frac{t}{d_i} \right\rfloor \end{cases},$$

$$d_i = \text{const} = \frac{1}{\text{FPS}} (\text{c}) .$$

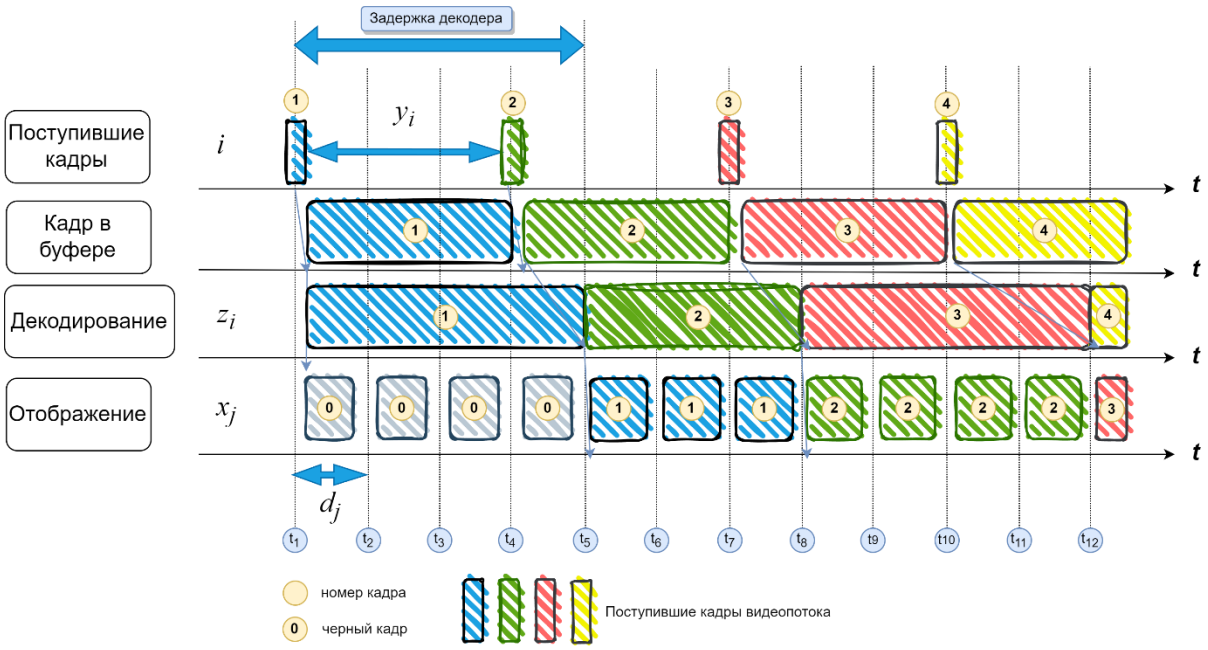

Рис. 5. Процесс параллельного отображения кадров видеопотока

Следует отметить, что свести задачу предсказания кадров FPV-видеопотока к задаче интерполяции промежуточных кадров только из кадров, поступивших из канала связи, невозможно [7]. В момент i на СВП имеется только кадр x_i, а также предыдущие промежуточные кадры x_{i-1}', x_{i-2}', x_{i-3}' и далее. Для интерполяции кадров x_{i+1}', x_{i+2}', x_{i+3}' требуется кадр x_{i+d}, но в момент времени i, когда требуется выполнить интерполяцию, на СВП его нет. В этой связи необходимо использовать нейросетевые алгоритмы, позволяющие предсказывать необходимые кадры на основе предыдущих.

В данной работе рассматривается частный случай предсказания n кадров на основе $n+1$ предыдущих кадров, включающих в том числе и один реальный. Это обусловлено высокой динамикой сцены при FPV-управлении БПЛА.

Однако после поступления в очередь первого кадра x_0 в ней отсутствуют необходимые n предыдущих предсказанных кадров. Соответственно, выполнение предсказания по предложенной схеме в начальный момент времени невозможно. Требуется некоторое время на запуск процесса предсказания, который может быть выполнен несколькими способами:

1. Предсказание с первых кадров. Получение первых n кадров из канала связи и запуск процесса предсказания из кадров x_0, x_1, x_2… x_n. Минусом данного подхода являются различные временные промежутки между этими кадрами и кадрами для предсказания, которые будут сглаживаться по мере продолжения процесса предсказания.

2. Первичная интерполяция. Получение первых двух кадров x_0 и x_1, интерполяция n–1 кадров между ними и запуск процесса предсказания. Минусом данного подхода является пропуск многих деталей при достаточно больших n и y_0.

3. Пошаговое предсказание. Пусть h_0, h_1, h_2, …, h_k – неубывающая последовательность шагов предсказания, такая что $h_k = n$. Сперва выполняется получение h_0+1 первых кадров и предсказывается h_0 промежуточных кадров. Далее после поступления дополнительных кадров из h_1+1 кадров предсказываются (из всего объема кадров) h_1 кадров и так далее до достижения n предсказываемых кадров. Метод является модификацией предсказания с первых кадров с некоторым сглаживанием различных временных промежутков. Для одной из наиболее простых модификаций предложенного алгоритма необходимо n шагов для начала процесса предсказания: $h_0 = 1$, $h_{i+1} = h_i + 1$.

4. Распределённая интерполяция. Пусть для интерполяции используется k кадров. Тогда между x_i и x_{i+1} кадрами интерполируется n_i кадров таким образом, что числа интерполируе-

мых кадров удовлетворяют выражению $\sum_{i=0}^{k}(n_i-1)-1=n$, если первый кадр входит в исходные для следующего предсказания, и выражению $\sum_{i=0}^{k}(n_i-1)=n$, если не входит. Данный метод является обобщённым случаем первичной интерполяции и позволяет уменьшить пропуск многих деталей при достаточно высоких n и y_i. Из него можно вывести ряд методик. Например, первый кадр в предсказании не участвует, а все n_i взяты равными, тогда k определяется по формуле

$$k=\frac{n}{n_i-1}$$

В настоящей работе используется наиболее простой первый способ запуска процесса предсказания. Способ запуска процесса важен только с точки зрения качества восстанавливаемых кадров, а не времени выполнения.

4. Методика проведения эксперимента

В ходе экспериментов определяются параметры нейросетевого декодера при поступлении 1000 кадров из авторского набора данных и измеряются целевые метрики. Поступление следующего кадра осуществляется сразу после завершения работы всех блоков по обработке предыдущего кадра.

Рис. 6. Пример кадров из авторского набора данных

Параметр разрешения исходных кадров $a \times b$ (a – ширина кадра в пикселях, b – высота кадра в пикселях) неизменен и равен 512×512 [11].

Число предсказываемых кадров n варьируется от 0 до 5. Согласно исследованию [6] уже на пятом предсказываемом кадре начинают проявляться видимые дефекты, поэтому предсказывать большее число кадров нецелесообразно из-за значительного ухудшения качества.

Выходное разрешение $w \times h$ принимает следующие значения: 512×512, 1280×720 (HD) и 2560×1440 (QHD) (размерность FullHD 1920×1080 не поддерживается моделью DMVFN). При разрешении HD число пикселей приблизительно в 3.52 раза больше, чем в исходном разрешении, а при разрешении QHD – в 14 раз. Пикселей в QHD больше, чем в HD, в 4 раза. Данных показателей достаточно для проверки линейности времени предсказания кадров моделью. Итого проводится 18 экспериментов.

Целевые метрики отражают время выполнения отдельных блоков нейросетевого декодера: время декомпрессии t_{DC}, время декодирования t_{DCD}, время интерполяции t_I и время предсказания t_P. На их основе можно определить полное время работы декодера t_n:

$$t_n = t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + t_P \left(\text{мс}\right). \tag{2}$$

Время декомпрессии t_{DC} также включает время выполнения операций чтения буфера и перевода данных во внутренний формат системы. Параметр t_I включает в себя в том числе время перевода внутреннего формата данных из torch.tensor в numpy.ndarray. Метрики времени работы измеряются в миллисекундах (мс). Частоту кадров FPS можно определить в соответствии с выражением:

$$\mathrm{FPS} = \frac{\left(n+1\right)\times 1000}{t_n}\left(\text{к/с}\right). \tag{3}$$

Время работы очереди кадров не учитывается из-за его незначительности. Используемая доверительная вероятность равна 95 %.

Следует отметить, что операции на GPU с помощью CUDA выполняются асинхронно, что не позволяет определить время их выполнения, а значит, и связанные метрики точно. Попытка измерения времени выполнения декодера приводит к получению достаточно точного общего времени работы с одним кадром, но некорректного времени выполнения отдельных блоков. Тем не менее способ приближённого вычисления времени выполнения каждого из блоков заключается в использовании синхронизации после выполнения каждого блока (torch.cuda.syncronize). Это приводит к некоторому увеличению общего времени работы, но позволяет корректно измерять целевые метрики.

После получения первичных результатов проводится их анализ. Получаются зависимости времени предсказания t_P от числа предсказываемых кадров n и числа пикселей N_p всех предсказываемых кадров $N_p = \mathrm{width} \times \mathrm{height} \times n\left(\text{пикселей}\right)$, где N_p для исходного разрешения равно 262144×n, для HD – 921600×n, а QHD – 3686400×n.

На основе данных о зависимости числа пикселей N_p и времени предсказания t_P методом наименьших квадратов строится регрессионная модель. В случае линейной зависимости она определяется формулой:

$$\hat{t}_P = b_0 + b_1 \times N_p = b_1 \times N_p \left(\text{мс}\right),$$

где $b_0 = 0$ и b_1 – коэффициенты регрессии.

Вычисляется интервальная оценка [12] в соответствии с выражением:

$$\hat{t}_P \in \left[b_1 \times N_p \pm t_{a,n-2} \times \hat{S} \times \sqrt{\frac{1}{n} \times \left(1 + \frac{\left(\bar{N}_p - N_p\right)^2}{Var\left(N_p\right)} \right)} \right], \tag{4}$$

где $t_{a,n\text{-}2}$ – коэффициент Стьюдента с уровнем значимости a и n–2 степенями свободы, $\hat{S}$ – стандартная ошибка остатков (стандартная ошибка оценки или регрессии), которая вычисляется по формуле:

$$\hat{S} = \sqrt{\frac{\sum\left(t_P - \hat{t}_P\right)^2}{n-2}}$$ [12, с. 53].

Модель обобщается на другое оборудование путём оценки зависимости и ввода дополнительного коэффициента. Оценивается, является ли она гомоскедастичной и какова её область применения.

Полученная модель используется для нахождения такой конфигурации блоков нейросетевого декодера, при которой максимизируется FPS, а также границ данных конфигураций. Во-первых, рассчитывается минимальный период прихода кадров, при котором модель способна обеспечить увеличение FPS ($T_{\min}$). Во-вторых, оценивается максимальное FPS_{max} данного нейросетевого кодека и условия его достижения. В-третьих, проверяется правильность порядка блоков в нейросетевом декодере для максимизации величины FPS.

5. Эксперименты

Результаты экспериментов по заданным исходным данным и средние значения целевых метрик с доверительными интервалами представлены в табл. 1.

Таблица 1. Результаты экспериментов

n, шт.	w x h, пикс.	t_{DC}, мс	t_{DCD}, мс	t_I, мс	t_P, мс	t_n, мс	FPS, к/с
0	512×512	0,001 ±0,002	26.01 ±0.028	1.08 ±0.0169	0	27.1 ±0.036	36.92 ±0.03
0	1280×720	0	26	1.13 ±0.02	0	27.13 ±0.02	36.87 ±0.027
0	2560×1440	0	26	1.12 ±0.021	0	27.12 ±0.021	36.88 ±0.028
1	512×512	0	26.41 ±0.03	1.02 ±0.008	11.76 ±0.109	39.18 ±0.113	51.12 ±0.12
1	1280×720	0	26 ±0.004	1.89 ±0.049	29.92 ±0.16	57.82 ±0.18	34.67 ±0.106
1	2560×1440	0	26.53 ±0.04	2.02 ±0.098	175.88 ±0.45	204.44 ±0.453	9.79 ±0.022
2	512×512	0	26 ±0.003	1.05 ±0.014	22.05 ±0.089	49.11 ±0.091	61.14 ±0.114
2	1280×720	0	26	1.58 ±0.03	54.52 ±0.152	82.1 ±0.155	36.57 ±0.07
2	2560×1440	0	26.57 ±0.037	2.02 ±0.01	325.17 ±0.56	353.76 ±0.563	8.48 ±0.014
3	512×512	0	26.23 ±0.026	1.06 ±0.015	32.95 ±0.195	60.24 ±0.217	66.62 ±0.231
3	1280×720	0	26.36 ±0.03	1.76 ±0.228	76.28 ±0.228	104.39 ±0.237	38.37 ±0.088
3	2560×1440	0	27 ±0.004	2.18 ±0.024	433.36 ±2.738	462.55 ±2.748	8.73 ±0.052
4	512×512	0	26.06 ±0.015	1.11 ±0.02	42.84 ±0.179	70.02 ±0.191	71.54 ±0.186
4	1280×720	0	26.65 ±0.03	2.09 ±0.046	104.05 ±0.264	132.79 ±0.284	37.7 ±0.083
4	2560×1440	0	27 ±0.005	2.59 ±0.031	466.38 ±1.691	495.98 ±1.688	10.11 ±0.029

5	512×512	0	26.03 ±0.01	1.17 ±0.023	52.94 ±0.189	80.14 ±0.197	74.98 ±0.177
5	1280×720	0	26 ±0.004	1.83 ±0.026	126.76 ±0.284	154.59 ±0.285	38.84 ±0.072
5	2560×1440	0	26.01 ±0.006	2.04 ±0.014	767.23 ±1.129	795.279 ±1.129	7.55 ±0.012

Несмотря на наличие некоторых флуктуаций средних значений и половин длин доверительных интервалов во времени выполнения декодирования и интерполяции, которые при соответствующих экспериментах выполняются идентично, они приблизительно не превышают одной миллисекунды и не могут рассматриваться как существенные. Таким образом, для проведения экспериментов по определению времени выполнения на производительном оборудовании желательно использовать б*о*льший набор данных и проводить измерения уже в наносекундном диапазоне. Однако полученные результаты (табл. 1) достаточно точны для выполнения цели и задач настоящей работы.

Кроме этого, получены зависимости среднего времени предсказания t_P от числа предсказываемых кадров n (рис. 7) и корреляционная зависимость времени предсказания t_P числа пикселей N_p (рис. 8). Данные зависимости близки к линейным. В настоящей работе для предсказания n кадров используется n+1 кадров, следовательно, имеется прямая функциональная зависимость между n и N_p. Соответственно, линейность зависимости средней t_p от n при каждом отдельном разрешении обуславливается линейной корреляционной зависимостью t_p от N_p. На основе всех 18000 проведённых испытаний для этих двух величин был рассчитан коэффициент корреляции Пирсона: 0.9838, что близко к функциональной зависимости.

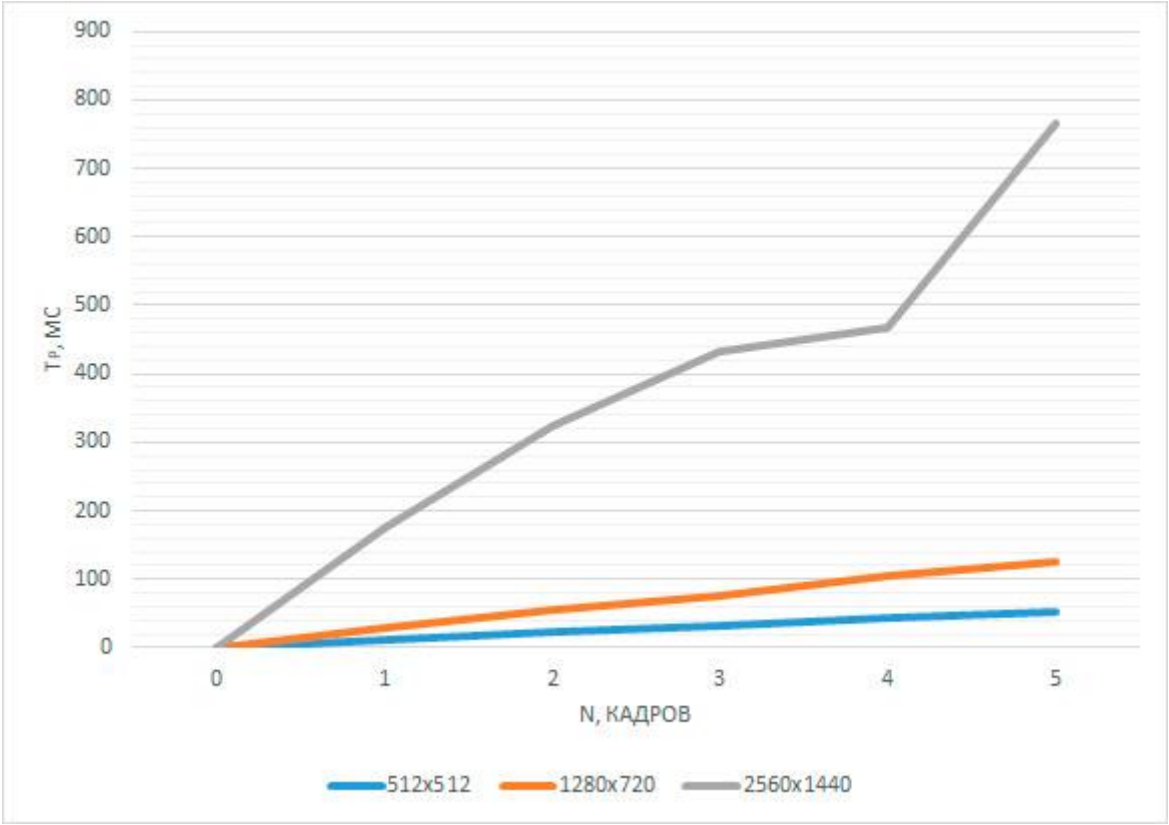

Рис. 7. Зависимость среднего времени предсказания от числа предсказанных кадров

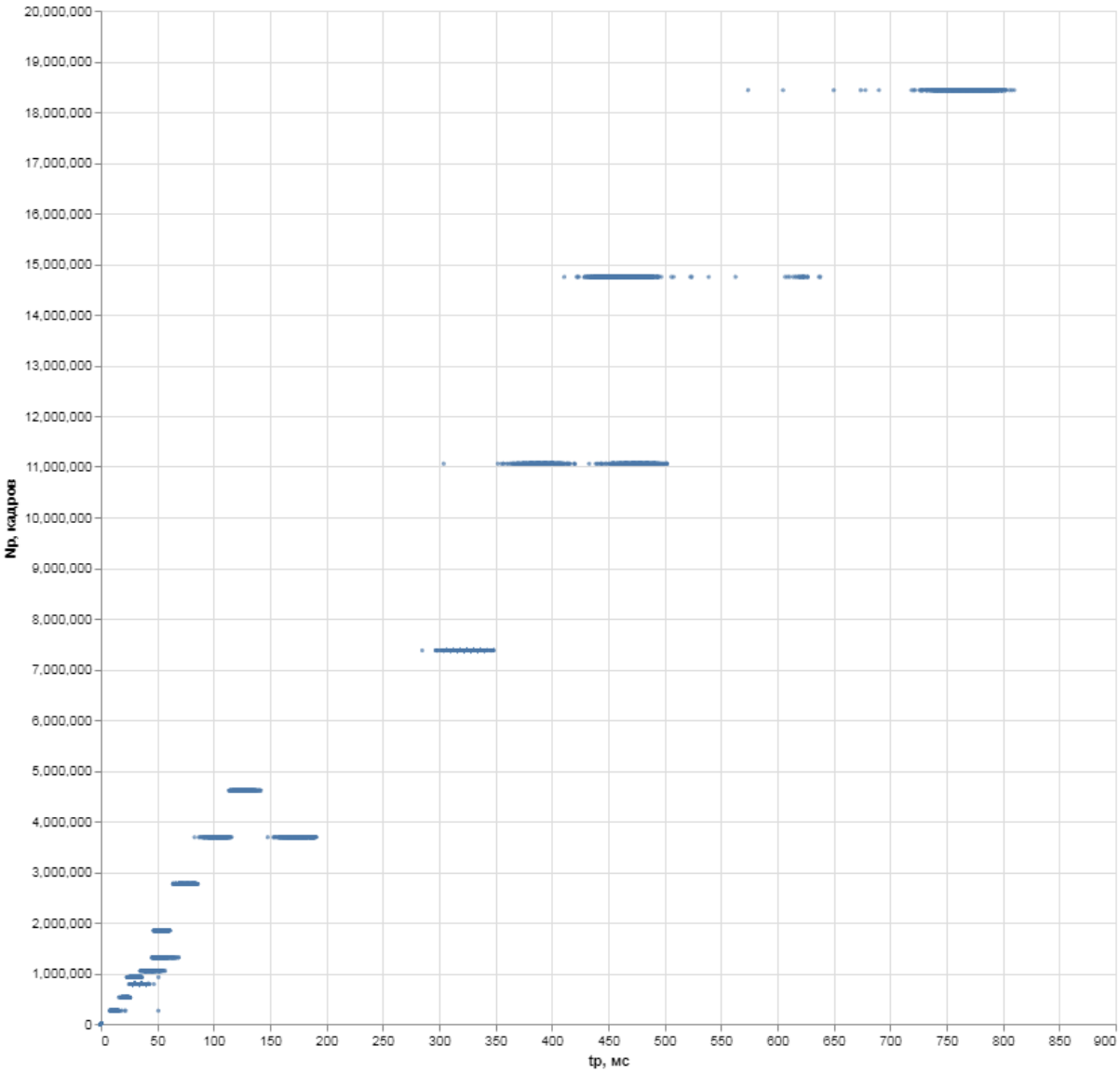

Рис. 8. Корреляционная зависимость времени предсказания от числа пикселей кадров

Отметим, что влияние отдельных компонент $N_{\text{полн}}$:

$$N_{\text{полн}} = N_p + N_{sp} = \text{height} \times \text{width} \times \left(n_s + n_d\right)\left(\text{пикселей}\right),$$

где n_s – число исходных кадров, n_d – число предсказываемых кадров, N_{sp} – число пикселей исходных кадров, а $N_{\text{полн}}$ – число пикселей всех кадров, требует отдельного исследования. В данной работе N_{sp}, а значит, и $N_{\text{полн}}$ функционально зависят от N_p, что не позволяет выполнить полноценное исследование влияния отдельных компонент на t_p. Отдельного рассмотрения также требует большое число предсказываемых кадров n. Предполагается, что по мере исчерпания памяти видеокарты зависимость между t_p и N_p перестанет быть линейной и будет ближе к степенной при степени, лежащей в интервале (0; 1).

6. Прогнозирование времени предсказания кадров

Построена линейная регрессионная модель для предсказания времени предсказания кадров от числа пикселей исходных кадров при выполнении неоптимизированной нейронной сети DMVFN на описанном ранее оборудовании $\hat{t}_P = 0.0000378 \times N_p\left(\text{мс}\right)$.

При доверительной вероятности, равной 95 %, числе испытаний 18000 (по 1000 в каждом эксперименте) на основе (4) вычислена интервальная оценка прогнозируемого времени предсказания t_p (рис. 9):

$$\hat{t}_P \in \left[0.0000378 \times N_p \pm 1767.15255 \times \sqrt{\frac{1}{17998} \times \left(1 + \frac{\left(4058453.33333 - N_p\right)^2}{27857767920071.113} \right)} \right]. \quad (5)$$

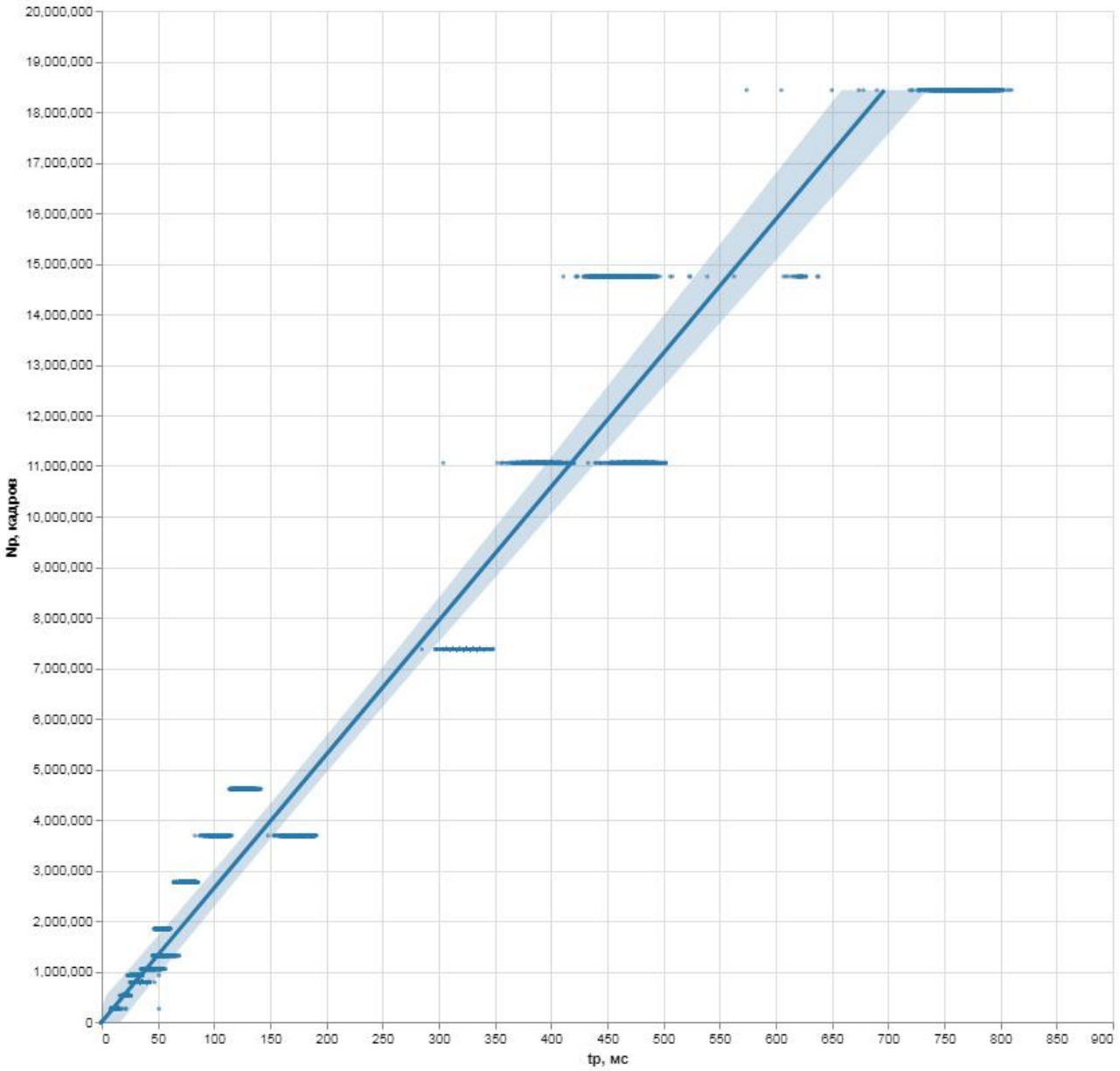

Рис. 9. Интервальная оценка предложенной регрессионной модели

Как можно видеть из рис. 9, модель гетероскедастична: в то время как математическое ожидание ошибки близко к нулю, дисперсия ошибки модели возрастает при возрастании N_p. Соответственно, оценки, получаемые с помощью данной модели, при достаточно высоких N_p неэффективны. Тем не менее в рамках текущей работы данная модель используется в зоне достаточно малой дисперсии ошибки (её области компетенции). В качестве альтернативы возможно увеличение интервала оценок при достаточно больших N_p.

Выдвинем гипотезу, требующую для подтверждения отдельного исследования. Пусть n насколько мало, что при выполнении нейронной сети DMVFN на CUDA видеопамять видеокарты далека от исчерпания. Тогда линейный характер корреляционной зависимости ве-

личин t_p и N_p сохраняется. Соответственно, приведённая выше модель может быть актуализирована и для другого оборудования.

Для актуализации регрессионной модели не обязательно повторять все проведённые эксперименты с нуля. Достаточно получить среднее время выполнения t'_{pcp} при некотором множестве N предсказываемых кадров с суммарным числом пикселей N_p. Тогда регрессионная модель при достаточно малых n может в общем случае быть представлена как:

$$\hat{t}_P = \frac{0.0000378 \times N_p}{k_p} (\text{мс}), \tag{6}$$

где k_p показывает, во сколько раз быстрее нейронная сеть выполняется на видеокарте A100, чем на тестируемой видеокарте, и вычисляется по формуле

$$k_p = \frac{t_{\text{pcp}}}{t'_{\text{pcp}}}$$

при

$$t'_{\text{pcp}} = \frac{\sum_{t'_p : N_p \in N} (t_p)}{x}.$$

где x – число испытаний на тестируемой видеокарте.

В свою очередь, t_{pcp} для нескольких N_p можно получить как среднее арифметическое значение при каждом конкретном значении N_p, т.к. в данной работе в каждом эксперименте – равное число испытаний (1000 шт.), а каждому N_p соответствует свой эксперимент. Обозначим данную регрессионную модель как *модель DMVFN малых n*. Соответственно, разновидности этой модели для различных тестируемых видеокарт – *частными случаями модели DMVFN малых n*.

Пусть k_p достаточно точно отражает отношение t_p к t'_p для всех N_p, на которых построена исходная регрессионная модель. Тогда интервал значений для модели DMVFN малых n элементарно выводится из формул (4), (5), (6) и может быть приблизительно рассчитан как

$$\hat{t}_P \in \left[\frac{0.0000378 \times N_p}{k_p} \pm \sqrt{\frac{1378.8659^2 \times 17998 + \sum \left(t'_p - \hat{t}_p \right)^2}{17998 + x}} \times t_{0,05,17998+x} \times \right.$$

$$\left. \times \sqrt{\frac{1}{17998 + x} \cdot \left(1 + \frac{(17999 + x) \times \left(\frac{4058453.33333 \times 18000 + \sum N'_p}{18000 + x} - N_p \right)^2}{27857767920071.113 \times 17999 + \sum \left(N'_p - \frac{4058453.33333 \times 18000 + \sum N'_p}{18000 + x} \right)} \right)} \right] (\text{мс}),$$

где N'_p – это N_p из испытаний на тестовой видеокарте, для которой находится частный случай модели DMVFN малых n. Данная формула основана на пересчёте числа степеней свободы, стандартной ошибки оценки, среднего значения и дисперсии.

7. Оценка условий увеличения FPS блоком предсказания

На основании (2) и (3) задачу максимизации FPS для текущей (рис. 2) конфигурации нейросетевого декодера можно свести к минимизации:

$$\text{FPS} = \frac{(n+1) \times 1000}{t_{\text{DC}} + t_{\text{DCD}} + t_I + t_P + T_з} \left(\frac{\text{кадров}}{\text{с}} \right),$$

где $T_з$ – некоторая дополнительная задержка до поступления кадра из канала связи.

Время выполнения декомпрессии t_{DC} алгоритмом DEFLATE в рамках блока декомпрессии измеряется в наносекундах и несущественно относительно времени выполнения других блоков.

Время декодирования кадра VQ-VAE-f16 в рамках блока декодирования приблизительно одинаково и на видеокарте A100 в среднем составляет 26.3 мс.

Значительные флуктуации средних оценок t_I при одних и тех же операциях блока интерполяции $I(X)$ не позволяют дать корректную оценку с помощью регрессионной модели, как в случае блока предсказания $P(x)$. Для каждого разрешения использованы средние t_I за 6000 испытаний (табл. 1). Таким образом, для разрешения 512×512 t_I = 1.08 мс, для разрешения 1280×720 – 1.71 мс, а для 2560×1440 – 1.99 мс. При этом существенная часть времени выполнения блока интерполяции $I(X)$ – это перевод формата из torch.tensor и numpy.ndarray, а время работы алгоритма бикубической интерполяции ввиду сопоставимости значений обоих показателей и их зависимости от размерности кадра выделить без проведения отдельных измерений не представляется возможным.

Оценим минимальную дополнительную задержку $T_{\min} \geq 0$, при которой возможно увеличение FPS с помощью предиктора. Первым шагом к решению данной задачи является нахождение минимальной дополнительной задержки поступления кадра из канала связи $T_з \geq 0$, такой что неравенство

$$\frac{1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з} < \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + t_P + T_з},$$

$$\frac{1000}{26.2718 + t_I + T_з} < \frac{(n+1)\times 1000}{26.2718 + t_I + 0.0000378\times N_p + T_з},$$

$$\frac{1000}{26.2718 + t_I + T_з} < \frac{(n+1)\times 1000}{26.2718 + t_I + 0.0000378\times n\times \mathrm{width}\times \mathrm{height} + T_з}$$

имеет решение для некоторых n>0. Для разрешения 512×512:

$$\frac{1000}{26.2718 + 1.0833 + T_з} < \frac{(n+1)\times 1000}{26.2718 + 1.0833 + 0.0000378\times n\times 512\times 512 + T_з},$$

$$\frac{1000}{27.3551 + T_з} < \frac{(n+1)\times 1000}{27.3551 + 9.9090432\times n + T_з}.$$

С учетом

$$\forall T_з : T_з \geq 0 \Rightarrow \forall T_з : (27.3551 + T_з) > 0,$$

$$\forall n, \forall T_з : n > 0, T_з \geq 0 \Rightarrow \forall n, \forall T_з : (27.3551 + 9.9090432\times n + T_з) > 0$$

справедливы следующие неравенства

$$1000\times(27.3551 + 9.9090432\times n + T_з) < (n+1)\times 1000\times(27.3551 + T_з),$$

$$27355.1 + 9909.0432\times n + 1000\times T_з < 27355.1\times n + 1000\times Tз\times n + 27355.1 + 1000\times T_з,$$

$$9909.0432\times n < 27355.1\times n + 1000\times Tз\times n.$$

Следует отметить, что при n = 0 обе стороны неравенства равны, что показывает правильность выполняемых действий. В данном случае предсказание вырождается: предсказание 0 кадров выполняется за 0 секунд, при этом не добавляя кадров, а FPS становится таким же, как и в случае без блока предсказания. Соответственно, $T_з \geq 0$.

Таким образом, при заданных условиях и ограничениях использование модели DMVFN для предсказания изображений разрешения 512×512 не требует дополнительной задержки $T_з$. Это позволяет оценить $T_{\min}$. Если дополнительная задержка $T_з > 0$, то $T_{\min}$ равен сумме времени обработки нейросетевым декодером предыдущего кадра и дополнительной задержки. Если же $T_{з\,\min} = 0$, то кадр из канала связи может поступать как в момент завершения обработки декодером предыдущего кадра, так и во время выполнения блоков нейросетевого декодера. Это не оказывает влияния на FPS, т.к. последний кадр буферизируется и поступает

на блок декомпрессии $DC(T^*)$ сразу по завершении обработки нейросетевым декодером предыдущего кадра. Следовательно, в этом случае $T_{\min} = 0$.

Перед анализом остальных разрешений выведем более общую формулу, которая может быть использована в том числе для других видеокарт при достаточно малых n. В случае другого оборудования время выполнения декомпрессии t_{DC} может не быть пренебрежительно мало. По условию

$$\forall T_з, \forall t_I, \forall t_{\text{DC}} : T_з, t_I, t_{\text{DC}} \geq 0, \forall \text{width}, \forall \text{height}, \forall n, \forall t_{\text{DCD}} : \text{width}, \text{height}, n, t_{\text{DCD}} > 0$$

для всех переменных выполняются неравенства

$$\forall T_з, \forall t_I, \forall t_{\text{DC}}, \forall t_{\text{DCD}} : t_{\text{DC}} + t_{\text{DCD}} + t_I + T_з > 0,$$

$$\forall T_з, \forall t_I, \forall t_{\text{DC}}, \forall t_{\text{DCD}}, \forall n, \forall \text{width}, \forall \text{height} :$$

$$t_{\text{DC}} + t_{\text{DCD}} + t_I + \frac{0.0000378 \times n \times \text{width} \times \text{height}}{k_p} + T_з > 0.$$

Следовательно, возможно свести решение

$$\frac{1000}{t_{\text{DC}} + t_{\text{DCD}} + t_I + T_з} < \frac{(n+1) \times 1000}{t_{\text{DC}} + t_{\text{DCD}} + t_I + \frac{0,0000378 \times n \times \text{width} \times \text{height}}{k_p} + T_з}$$

для нахождения допустимых $T_з$ к:

$$1000 \times \left(t_{\text{DC}} + t_{\text{DCD}} + t_I + \frac{0.0000378 \times n \times \text{width} \times \text{height}}{k_p} + T_з \right) <$$

$$< (n+1) \times 1000 \times (t_{\text{DC}} + t_{\text{DCD}} + t_I + T_з)$$

$$1000 \times t_{\text{DC}} + 1000 \times t_{\text{DCD}} + 1000 \times t_I + 1000 \times T_з +$$

$$+ \frac{0.0378 \times n \times \text{width} \times \text{height}}{k_p} < 1000 \times n \times t_{\text{DC}} + 1000 \times n \times t_{\text{DCD}} +$$

$$+ 1000 \times n \times t_I + 1000 \times n \times T_з + 1000 \times t_{\text{DC}} + 1000 \times t_{\text{DCD}} + 1000 \times t_I + 1$$

$$\frac{0.0378 \times n \times \text{width} \times \text{height}}{k_p} < 1000 \times n \times (t_{\text{DC}} + t_{\text{DCD}} + t_I + T_з),$$

$$\frac{0.0000378 \times \text{width} \times \text{height}}{k_p} < t_{\text{DC}} + t_{\text{DCD}} + t_I + T_з.$$

Соответственно, $T_з$ и $T_{\min}$, при которых предикторная модель DMVFN позволяет увеличить FPS нейросетевого декодера, можно оценить как

$$\begin{cases} T_з > \dfrac{0.0000378 \times \text{width} \times \text{height}}{k_p} - t_{\text{DC}} - t_{\text{DCD}} - t_i \\ T_з \geq 0 \end{cases}, \tag{7}$$

$$T_{\min} = \begin{cases} 0, & \dfrac{0.0000378 \times \text{width} \times \text{height}}{k_p} - t_{\text{DC}} - t_{\text{DCD}} - t_i \leq 0 \\ t_{DC} + t_{DCD} + t_I + t_P + T_{з\min}, [...] > 0 \end{cases}. \tag{8}$$

Вторую часть выражения (8) можно значительно упростить:

$$T_{\min} = t_{\text{DC}} + t_{\text{DCD}} + t_I + t_P + \frac{0.0000378 \times \text{width} \times \text{height}}{k_p} - t_{\text{DC}} - t_{\text{DCD}} - t_I,$$

$$T_{\min} = t_P + \frac{0.0000378 \times \text{width} \times \text{height}}{k_p},$$

$$T_{\min} = \frac{(n+1)\times 0.0000378\times \text{width}\times \text{height}}{k_p}.$$

Таким образом, (8) преобразуется в

$$T_{\min} = \begin{cases} 0, & \dfrac{0.0000378\times \text{width}\times \text{height}}{k_p} - t_{\text{DC}} - t_{\text{DCD}} - t_I \le 0 \\ \dfrac{(n+1)\times 0.0000378\times \text{width}\times \text{height}}{k_p}, & \dfrac{0.0000378\times \text{width}\times \text{height}}{k_p} - t_{\text{DC}} - t_{\text{DCD}} - t_I > 0 \end{cases} \quad (9)$$

При определённом условии (9) $T_{\min}$ может быть оценен как время предсказания t_P (n+1) кадра.

Проверка полученных оценок для ранее рассчитанного частного случая видеокарты A100 и разрешения 512×512 показала соответствие ранее полученному результату $T_з \ge 0$. Соответственно, (7) и (9) можно применять для оценки целесообразности использования блока предиктора модели DMVFN при использовании для оценки времени выполнения предиктора t_P модели DMVFN малых n. Так как решение (7) существует и единственно, подтверждается ещё одно утверждение: при выполнении модели DMVFN на любом оборудовании при любой конфигурации нейросетевого декодера обязательно найдётся такое $T_{\min}$, при котором использование блока предсказания $P(x)$ даёт увеличение FPS.

Оценки для остальных разрешений: (1280×720) $T_з > 6{,}8376\times(n+1)$; (2560×1440) $T_з > 139{,}3459\times(n+1)$. Безусловного увеличения FPS не наблюдается.

Полученные результаты хорошо согласуются с экспериментальными данными. Если в случае разрешения 512×512 при использовании DMVFN FPS однозначно повышается, то в случае 1280×720 FPS колеблется, а в случае 2560×1440 – уменьшается и колеблется.

Гипотеза, доказательство которой необходимо для реализации анализатора КИО и его связки с КИВ. Пусть исходный период прихода кадров с кодера $T_к \ge T_{\min}$. Тогда возможно нахождение такого n, что добавление блока предсказания DMVFN в декодер повышает FPS.

Доказательство. *Пусть $T_{min} = 0$, тогда добавление блока предсказания DMVFN безусловно увеличивает FPS. Пусть $T_k > T_{min}$. Тогда, исходя из (9), такие параметры n могут быть найдены как*

$$T_k > \frac{(n+1)\times 0.0000378\times \text{width}\times \text{height}}{k_p},$$

$$n < \frac{T_k \times k_p}{0.0000378\times \text{width}\times \text{height}} - 1.$$

Тем не менее сам факт использования блока предсказания означает, что $n \ge 1$. Поставим условие существования решения

$$\frac{T_k \times k_p}{0.0000378\times \text{width}\times \text{height}} - 1 \ge 1,$$

$$T_k > \frac{2\times 0.0000378\times \text{width}\times \text{height}}{k_p}.$$

Иными словами, задержка должна быть выше, чем время предсказания двух кадров с помощью DMVFN и общее время работы декодера. Исходная гипотеза подтверждается, таким образом, лишь в этом частном случае.

Теорема о целесообразности блока предсказания КИВ. *Если период y_i поступления кадров из канала связи на декодер больше общего времени работы декодера и больше или равен времени предсказания двух кадров нейронной сетью DMVFN, существует такое число предсказываемых кадров n, что для любых разрешений кадров добавление блока предсказания DMVFN в нейросетевой декодер обеспечивает увеличение FPS.*

Первая задача работы, которая заключается в оценке целесообразности использования блока DMVFN, выполнена.

8. Анализ максимального FPS

Оценим максимальное FPS (FPS_{max}) для каждого из разрешений: 512×512, 1280×720 и 2560×1440. Справедливо выражение

$$\mathrm{FPS} = \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + t_P + T_з}\left(\frac{\text{кадров}}{\text{с}}\right). \tag{10}$$

Преобразуем (10) в

$$\mathrm{FPS} = \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з}\left(\frac{\text{кадров}}{\text{с}}\right).$$

Если t_{DC}, t_{DCD}, t_I, $T_з$, width и height обязательны при увеличении уменьшают FPS, то интерес представляет последний параметр – число предсказываемых кадров n. Возьмём производную FPS по n:

$$\frac{\delta \mathrm{FPS}}{\delta n} = \delta\left(\frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з}\right)\Big/\delta n,$$

$$\frac{\delta \mathrm{FPS}}{\delta n} = \delta\left(\frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з}\right)\Big/\delta n,$$

$$\frac{\delta \mathrm{FPS}}{\delta n} = \frac{1000\times\left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з\right)}{\left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з\right)^2} -$$

$$- \frac{\dfrac{0.0378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + \dfrac{0,0378\times \mathrm{width}\times \mathrm{height}}{k_p}}{\left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з\right)^2},$$

$$\frac{\delta \mathrm{FPS}}{\delta n} = \frac{1000\times\left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з\right) - \dfrac{0.0378\times \mathrm{width}\times \mathrm{height}}{k_p}}{\left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \dfrac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з\right)^2}.$$

Производная функции FPS от n имеет кратный разрыв 2-го рода в точке

$$t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + \frac{0.0000378\times n\times \mathrm{width}\times \mathrm{height}}{k_p} + T_з = 0,$$

$$n = -\frac{k_p \times \left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з\right)}{0.0000378 \times n \times \mathrm{width} \times \mathrm{height}}. \quad (11)$$

Тем не менее, так как $n > 0$, что вследствие дискретности n тождественно $n \geq 1$, а кратная точка разрыва (11) имеет значение меньше нуля, она выпадает из рассмотрения. Соответственно, при $n \geq 1$ FPS является гладкой функций и знак производной FPS от n одинаков. Существует три случая задачи максимизации FPS: 1) когда по мере увеличения n он убывает, 2) когда по мере увеличения n он возрастает и 3) когда он статичен. Рассмотрим условие первого случая:

$$1000 \times \left(t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з\right) - \frac{0.0378 \times \mathrm{width} \times \mathrm{height}}{k_p} < 0,$$

$$t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з < \frac{0.0000378 \times \mathrm{width} \times \mathrm{height}}{k_p}.$$

Его можно сформулировать следующим образом: время предсказания одного кадра превышает время выполнения всех остальных операций декодера. В этом случае $\mathrm{FPS}_{\max}$ достигается минимизацией n, то есть предсказанием исключительно одного кадра.

Условие второго случая обратно первому

$$t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з > \frac{0.0000378 \times \mathrm{width} \times \mathrm{height}}{k_p}. \quad (12)$$

Время предсказания одного кадра меньше времени выполнения всех остальных операций декодера. В этом случае определим *теоретический предельный FPS*. Он достигается при стремлении n к бесконечности, и его фактическое достижение требует бесконечной видеопамяти используемой видеокарты, что практически невозможно.

Условие третьего случая представляется выражением

$$t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_I + T_з = \frac{0.0000378 \times \mathrm{width} \times \mathrm{height}}{k_p}.$$

Если число предсказываемых кадров n достаточно мало, так что при выполнении DMVFN на CUDA видеопамять используемой видеокарты далека от исчерпания, то для всех таких n FPS неизменен. Таким образом, для экономии видеопамяти следует предсказывать в данном случае только 1 кадр.

$\mathrm{FPS}_{\max}$ для разрешений 512×512, 1280×720 и 2560×1440 при использовании видеокарты A100 соответственно равно: 100.9179, 31.8365 и 11.9323 к/с.

Полученные результаты в целом соотносятся с экспериментальными данными. В то время как при увеличении n при 512×512 наблюдалось стойкое увеличение FPS, при 1280×720 были некоторые флуктуации как с увеличением, так и с уменьшением, а при 2560×1440 – явное понижение FPS.

В то же время результаты показывают ограниченность модели. Согласно экспериментам, $\mathrm{FPS}_{\max}$ (при $n = 1$) для 1280×720 составил 34.6748 к/с, а для 2560×1440 – 9.7956 к/с, при этом при больших n для данных разрешений удалось достичь незначительно больших FPS. Данные особенности объясняются ограниченностью модели и сильными флуктуациями FPS в процессе измерений. Тем не менее можно утверждать, что предложенный метод применим при необходимости достаточно грубой оценки и проектировании систем анализатора КИО и модулей КИВ.

9. Анализ конфигурации блоков нейросетевого декодера

Процесс интерполяции и прогнозирования рассматриваемыми в настоящей работе моделями невозможно выполнить в латентном пространстве признаков, соответственно, данные блоки должны располагаться после блока декодирования *DCD*(*T*). Декомпрессию нельзя

проводить после декодирования, так как для декодирования необходимо наличие уже декомпрессированных данных. Следовательно, блок декомпрессии $DC(T^*)$ должен быть расположен до блока декодирования. С учетом этого возможна только одна альтернативная конфигурация нейросетевого декодера: прогнозирование кадров с последующей их интерполяцией. FPS для такой конфигурации оценивается как (рис. 10):

$$\mathrm{FPS} = \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_{P(512\times512)} + (n+1)\times t_I + T_з}\left(\frac{\text{кадров}}{\text{с}}\right).$$

Данный вариант обладает следующими преимуществами: значительное уменьшение времени прогнозирования, возможность полноценного увеличения FPS нейросетевого декодера (см. предыдущий раздел) и вместе с тем предоставление кадров видеопотока оператору БВС в нужном разрешении. При этом есть и недостаток: при использовании для увеличения разрешения кадров моделей нейронных сетей сверхразрешения [13–14] возможно значительное уменьшение FPS.

Рис. 10. Альтернативный нейросетевой декодер

Данная конфигурация нейросетевого декодера (рис. 10) предпочтительней исходной (рис. 2) при определённых условиях:

$$\frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_{P(512\times512)} + (n+1)\times t_I + T_з} > \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + t_P + t_I + T_з},$$

$$\frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + n\times\dfrac{0.0000378\times512\times512}{k_p} + (n+1)\times t_I + T_з} >$$

$$> \frac{(n+1)\times 1000}{t_{\mathrm{DC}} + t_{\mathrm{DCD}} + n\times\dfrac{0.0000378\times \mathrm{width}\times \mathrm{height}}{k_p} + t_I + T_з},$$

$$t_{\mathrm{DC}} + t_{\mathrm{DCD}} + n\times\frac{0.0000378\times \mathrm{width}\times \mathrm{height}}{k_p} + t_I + T_з >$$

$$> t_{\mathrm{DC}} + t_{\mathrm{DCD}} + n\times\frac{0.0000378\times512\times512}{k_p} + (n+1)\times t_I + T_з,$$

$$n\times\frac{0.0000378\times \mathrm{width}\times \mathrm{height}}{k_p} + t_I > n\times\frac{0.0000378\times512\times512}{k_p} + n\times t_I + t_I,$$

$$n \times \frac{0.0000378 \times \text{width} \times \text{height}}{k_p} > n \times \frac{0.0000378 \times 512 \times 512}{k_p} + n \times t_I$$

$$t_I < \frac{1}{k_p} \times \left(0.0000378 \times \text{width} \times \text{height} - 9.9090432\right).$$

Под предпочтительностью понимается более высокое FPS.

В результате получено ограничение на время интерполяции одного кадра, при котором альтернативная модель нейросетевого декодера является предпочтительной. При использовании бикубической интерполяции данное условие выполняется при разрешении, которое несколько больше чем 512×512, в том числе для разрешений 1280×720 и 2560×1440.

Тем не менее переход к альтернативной архитектуре нейросетевого декодера не позволяет увеличить FPS при разрешении кадров видеопотока 512×512. Так как время бикубической интерполяции t_I очень мало по сравнению со временем прогнозирования t_P, случай нахождения FPS_{max} тот же, что и у прогнозирования кадров разрешения 512x512 – второй (при увеличении n FPS увеличивается (12)). Через предел n, стремящегося к бесконечности, применительно к функции FPS для альтернативной архитектуры нейросетевого декодера, рассчитывается FPS_{max} для разрешения 1280×720 – 86.04 к/с и для 2560×1440 – 84 к/с.

В случае, если время интерполяции сопоставимо со временем предсказания, альтернативная архитектура дает результаты, сравнимые с исходной. Соответственно, достижение большего FPS в этом случае связано с оптимизацией и выбором альтернативных моделей интерполяции и прогнозирования, что, вероятнее всего, приведёт к ухудшению качества изображения, подаваемого оператору БВС. В этой связи перед использованием тех или иных моделей в рамках нейросетевого декодера должен быть проведен анализ FPS_{max} и условий достижения данного значения FPS.

10. Заключение

Полученные результаты показывают возможность применения блока прогнозирования для увеличения FPS на стороне оператора БВС. Экспериментально наибольший FPS в 74.98 кадров в секунду был достигнут при выходном разрешении 512×512, модели предсказания DMVFN и предсказании 5 кадров. При выходном разрешении теоретический предел увеличения FPS составляет 100.92 кадров в секунду, к которому можно приблизиться при увеличении числа предсказываемых кадров. Кроме того, при использовании предложенной альтернативной архитектуры теоретический предел при выходном разрешении 1280×720 составляет 86.04 кадров в секунду, а при выходном разрешении 2560×1440 – 84 кадра в секунду.

На основе предложенной регрессионной модели времени предсказания, модели DMVFN малых n, в настоящей работе был получен ряд результатов.

1. Время декомпрессии алгоритмом LZMA в рамках нейросетевого декодера является крайне малым и не оказывает заметного влияния на время обработки поступающего с кодера кадра нейросетевым декодером.

2. При достаточно малом числе предсказываемых кадров зависимость между временем предсказания и числом предсказываемых кадров близка к линейной, что обуславливается линейной зависимостью между числом пикселей предсказываемых изображений.

3. При разрешении кадров видеопотока 512x512 использование модели DMVFN безусловно приводит к увеличению FPS на нейросетевом декодере. В то же время при разрешениях кадров 1280x720 и 2560x1440 увеличение FPS возможно при задержке прихода кадров с кодера, превышающей время работы декодера. При отсутствии таковой задействование модели приводит к колебанию либо уменьшению FPS.

4. Всегда найдётся такая достаточно высокая задержка прихода кадров с кодера, что использование модели DMVFN позволит увеличить FPS.

5. Если исходный период прихода кадров с кодера больше общего времени работы декодера и больше или равен времени предсказания двух кадров нейронной сетью DMVFN, тогда для любых разрешений кадров возможно добавление в декодер блока предсказания DMVFN с таким числом предсказываемых кадров, что FPS на стороне оператора БВС повысится. Для некоторых отдельных разрешений кадров, таких как 512x512, добавление блока предсказания позволяет безусловно увеличивать FPS.

6. Если время предсказания одного кадра нейронной сетью DMVFN больше или равно сумме времён выполнения остальных операций нейросетевого декодера, стоит предсказывать только один кадр. В противном случае – наибольшее количество, которое позволяет видеопамять используемой для работы нейронной сети видеокарты.

7. Если используется интерполяция кадров видеопотока, то при достаточно малом времени её выполнения можно увеличить FPS нейросетевого декодера за счёт перестановки местами блоков интерполяции и прогнозирования.

Предложенные в настоящей работе модели и методы формализуют проблему использования моделей предсказания в нейросетевых декодировщиках и могут служить в качестве методической основы для проектирования соответствующих систем.

Возможны несколько направлений развития исследования.

1. Исследование влияния числа пикселей исходных и предсказываемых кадров на время предсказания.

2. Проведение испытаний на ином оборудовании.

3. Проведение испытаний с б*о*льшим числом предсказываемых кадров.

4. Рассмотрение возможности интерполяции и предсказания кадров видеопотока в латентном пространстве.

5. Проверка и уточнение модели прогнозирования времени предсказания кадров с помощью DMVFN для других видеокарт.

6. Построение модели нейросетевого декодера с учётом блока квантования.

7. Проведение испытаний с набором данных достаточно большого размера, чтобы существенные флуктуации в измеряемых временных показателях и FPS в существенной степени сглаживались.

8. Построение гомоскедастичной регрессионной модели прогнозирования времени предсказания.

Литература

1. *Чмелев В. С., Калюка В. И., Дмитренко М. Е.* Обзор систем управления беспилотных летательных аппаратов общего пользования // Сборник материалов научно-практической конференции «Технологии. Инновации. Связь», Санкт-Петербург, 19 апреля 2021 года. С. 279–286.
2. Стратегия развития отрасли связи Российской Федерации на период до 2035 года [Электронный ресурс] // Правительство России: официальный сайт. 2023. URL: http://government.ru/news/50304/ (дата обращения: 21.05.2024).
3. Юрий Борисов выступил на стратегической сессии, посвящённой серийному производству спутников [Электронный ресурс] // Роскосмос: официальный сайт. 2023. URL: https://www.roscosmos.ru/40096/ (дата обращения: 21.05.2024).
4. Число учтённых в России дронов выросло на треть [Электронный ресурс] // Федеральное агентство воздушного транспорта – Росавиация: официальный сайт, 2024. URL: https://favt.gov.ru/novosti-novosti/?id=10676 (дата обращения: 21.05.2024).
5. *Березкин А. А., Паршин А. А., Лазарев А. А.* Адаптивный контроль интенсивности видеопотока при передаче FPV-трафика беспилотных систем // Материалы 79-й научно-

технической конференции СПб НТО РЭС им. А. С. Попова, посвященной Дню радио, Санкт-Петербург, 22–26 апреля 2024. С. 162–165.
6. *Березкин А. А., Савелов Д. Ю., Суходоева А. В., Туманов И. А., Киричек Р. В.* Исследование нейросетевых моделей предсказания видеопотока при управлении беспилотными системами от первого лица // Труды Научно-исследовательского института радио. 2023. № 3–4. С. 40–56.
7. *Гончаров В. Г., Березкин А. А.* Анализ моделей предсказания кадров FPV-видеопотока в каналах информационного обмена беспилотных систем // Материалы 79-й научно-технической конференции СПб НТО РЭС им. А. С. Попова, посвященной Дню радио, Санкт-Петербург, 22–26 апреля 2024. С. 194–197.
8. *Березкин А. А., Вивчарь Р. М., Киричек Р. В., Захаров А. А.* Метод декомпрессии FPV-видеопотока от беспилотных систем на основе латентной диффузионной нейросетевой модели // Электросвязь. 2024. № 1. С. 42–53.
9. *Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.* High-resolution image synthesis with latent diffusion models // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022. P. 10684–10695.
10. *Hu X., Huang Z., Huang A., Xu J., Zhou S.* A dynamic multi-scale voxel flow network for video prediction // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. P. 6121–6131.
11. *Березкин А. А., Вивчарь Р. М., Слеппев А. В., Киричек Р. В., Захаров А. А.* Метод сжатия видеопотока при управлении беспилотными системами в гибридных орбитально-наземных сетях связи // Электросвязь. 2023. № 10. С. 48–56.
12. *Бардасов С. А.* Эконометрика: учебное пособие. 2-е издание, переработанное и дополненное. Тюмень: Издательство Тюменского государственного университета, 2010. 264 с.
13. *Berezkin A. A., Phuc H. Do., Tran D. Le., Kirichek R. V.* A comparative analysis of diffusion-based super-resolution techniques in a video stream compression system in FPV control of unmanned systems // Труды Научно-исследовательского института радио. 2023. № 3–4. P. 26–39.
14. *Козлова А. И., Облаков Н. А., Березкин А. А.* Анализ моделей предсказания кадров FPV-видеопотока в каналах информационного обмена беспилотных систем // Материалы 79-й научно-технической конференции СПб НТО РЭС им. А. С. Попова, посвященной Дню радио, Санкт-Петербург, 22–26 апреля 2024. С. 170–173.

Березкин Александр Александрович
к.т.н., доцент кафедры программной инженерии и вычислительной техники (ПИиВТ), Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича (СПбГУТ, 193232, Санкт-Петербург, пр. Большевиков, д. 22, к. 1), директор Центра перспективных проектов и разработок, e-mail: `berezkin.aa@sut.ru`, ORCID ID: 0000-0002-1748-8642.

Ченский Александр Александрович
магистрант кафедры ПИиВТ, инженер Центра перспективных проектов и разработок, СПбГУТ, e-mail: `chenskii.aa@sut.ru`, ORCID ID: 0009-0005-0832-8590.

Киричек Руслан Валентинович
д.т.н., ректор, профессор кафедры ПИиВТ, СПбГУ, тел. +7 812 3051 200, e-mail: `kirichek@sut.ru`, ORCID ID: 0000-0002-8781-6840.

Авторы прочитали и одобрили окончательный вариант рукописи.
Авторы заявляют об отсутствии конфликта интересов.

Вклад соавторов: *Каждый автор внес равную долю участия как во все этапы проводимого теоретического исследования, так и при написании разделов данной статьи.*

Research of Video Stream Intensity Limits in UAV FPV Control in Frame Prediction Mode. Part I: Models and Methods

Alexander A. Berezkin, Alexander A. Chenskiy, Ruslan V. Kirichek

Bonch-Bruevich State university of telecommunications (SPbSUT)

Abstract: Currently, unmanned aerial vehicles have found wide application in various spheres of the national economy. FPV control refers to a method of controlling unmanned aerial vehicles in which the video stream from the unmanned aircraft is transmitted to the remote pilot station in real time. Due to failures in the communication network, packets with video stream data may be lost or delayed and be delivered late. One of the ways to compensate for the drop in FPS in case of frame loss or delay at the remote pilot station is to predict intermediate frames of the video stream. This paper presents a scheme for predicting intermediate frames of a video stream and presents the results of experiments to establish a realistically possible increase in FPS in a neural network codec, in which various autoencoders such as VQ-f16 are used for video compression as well as the lossless latent feature space compression algorithm DEFLATE, while the dynamic multiscale voxel flow neural network DMVFN is used for prediction. A regression model was developed to predict the prediction time. The task of analyzing FPS for various configurations of the neural network decoder on the side of the external pilot is formalized.

Keywords: FPS, variational autoencoder, neural decoder, DMVFN, neural networkы, video stream prediction.

For citation: Berezkin A. A., Chenskiy A. A., Kirichek R. V. Research of video stream intensity limits in UAV FPV control in frame prediction mode. Part I: models and methods (in Russian). *Vestnik SibGUTI*, 2024, vol. 18, no. 3, pp. 115-139. https://doi.org/10.55648/1998-6920-2024-18-3-115-139.

The article was submitted: 05.06.2024; accepted for publication 20.06.2024.

References

1. Chmelev B. S., Kalyuka V. I., Dmitrenko M. E. Obzor sistem upravlenija bespilotnyh letatel'nyh apparatov obshhego pol'zovanija [Overview of general purpose unmanned aerial vehicle control systems]. *Tehnologii. Innovacii. Svjaz': Sbornik materialov nauchno-prakticheskoj konferencii, Sankt-Peterburg, 19 April, 2021*. Sankt-Peterburg, Federal'noe gosudarstvennoe kazjonnoe voennoe uchrezhdenie vysshego obrazovanija "Voennaja akademija svjazi imeni marshala Sovetskogo Sojuza S. M. Budjonnogo" Ministerstva oborony Rossijskoj Federacii, 2022. pp. 279-286.
2. *Strategija razvitija otrasli svjazi Rossijskoj Federacii na period do 2035 goda* [The strategy for the development of the telecommunications industry of the Russian Federation for the period until 2035], available at: http://government.ru/news/50304/ (accessed: 21.05.2024).
3. *Jurij Borisov vystupil na strategicheskoj sessii, posvjashhjonnoj serijnomu proizvodstvu sputnikov* [Yura Borisov made a presentation at the strategic session on serial production of satellites], available at: https://www.roscosmos.ru/40096/ (accessed: 21.05.2024).
4. *Chislo uchtjonnyh v Rossii dronov vyroslo na tret'* [The number of drones registered in Russia increased by a third], available at: https://favt.gov.ru/novosti-novosti/?id=10676 (accessed: 21.05.2024).
5. Berezkin A. A., Parshin A. A., Lazarev A. A. Adaptivnyj kontrol' intensivnosti videopotoka pri peredache FPV-trafika bespilotnyh sistem [Adaptive control of video stream intensity in FPV traffic transmission of unmanned systems]. *79-ja nauchno-tehnicheskaja konferencija SPb NTO RES im. A.S. Popova, posvjashhennaja Dnju radio: sbornik dokladov regional'noj konferencii*, Saint Peters-

burg,
22-26 April, 2024, pp. 162-165.
6. Berezkin A. A., Savelov D. Ju., Suhodoeva A. V., Tumanov I. A., Kirichek R. V. Issledovanie nejrosetevyh modelej predskazanija videopotoka pri upravlenii bespilotnymi sistemami ot pervogo lica [Research of neural network models for video stream prediction in first-person control in unmanned systems]. *Trudy Nauchno-issledovatel'skogo instituta radio*, 202, no. 3-4, pp. 40-56.
7. Goncharov V. G., Berezkin A. A. Analiz modelej predskazanija kadrov FPV-videopotoka v kanalah informacionnogo obmena bespilotnyh sistem [Analysis of FPV video stream frame prediction models in information exchange links of unmanned systems]. *79-ja nauchno-tehnicheskaja konferencija SPb NTO RES im. A.S. Popova, posvjashhennaja Dnju radio: sbornik dokladov regional'noj konferencii*, Saint-Petersburg, 22-26 April, 2024, pp. 194-197.
8. Berezkin A. A., Vivchar' R. M., Kirichek R. V., Zaharov A. A. Metod dekompressii FPV-videopotoka ot bespilotnyh sistem na osnove latentnoj diffuzionnoj nejrosetevoj modeli [Method of decompression of FPV video stream from unmanned systems based on latent diffusion neural network model]. *Electrosvjaz'*, 2024, №1, pp. 42-53.
9. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684-10695.
10. Hu X., Huang Z., Huang A., Xu J., Zhou S. A dynamic multi-scale voxel flow network for video prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6121-6131.
11. Berezkin A. A., Vivchar' R. M., Slepnev A. V., Kirichek R. V., Zaharov A. A. Metod szhatija videopotoka pri upravlenii bespilotnymi sistemami v gibridnyh orbital'no-nazemnyh setjah svjazi [Video compression method for controlling unmanned systems in hybrid orbital-terrestrial communication networks]. *Electrosvjaz'*, 2023, no. 10, pp. 48-56.
12. Bardasov S. A. *Ekonometrika: uchebnoe posobie. 2 izdanie, pererabotannoe i dopolnennoe* [Econometrics: textbook. 2nd edition, revised and expanded]. Tjumen', Izdatel'stvo Tjumenskogo gosudarstvennogo universiteta, 2010. 264 p.
13. Berezkin A. A., Phuc H. Do., Tran D. Le., Kirichek R. V. A comparative analysis of diffusion-based super-resolution techniques in a video stream compression system in FPV control of unmanned systems. *Trudy Nauchno-issledovatel'skogo instituta radio*, 2023, no. 3-4, pp. 26-39.
14. Kozlova A. I., Oblakov N. A., Berezkin A. A. Analiz modelej predskazanija kadrov FPV-videopotoka v kanalah informacionnogo obmena bespilotnyh sistem [Analysis of FPV video stream frame prediction models in information exchange channels of unmanned systems]. *79-ja nauchno-tehnicheskaja konferencija SPb NTO RES im. A.S. Popova, posvjashhennaja Dnju radio: sbornik dokladov regional'noj konferencii*, Saint-Petersburg, 22-26 April, 2024, pp. 170-173.

Berezkin Alexander Alexandrovich

PhD, associate professor of the Program Engineering and Computer Science Department, Bonch-Bruevich State university of telecommunications (SPbSUT, 193232, Saint-Petersburg, Bolshevikov Avenue, 22-1), director of the Center of Perspective Projects and Developments, e-mail: berezkin.aa@sut.ru, ORCID ID: 0000-0002-1748-8642.

Chenskiy Alexander Alexandrovich

Master's student of the Program Engineering and Computer Science Department, engineer of Center of Perspective Projects and Developments, SPbSUT, e-mail: chenskii.aa@sut.ru, ORCID ID: 0009-0005-0832-8590.

Kirichek Ruslan Valentinovich

Doctor of technical science, rector, professor of the Program Engineering and Computer Science Department, SPbSUT, phone. +7 812 3051 200, e-mail: kirichek@sut.ru, ORCID ID: 0000-0002-8781-6840.