

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

На правах рукописи

Веселов Антон Игоревич

**ОБРАБОТКА ВИДЕОИНФОРМАЦИИ В СИСТЕМАХ
СЖАТИЯ, ОСНОВАННЫХ НА ПРИНЦИПАХ
КОДИРОВАНИЯ ЗАВИСИМЫХ ИСТОЧНИКОВ**

Специальность 05.12.13 —
«Системы, сети и устройства телекоммуникаций»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Тюрликов Андрей Михайлович

Санкт-Петербург — 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 СЖАТИЕ ВИДЕОДАНЫХ С ИСПОЛЬЗОВАНИЕМ ПРИНЦИПОВ КОДИРОВАНИЯ ЗАВИСИМЫХ ИСТОЧНИКОВ . .	10
1.1 Вводные замечания	10
1.2 Теоретические предпосылки распределенного видеокодирования .	13
1.3 Применение помехоустойчивого кодирования для сжатия информации	16
1.4 Пример реализации распределенного кодирования видеоисточников в рамках простой модели видеопотока	22
1.5 Классификация методов распределенного кодирования источников видеоинформации	27
1.6 Основные концепции распределенного кодирования источников видеоинформации	30
1.6.1 Концепция Стэнфорд	30
1.6.2 Концепция PRISM	32
1.7 Обоснование выбора концепции распределенного видеокодека . . .	35
1.8 Модель распределенного кодирования на базе проекта DISCOVER	35
1.9 Области применения распределенного кодирования	38
1.10 Выводы по разделу	40
2 ГЕНЕРАЦИЯ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ	41
2.1 Вводные замечания	41
2.2 Обзор методов генерации дополнительной информации	41
2.3 Аппроксимация промежуточных кадров в модели DISCOVER . . .	45
2.3.1 Основные определения и обозначения	45
2.3.2 Однонаправленная оценка движения	47
2.3.3 Билатеральная оценка движения	48
2.3.4 Сглаживание векторного поля	49
2.3.5 Интерполяция с компенсацией движения	50
2.3.6 Параметры процедуры аппроксимации кадров	50

2.3.7	Недостатки базового алгоритма	51
2.4	Модель истинного движения в задаче временной интерполяции кадров	51
2.5	Описание разработанного алгоритма аппроксимации промежуточных кадров с учетом истинного движения	54
2.5.1	Обоснование разработанного алгоритма	54
2.5.2	Оценка движения	56
2.5.3	Компенсация движения	62
2.6	Выводы по разделу	63
3	ОЦЕНКА ПАРАМЕТРОВ ОШИБОК В ВИРТУАЛЬНОМ КАНАЛЕ .	65
3.1	Вводные замечания	65
3.2	Существующие методы оценки параметров ошибок межкадрового предсказания для систем распределенного кодирования	67
3.2.1	Описание общепринятых допущений о виртуальном канале	67
3.2.2	Описание базового алгоритма оценки параметров ошибок межкадрового предсказания	69
3.2.3	Недостатки базового алгоритма	70
3.3	Предложенный модифицированный алгоритм оценки параметров ошибок межкадрового предсказания	73
3.4	Описание предложенной аналитической модели виртуального канала	80
3.5	Выводы	90
4	СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗРАБОТАННЫХ АЛГОРИТМОВ .	92
4.1	Вводные замечания	92
4.2	Критерии сравнения алгоритмов сжатия видеоданных	93
4.3	Формирование тестового множества видеопоследовательностей	95
4.4	Описание разработанной программной модели распределенного видеокодека	96
4.4.1	Обработка базовых кадров	97
4.4.2	Кодер промежуточных кадров	97
4.4.3	Декодер промежуточных кадров	100

4.4.4	Сравнение разработанного эталонного распределенного кодека с кодеком на основе базовой модели DISCOVER . . .	105
4.5	Схема эксперимента для сравнительной оценки разработанных алгоритмов	105
4.5.1	Общие замечания по сравнению алгоритмов обработки видеоинформации в схеме распределенного видеокодирования	105
4.5.2	Сравнительная оценка алгоритмов генерации дополнительной информации	107
4.5.3	Сравнение алгоритмов оценки параметров ошибок межкадрового предсказания	114
4.6	Сравнительный анализ разработанного алгоритма распределенного видеокодирования	118
4.7	Выводы по разделу	119
ЗАКЛЮЧЕНИЕ		122
СПИСОК ЛИТЕРАТУРЫ		124
СПИСОК РИСУНКОВ		131
СПИСОК ТАБЛИЦ		133
ПРИЛОЖЕНИЕ А СЛУЧАЙНЫЕ МАРКОВСКИЕ ПОЛЯ		134
ПРИЛОЖЕНИЕ Б ПАРАМЕТРЫ КОДЕКА КЛЮЧЕВЫХ КАДРОВ		138

ВВЕДЕНИЕ

В последнее время все большее распространение получает передача видеоданных от мобильных устройств по беспроводным сетям связи. Это связано как с существенным развитием технологий беспроводной передачи данных, так и с развитием мобильных устройств: беспроводных сенсоров, беспроводных камер видеонаблюдения, мобильных пользовательских устройств и т. д. В таких системах передатчик, как правило, характеризуется малыми вычислительными мощностями и ограниченной емкостью аккумулятора устройства. При этом существенные ограничения накладываются как на мощность процессора, так и на объем памяти.

Существующие технологии сжатия видеоданных, в первую очередь подходы, описанные в стандартах серий ITU-T H.26x и ISO/IEC MPEG, не учитывают специфику источника информации в подобных системах, поэтому разработка новых методов сжатия визуальных данных, не требующих больших вычислительных затрат на стороне передатчика, является важной и актуальной задачей.

В диссертационной работе рассматривается распределенное кодирование видеоданных – подход к сжатию, основанный на методах кодирования зависимых источников, позволяющий существенно снизить сложность обработки на стороне передатчика. Решается **задача** эффективного восстановления промежуточных кадров на стороне декодера.

Различные аспекты кодирования зависимых источников (или распределенного кодирования источников) представлены в работах известных отечественных и зарубежных авторов (С.И.Гельфанд, В.Д.Колесник, Б.Д.Кудряшов, М.С.Пинскер, Г.Ш.Полтырев, А.Вайнер, Д.Вулф, Д.Слепьян). Однако, до недавнего времени практическая реализация этих идей не была востребована. Только в конце 1990-х годов появились прикладные задачи, в которых применение распределенного кодирования могло дать преимущества по сравнению с существующими на тот момент подходами. В качестве перспективной области рассматривалось сжатие видеоданных в системах с ограниченными вычислительными ресурсами на стороне передатчика информации. Одним из существенных преимуществ от применения распределенного кодирования в данной системе, является то, что процедуру оценки и устранения временной избыточности, основанную на пред-

сказании промежуточных кадров, можно выполнять на приемнике, за счет чего существенно снижается сложность обработки на передатчике. Процедуру межкадрового предсказания в таких системах принято называть *генерацией дополнительной информации*. Кроме того, т. к. задача восстановления промежуточных кадров решается декодером, повышение степени сжатия возможно только за счет модификации приемника. В рамках данного класса прикладных задач были разработаны концепции кодеков, основанные на принципах распределенного кодирования источников. В последнее время большое внимание уделяется расширению и модификации этих концепций с учетом появляющихся новых прикладных задач, таких как эффективное кодирование многомерных и многоакурсных видеопоследовательностей. Несмотря на это, в базовых принципах остается ряд открытых вопросов. К их числу следует отнести учет особенностей входных данных для процедуры межкадрового предсказания на стороне декодера, а также оценку параметров ошибок межкадрового предсказания. Также следует отметить, что в большинстве современных работ приводится описание эвристических подходов для решения практических задач, возникающих при распределенном кодировании визуальных данных. При этом теоретические модели данных исследованы не в полной мере, что не позволяет эффективно разрабатывать новые и улучшать существующие алгоритмы сжатия, использующие данный подход.

Целью диссертационного исследования является повышение степени сжатия без ухудшения качества восстановления видеоданных в кодеках, основанных на принципах кодирования зависимых источников с дополнительной информацией на декодере, за счет усовершенствования существующих и разработки новых способов обработки данных на стороне декодера.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать типовые методы сжатия видеоданных, основанные на принципах распределенного кодирования источников.
2. Исследовать способы формирования дополнительной информации на стороне декодера.
3. Предложить новый алгоритм генерации дополнительной информации, учитывающий особенности входных данных.
4. Исследовать статистические характеристики ошибок межкадрового предсказания, возникающих при генерации дополнительной информации.

5. Разработать модель ошибок предсказания промежуточных кадров на стороне декодера.
6. Предложить алгоритм оценки параметров ошибок межкадрового предсказания.

Объект и предмет исследования. Объектом исследования является система сжатия видеоданных, основанная на принципах кодирования зависимых источников с дополнительной информацией на декодере.

Предмет исследования составляет процесс восстановления промежуточных кадров на стороне декодера.

Методы исследования. При получении основных результатов работы использовались общие методы системного анализа, методы теории вероятностей и математической статистики, теории случайных процессов, в частности марковских случайных полей, теории принятия решений, методы машинного зрения, а также методы имитационного моделирования.

Научная новизна:

1. Построена модель системы сжатия на базе распределенного кодирования для анализа алгоритмов восстановления промежуточных кадров на стороне декодера, позволяющая в одинаковых условиях производить сравнение различных методов сжатия видеоданных.
2. Предложен алгоритм генерации дополнительной информации декодера, отличающийся от существующих алгоритмов тем, что использует метод оценки движения, основанный на модели истинного движения объектов в видеоданных.
3. Предложен способ сравнительной оценки различных алгоритмов межкадрового предсказания в системах распределенного кодирования видеоданных, отличающийся от существующих тем, что позволяет в одинаковых условиях производить сравнительную оценку методов предсказания, не учитывая влияние прочих методов системы сжатия.
4. Впервые предложена модель ошибок предсказания промежуточных кадров, в явном виде учитывающая пространственные особенности артефактов интерполяции.
5. Предложен алгоритм оценки параметров ошибок предсказания промежуточных кадров, отличающийся от существующих тем, что позволяет

в явном виде учитывать пространственную зависимость между ошибками.

Научная и практическая значимость диссертационной работы. Полученные в диссертационной работе результаты позволяют повысить по сравнению с существующими алгоритмами эффективность сжатия видеоданных в системах, основанных на распределенном кодировании источников, что способствует уменьшению энергопотребления и габаритных размеров кодеров видеoinформации.

Степень достоверности. Результаты, полученные в диссертационной работе, согласуются с известными теоретическими моделями распределенного кодирования. Основные результаты опубликованы в рецензируемых журналах и доложены на крупных международных конференциях.

Апробация работы. Основные результаты работы докладывались и обсуждались на следующих конференциях и симпозиумах в период с 2009 по 2014 гг.: на научных сессиях ГУАП; на 14-ой конференции «IEEE Multimedia Signal Processing»; на 6-ой конференции «International Congress on Ultra Modern Telecommunications and Control Systems»; на 13-м симпозиуме «Problems of Redundancy in Information and Control Systems»; на 16-ой конференции «IEEE Multimedia Signal Processing»; на 8-й международной конференции «KES Conference on Intelligent Interactive Multimedia: Systems And Services».

Внедрение результатов. Результаты работы были использованы в рамках проекта «Разработка цепочки фильтров постобработки видеоданных», осуществляемого ЗАО «Интел А/О». Кроме того, результаты работы используются в учебном процессе кафедры инфокоммуникационных систем СПбГУАП.

Личный вклад. Все результаты, представленные в тексте диссертационной работы, получены автором лично.

Публикации. Материалы, отражающие основное содержание и результаты диссертационной работы, опубликованы в 15 печатных работах. Из них 2 работы опубликованы в рецензируемых научных журналах, утвержденных в перечне ВАК, и 5 работ опубликованы в журналах, индексируемых в Scopus.

Основные положения, выносимые на защиту:

1. Алгоритм межкадрового предсказания для кодеков видеoinформации, основанных на принципах кодирования зависимых источников с дополнительной информацией на декодере, позволяющий уменьшить по срав-

нению с существующими алгоритмами число ошибок предсказания за счет использования временной интерполяции с учетом истинного движения объектов.

2. Модель кодека без обратной связи, позволяющая производить сравнение алгоритмов межкадрового предсказания в системах кодирования зависимых источников видеoinформации с дополнительной информацией на декодере.
3. Модифицированный алгоритм оценки параметров ошибок межкадрового предсказания в спектральной области, который за счет учета неоднородности ошибок в спектральных коэффициентах позволяет уменьшить битовые затраты на восстановление промежуточного кадра.
4. Модель ошибок межкадрового предсказания, основанная на Марковских случайных полях, которая позволяет учитывать пространственную зависимость между ошибками в спектральных коэффициентах.

Объем и структура работы. Диссертация состоит из введения, четырёх разделов, заключения и двух приложений. Полный объём диссертации составляет 142 страницы с 28 рисунками и 7 таблицами. Список литературы содержит 66 наименований.

1 СЖАТИЕ ВИДЕОДАНЫХ С ИСПОЛЬЗОВАНИЕМ ПРИНЦИПОВ КОДИРОВАНИЯ ЗАВИСИМЫХ ИСТОЧНИКОВ

1.1 Вводные замечания

В течение последних десятилетий наблюдается существенное развитие технологий беспроводного обмена информацией. Пропускная способность существующих беспроводных сетей передачи данных позволяет осуществлять передачу по ним таких объемных данных как видеоданные. Следует отметить, что все чаще в современных сетях передачи данных в качестве источников такой информации выступают мобильные устройства, которые, как правило, характеризуются малыми вычислительными возможностями (при этом ограничения накладываются как на вычислительную мощность процессора, так и на доступный объем памяти) и ограниченным, зачастую трудно восполняемым, запасом аккумуляторной батареи. Существующие технологии сжатия видеоданных, в первую очередь подходы, описанные в стандартах серий ITU-T H.26x и ISO/IEC MPEG [1], характеризуются наличием «сложного» кодера и «простого» декодера, и плохо подходят для использования на мобильных устройствах. В качестве перспективной технологии для использования на мобильных устройствах во многих работах рассматриваются подходы, связанные с *распределенным кодированием видеоданных* (Distributed Video Coding – DVC) [2].

Большинство современных видеокодеков используют следующую схему устранения временной избыточности: сжатию подвергаются не данные исходного кадра, а ошибки предсказания. Предсказание пикселей текущего кадра осуществляется по данным одного или нескольких ранее обработанных и уже восстановленных (декодированных) кадров. Такие кадры называют *опорными* или *базовыми*. Процедура поиска в опорных кадрах пикселей, которые будут использоваться для предсказания, называется *оценкой движения* (Motion Estimation – ME), причем эта процедура может занимать до 70% от общего времени кодирования [3].

В схемах на основе DVC устранение временной избыточности также осуществляется с помощью предсказания, однако оно выполняется на стороне декодера, что позволяет существенно снизить сложность кодирования. С точки зрения распределенного кодирования результат предсказания на стороне декодера

является *дополнительной информацией*, а саму процедуру предсказания называют *генерацией дополнительной информации*. Кодеру остается только сформировать уточняющую информацию, которая позволит декодеру исправить ошибки предсказания. Существующие реализации DVC используют для исправления ошибок предсказания методы помехоустойчивого кодирования. Наиболее популярным подходом является интерпретация предсказания с помощью так называемого *виртуального канала* (Virtual Dependency Channel), по которому «передаются» исходные данные x , а на выходе наблюдается дополнительная информация (результаты предсказания) y . Для того чтобы исправить возникшие «ошибки», кодер интерпретирует x как информационную последовательность и формирует по ней проверочные символы, используя заранее определенный код. Декодер конкатенирует полученные проверочные символы с y , после чего применяет процедуру декодирования, восстанавливая в y несовпадающие с x позиции. Таким образом, сжатие достигается в том случае, если количество проверочных символов меньше, чем количество исходных данных x . Следует отметить, что описанный подход является только одним из способов практической реализации идеи DVC. Альтернативные способы будут рассмотрены в данном разделе далее.

В данной работе рассматривается процесс сжатия видеопоследовательностей с использованием методов распределенного кодирования источников. Для краткости изложения будем далее в тексте называть кодеки видеоданных, основанные на распределенном кодировании источников, *распределенными кодеками видеоданных* или, если это не будет вызывать неоднозначности, просто *распределенными кодеками*.

Данный раздел организован следующим образом. В начале приводится краткое введение в теорию распределенного кодирования источников. Затем вводятся основные понятия, характерные для такой системы обработки информации, и дается ряд иллюстративных примеров для пояснения ключевых идей, используемых при сжатии данных в подобных системах. Далее вводится классификация методов распределенного кодирования видеоданных и рассматриваются две базовые концепции, лежащие в основе существующих алгоритмов сжатия с использованием данного подхода. После этого вводится в рассмотрение модель распределенного кодирования на базе проекта DISCOVER, рассматриваются способы практической реализации основных модулей системы сжатия, использующей эту

модель. Описание распределенного кодирования заканчивается перечислением основных прикладных задач, в которых подобный подход может использоваться.

Перед тем как переходить к дальнейшему изложению, приведем краткое описание используемых в данной работе обозначений.

- Скаляры – символы, набранные в нижнем регистре, например p .
- Векторы – символы, набранные в нижнем регистре с полужирным начертанием, например $\mathbf{a} = (a_1, a_2, \dots, a_n)$ задает вектор в n -мерном пространстве. Для определенности будем полагать, что векторы представляют собой векторы-столбцы. В таком случае вектор-строку будем определять с использованием операции транспонирования, например \mathbf{a}^T .
- Матрицы – символы, набранные в верхнем регистре с полужирным начертанием, например \mathbf{A} . Элемент, находящийся матрице \mathbf{A} в строке с номером k и столбце с номером l будем обозначать в зависимости от контекста либо через $\mathbf{A}[k,l]$, либо через $a_{k,l}$.
- Случайные величины – символы, набранные в верхнем регистре, например X .
- Множества – символы, набранные в верхнем регистре курсивом, например \mathcal{X} . Там, где это возможно, будем задавать множества перечислением их элементов или указанием условия, которому удовлетворяют все элементы, например:
 - $\mathcal{X} = \{1, 2, \dots, q\}^m$ задает множество q -ичных векторов в m -мерном пространстве;
 - $\mathcal{Y} = \{(i, j) | i \in \mathcal{R}^2, i^2 + j^2 < r^2\}$ содержит множество всех точек вещественной плоскости, лежащих внутри окружности радиуса r .

Для того, чтобы избежать неоднозначности, при использовании верхних индексов будут использоваться скобки, например $a^{(i)}$. Данная система обозначений будет применяться далее в течение всего изложения.

Материал раздела изложен в работах [4], [5].

1.2 Теоретические предпосылки распределенного видеокодирования

Методы распределенного кодирования видеоданных основаны на задаче кодирования зависимых источников без памяти. Эта задача, также известная как *распределенное кодирование зависимых источников* (Distributed Source Coding), была впервые сформулирована и решена в работе [6], где была рассмотрена система передачи информации (рисунок 1.1) с парой зависимых дискретных источников без памяти U_X и U_Y , на выходе которых наблюдаются ансамбли сообщений из \mathcal{X} и \mathcal{Y} соответственно, причем

$$p(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^n p(x_i, y_i),$$

где $\mathbf{x} \in \mathcal{X}^n$, $\mathbf{y} \in \mathcal{Y}^n$.

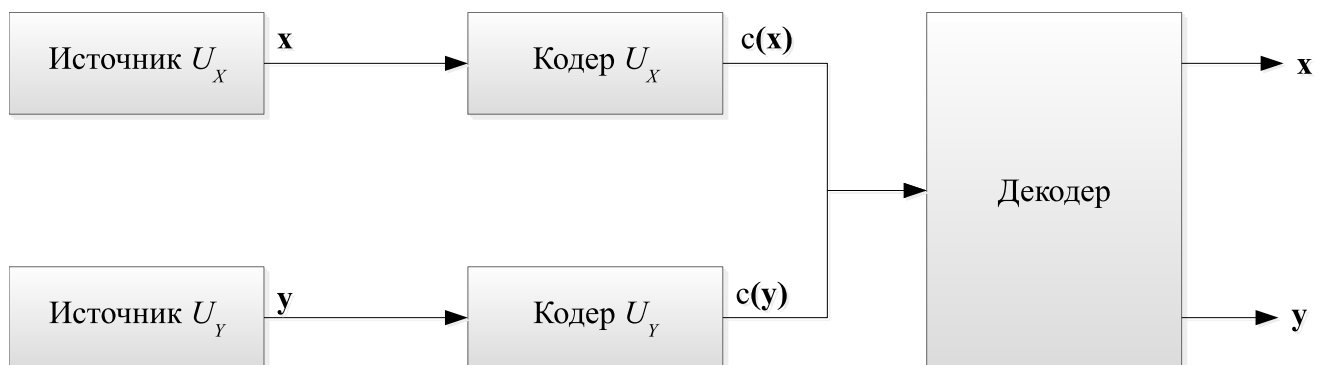


Рисунок 1.1 — Структура системы обработки информации с использованием независимого кодирования зависимых источников

Одним из ключевых результатов работы [6] является определение области допустимых скоростей для случая независимого кодирования и совместного декодирования источников. В такой системе обработки информации кодирование выходов каждого из источников выполняется независимо от другого своим собственным кодером. Кодер источника U_X отображает последовательность символов \mathbf{x} в двоичную последовательность $c(\mathbf{x})$, кодер U_Y отображает \mathbf{y} в $c(\mathbf{y})$. Обозначим через $H(X, Y)$, $H(X|Y)$, $H(Y|X)$ энтропии, определяемые из совместного распределения $p(\mathbf{x}, \mathbf{y})$. Любая пара чисел (r_X, r_Y) , удовлетворяющая следующим

соотношениям:

$$\begin{aligned} r_X &\geq H(X|Y) \\ r_Y &\geq H(Y|X), \\ r_X + r_Y &\geq H(X,Y) \end{aligned}$$

является парой допустимых скоростей при независимом кодировании и совместном декодировании зависимых источников. Этот регион скоростей представлен графически на рисунке 1.2. Приведенное выше утверждение известно как прямая теорема кодирования зависимых источников. Доказательство можно найти, например, в работе [6] или в учебнике [7].

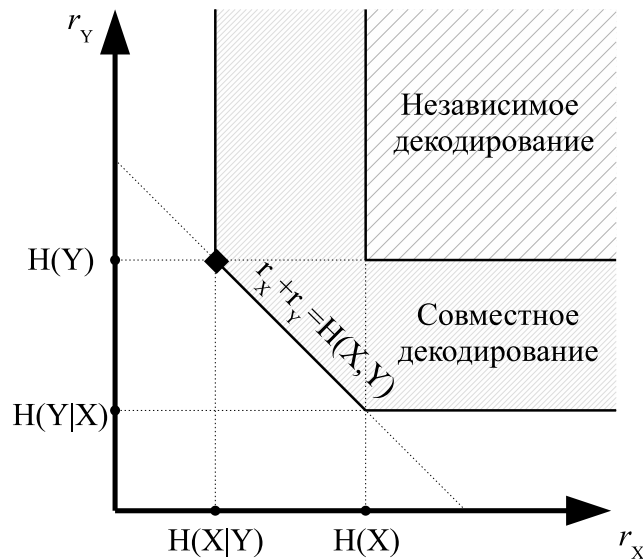


Рисунок 1.2 — Область допустимых скоростей при кодировании зависимых источников

Основной результат данной теоремы заключается в том, что независимое кодирование зависимых источников без памяти не приводит в общем к увеличению минимальной допустимой суммарной скорости кодирования по сравнению с совместным кодированием.

Процесс сжатия в описанной выше системе обработки информации принято называть кодированием Слепяна-Вулфа (Slepian-Wolf coding). Отметим, что кодирование Слепяна-Вулфа является сжатием без потерь, т. к. в процессе совместного декодирования обе последовательности восстанавливаются с произвольно малой вероятностью ошибки.

В 1976 году Вайнер и Зив [8] рассмотрели частный случай кодирования зависимых источников, когда требуется сжать с потерями выход источника U_X , если на декодере доступен без потерь выход U_Y (рисунок 1.3). В таком случае, говорят, что y – это дополнительная информация декодера, а процесс сжатия в такой системе называют кодированием с дополнительной информацией на декодере. В англоязычной литературе для обозначения дополнительной информации используют термин *Side Information* (сторонняя информация). Графическое представление такой системы передачи информации приведено на рисунке 1.3. Вайнер и Зив исследовали поведение функции «скорость-искажение» [9] и показали, что в общем случае эффективность сжатия в такой системе ниже, чем в схеме с совместным кодированием. Однако, в некоторых случаях, в частности, когда источники U_X и U_Y являются совместно Гауссовыми, а в качестве меры потерь используется функция среднеквадратичной ошибки, кривые «скорость-искажение» совпадают. Этот результат известен как теорема Вайнера-Зива, доказательство которой можно найти в работе [8].

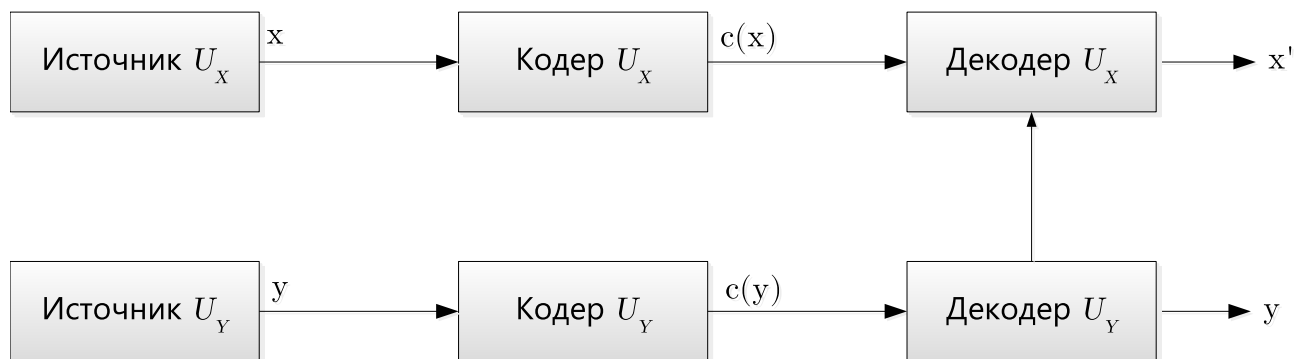


Рисунок 1.3 — Структура системы передачи информации при кодировании зависимых источников с дополнительной информацией на декодере

Теоретические результаты, приведенные в работах Слепяна-Вулфа и Вайнера-Зива, предполагают, что возможно сжать выходы двух зависимых источников с использованием методов распределенного кодирования (независимое кодирование и совместное декодирование), при этом не потеряв в эффективности сжатия по сравнению со схемами, основанными на совместном кодировании и декодировании. В работе [10] утверждается, например, что система с кодированием при наличии дополнительной информации на декодере не проигрывает

совместному кодированию, в том случае, если разница между выходами источников является Гауссовой.

1.3 Применение помехоустойчивого кодирования для сжатия информации

Практические подходы к распределенному кодированию зависимых источников во многом связаны с методами помехоустойчивого кодирования. В связи с этим перед описанием применения распределенного кодирования для сжатия видеоданных рассмотрим несколько простых примеров, показывающих реализацию распределенного кодирования на практике с использованием методов помехоустойчивого кодирования.

Так как символы на выходе источников U_X и U_Y обладают взаимной корреляцией, то вектор \mathbf{y} можно рассматривать как искаженную версию вектора \mathbf{x} . В таком случае, процесс обработки данных в распределенном кодировании можно интерпретировать следующим образом (рисунок 1.4). Последовательность \mathbf{x} передается через некоторый канал с шумом. В результате на декодер поступает новая последовательность \mathbf{y} , причем

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

где вектор \mathbf{n} описывает шум (различия между \mathbf{y} и \mathbf{x}).

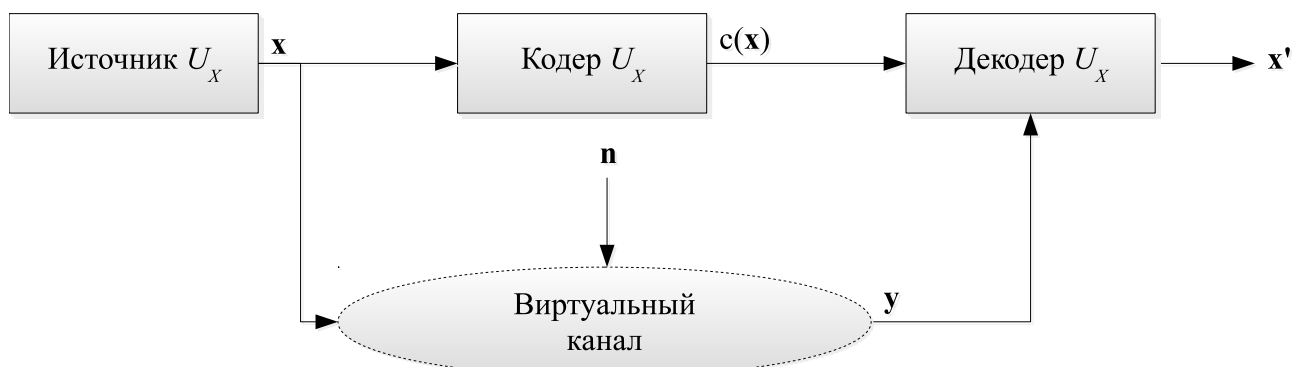


Рисунок 1.4 — Альтернативная интерпретация кодирования с дополнительной информацией на декодере

Задача совместного декодера сводится к тому, чтобы устранить шумовые составляющие из \mathbf{y} и восстановить \mathbf{x} , т. е. исправить ошибки, «возникшие» в процессе передачи по каналу. В распределенном кодировании принято называть описанный выше канал *каналом с виртуальной зависимостью* или *виртуальным каналом*, а «возникающий» в нем шум – *корреляционным шумом* (Correlation Noise). Задача кодера источника U_X в таком случае заключается в том, чтобы выполнить помехоустойчивое кодирование последовательности \mathbf{x} и отослать совместному декодеру проверочную информацию. При этом считается, что проверочная информация передается без ошибок. Сжатие достигается в том случае, если средний объем проверочной информации, достаточной для восстановления исходной последовательности \mathbf{x} , меньше объема исходной последовательности.

В качестве демонстрации применения методов помехоустойчивого кодирования в задаче сжатия зависимых источников, а также для того, чтобы представить реализацию схемы кодирования Слепяна-Вулфа на практике, рассмотрим следующий пример [11]. Пусть на выходе источников U_X и U_Y наблюдаются равновероятные последовательности $\mathbf{x}_i \in \mathcal{X}$ и $\mathbf{y}_i \in \mathcal{Y}$ двоичных символов длины $m = 3$, т. е. $\mathcal{X} = \mathcal{Y} = \{0,1\}^3$. Будем считать, что выходы источников являются зависимыми, причем зависимость выражается в том, что расстояние Хэмминга между каждой парой \mathbf{x}_i и \mathbf{y}_i не превышает единицу, т. е. последовательности различаются не более, чем в одной позиции. В качестве примера рассмотрим передачу сообщений, состоящих из одного символа ($n = 1$): $\mathbf{x}_1 = (101)$, $\mathbf{y}_1 = (100)$. Для упрощения обозначений далее не будем указывать индекс 1. Рассмотрим три схемы кодирования и определим для каждой из них минимальную скорость (минимальное число бит, которое кодер должен передать декодеру), необходимую для точного восстановления выходов обоих источников на декодере.

1. *Независимое кодирование и декодирование* (рисунок 1.5). Выходы источников U_X и U_Y кодируются и декодируются независимо друг от друга. В таком случае, так как последовательности на выходе источников являются равновероятными, суммарную минимальную скорость можно опреде-

лечь следующим образом:

$$\begin{aligned}
 r_{XY}^{(1)} &= r_X + r_Y = \\
 &= - \sum_{\mathbf{x} \in \{0,1\}^3} p(\mathbf{x}) \log_2(p(\mathbf{x})) - \sum_{\mathbf{y} \in \{0,1\}^3} p(\mathbf{y}) \log_2(p(\mathbf{y})) = \\
 &= - \sum_{\mathbf{x} \in \{0,1\}^3} \frac{1}{8} \log_2 \frac{1}{8} - \sum_{\mathbf{y} \in \{0,1\}^3} \frac{1}{8} \log_2 \frac{1}{8} = 3 + 3 = 6 \text{ бит}
 \end{aligned}$$

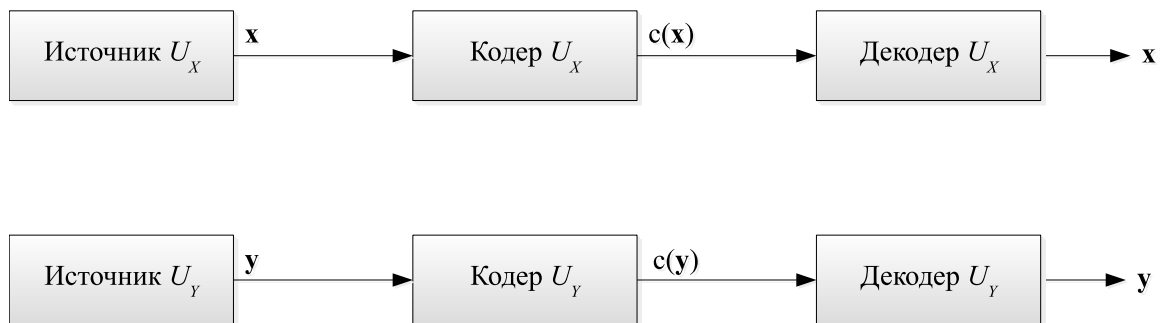


Рисунок 1.5 — Система передачи информации с независимой обработкой

Таким образом, при независимом кодировании источников U_X и U_Y необходимо передать 6 бит, для того, чтобы декодер смог восстановить последовательности без потерь. Для данного примера кодер должен отправить декодеру вектор $\mathbf{v}^{(1)} = (101100)$, который получается простой конкатенацией векторов \mathbf{x} и \mathbf{y} .

2. Совместное кодирование и декодирование (рисунок 1.6).

Выходы источников кодируются совместно, декодирование тоже происходит совместно. Так как расстояние Хэмминга между \mathbf{x} и \mathbf{y} не превышает единицу, вектор-разница \mathbf{z} между этими последовательности принадлежит упорядоченному множеству $\mathcal{Z} = \{000, 001, 010, 100\}$, причем элементы этого множества равновероятны. Если последовательность \mathbf{y} известна и кодеру, и декодеру, то для передачи \mathbf{x} достаточно закодировать только индекс вектора-разницы. Минимальная суммарная скорость

в такой системе определяется как:

$$\begin{aligned}
 r_{XY}^{(2)} &= r_Z + r_Y = \\
 &= - \sum_{z \in Z} p(z) \log_2(p(z)) - \sum_{y \in \{0,1\}^3} p(y) \log_2(p(y)) = \\
 &= - \sum_{z \in Z} \frac{1}{4} \log_2 \frac{1}{4} - \sum_{y \in \{0,1\}^3} \frac{1}{8} \log_2 \frac{1}{8} = 2 + 3 = 5 \text{ бит.}
 \end{aligned}$$

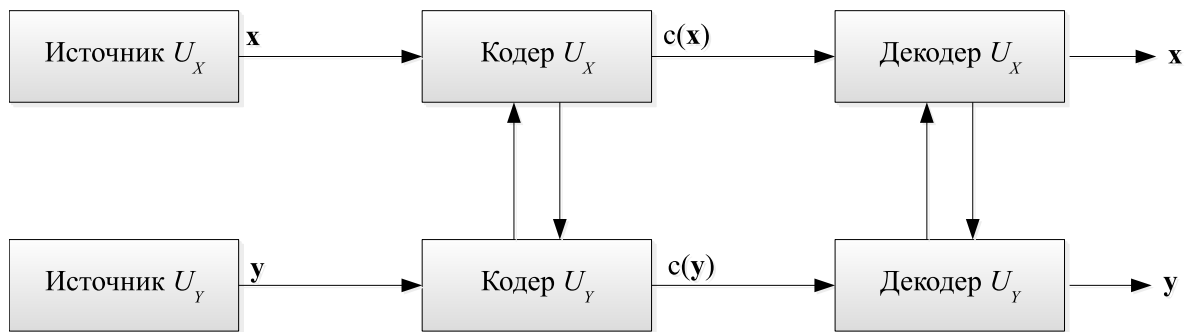


Рисунок 1.6 — Система передачи информации с совместной обработкой

Таким образом, при совместном кодировании источников U_X и U_Y декодеру необходимо передать 5 бит, для того, чтобы он смог восстановить последовательности без потерь. При этом считается, что множество Z известно кодери и декодеру.

В данном примере кодер должен отправить вектор $\mathbf{v}^{(2)} = (10001)$, в котором первые три бита (100) представляют собой последовательность y , а последние два бита (01) являются двоичным представлением индекса вектора в множестве Z , который в данном примере равен

$$\mathbf{z} = \mathbf{x} + \mathbf{y} = (101) + (100) = (001).$$

Декодер, получив последовательность $\mathbf{v}^{(2)}$, выделяет из неё сообщение y и индекс элемента в Z . Далее соответствующий элемент Z складывается с y для того, чтобы получить x :

$$\mathbf{y} + Z(1) = (100) + (001) = (101) = \mathbf{x}.$$

Таким образом, декодер безошибочно восстанавливает оба сообщения.

3. *Независимое кодирование и совместное декодирование* (рисунок 1.7).

В такой системе кодирование выходов источников осуществляется независимо, декодер сначала восстанавливает последовательность y и, зная y , восстанавливает x . Определим, сколько бит должен передать кодер декодеру, чтобы тот смог точно восстановить x .

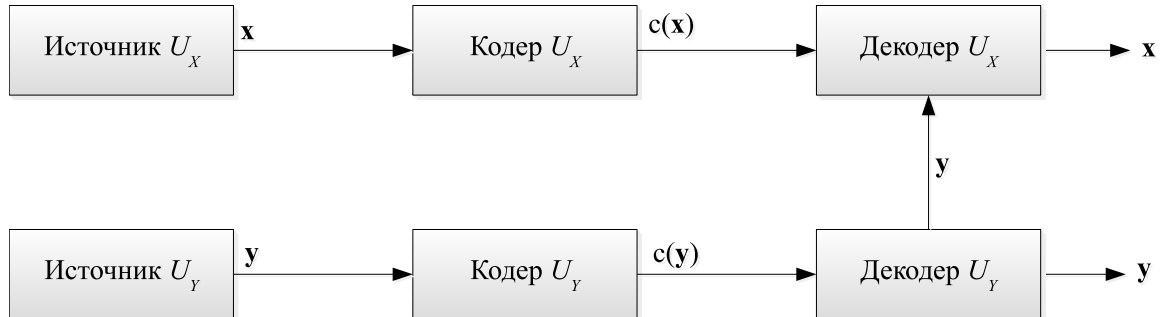


Рисунок 1.7 — Система передачи с обработкой информации по схеме Вайнера-Зива

Введем в рассмотрение следующую проверочную матрицу линейного кода:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Будем считать, что эта матрица известна кодеру и декодеру. Кодер рассчитывает синдром последовательности \mathbf{x} :

$$\mathbf{s} = \mathbf{x}^T \mathbf{H}^T,$$

где $\mathbf{s} \in \mathcal{S}$ и $\mathcal{Z} = \{0,1\}^2$ – множество всех возможных синдромов для рассмотренного линейного кода.

Далее синдром \mathbf{s} пересылается декодеру, который определяет смежный класс для вектора полученного синдрома \mathbf{s} по таблице стандартной расстановки линейного кода, которая известна для матрицы \mathbf{H} и строится на декодере заранее. Декодер находит лидера смежного класса, как вектор, ближайший по расстоянию Хэмминга к уже восстановленному вектору y . Лидер смежного класса и есть восстановленный вектор x . Определим минимальное количество бит, необходимых для точного восстановления

обеих последовательностей:

$$\begin{aligned}
 r_{XY}^{(3)} &= r_S + r_Y = \\
 &= - \sum_{\mathbf{s} \in S} p(\mathbf{s}) \log_2(p(\mathbf{s})) - \sum_{\mathbf{y} \in \{0,1\}^3} p(\mathbf{y}) \log_2(p(\mathbf{y})) = \\
 &= - \sum_{\mathbf{s} \in S} \frac{1}{4} \log_2 \frac{1}{4} - \sum_{\mathbf{y} \in \{0,1\}^3} \frac{1}{8} \log_2 \frac{1}{8} = 2 + 3 = 5 \text{ бит}
 \end{aligned}$$

В рассматриваемом примере синдром последовательности \mathbf{x} , рассчитанный кодером:

$$\mathbf{s} = \mathbf{x}^T \mathbf{H}^T = (101) \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (10)$$

Биты синдрома передаются декодеру. Таблица стандартной расстановки для рассматриваемого кода представлена в таблице 1.1.

Таблица 1.1

Таблица стандартной расстановки

Синдром	Смежный класс
(00)	{(000),(111)}
(01)	{(001),(011)}
(10)	{(010),(101)}
(11)	{(100),(011)}

Смежный класс для полученного синдрома $\{(010),(101)\}$. Найдем лидера смежного класса, зная, что $\mathbf{y} = (100)$. Найдем расстояние Хэмминга между $\mathbf{y} = (100)$ и векторами из смежного класса:

$$(100) + (010) = 2$$

$$(100) + (101) = 1$$

Таким образом, лидер смежного класса – это вектор (101). Отметим, что этот вектор совпадает с последовательностью \mathbf{x} , т. е. удалось успешно выполнить декодирование. При этом кодер отправил декодеру 5 бит, что

совпадает со скоростью, полученной в случае совместных кодирования и декодирования, т. е. $r_{XY}^{(3)} = r_{XY}^{(2)} < r_{XY}^{(1)}$.

Приведенные выше примеры являются демонстрацией практического применения методов распределенного кодирования для сжатия двух простых зависимых источников, на выходе которых наблюдаются одномерные векторы с коррелированными символами. Однако отметим, что видеопоследовательности представляют собой многомерные (например, два измерения на представление одной сцены и третье измерение на представление изменений в сцене во времени) данные, в связи с чем далее рассмотрим применение распределенного кодирования для сжатия видеоданных.

1.4 Пример реализации распределенного кодирования видеоисточников в рамках простой модели видеопотока

В начале 2000-х годов на основании теорем Вайнера-Зива и Слепяна-Вулфа был сформирован новый подход к сжатию видеоданных, известный как *распределенное кодирование видеоданных* (Distributed Video Coding, DVC, или Wyner-Ziv Video Coding, WZVC) [12]. В этом подходе сжимаемая видеопоследовательность интерпретируется как выход от нескольких зависимых источников, при этом источники могут порождать как кадры целиком, так и части кадров. Наиболее распространенной схемой является схема с двумя источниками, которые обрабатываются по следующим правилам:

- выход первого источника обрабатывается в независимом режиме (по аналогии с источником U_Y , рассмотренным ранее);
- выход второго источника кодируется независимо, но декодирование осуществляется с учетом восстановленных данных первого источника (по аналогии с источником U_X , рассмотренным ранее).

Перед тем, как рассматривать практические реализации распределенных кодеков видеоданных, приведем ещё один пример. Данный пример не демонстрирует новых принципов сжатия, однако позволяет получить некоторые оценки на простой модели видеоданных, как последовательности двумерных данных (кад-

ров), имеющих зависимость во времени, а также улучшить качественное понимание системы в целом.

Рассмотрим видеопоследовательность, состоящую из трех черно-белых кадров размером 5 пикселей в высоту и 5 пикселей в ширину. Единственным движущимся объектом на кадрах является черная точка (один пиксель) на белом фоне, начальное положение точки фиксировано и находится в центре кадра. Между соседними кадрами точка должна изменить свое местоположение, перейдя в один из 8 соседних пикселей (рисунок 1.8). Будем считать, что вероятности переходов равны.

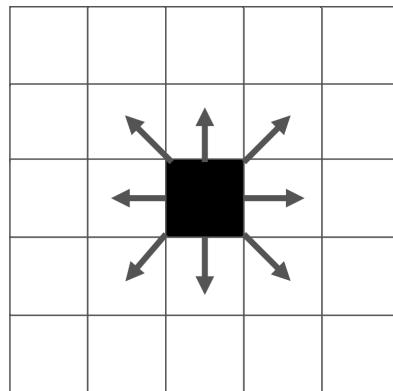


Рисунок 1.8 — Возможные направления смещения точки между соседними кадрами

Для того, чтобы сжать такую видеопоследовательность достаточно передать информацию о траектории движения точки. Рассмотрим два возможных сценария кодирования информации о траектории. Обобщенная схема системы передачи для обоих сценариев приведена на рисунке 1.9. Данная система включает два источника информации U_Y и U_X . Будем считать, что видеопоследовательность была заранее разбита на две подпоследовательности таким образом, что на выходе источника U_Y появляются кадры с номерами 1 и 3, а на выходе источника U_X – кадр 2.

Также в системе передачи информации присутствует ключ, определяющий режим работы кодеров источников. Если ключ разомкнут ($K = 0$), кодеры не взаимодействуют друг с другом, т. е. этот сценарий соответствует случаю независимого кодирования и совместного декодирования выходов зависимых источников U_Y и U_X . Если ключ замкнут ($K = 1$), то сжатие информации, поступающей от ис-

точников, выполняется совместно, т. е. этот сценарий соответствует совместному кодированию и декодированию.

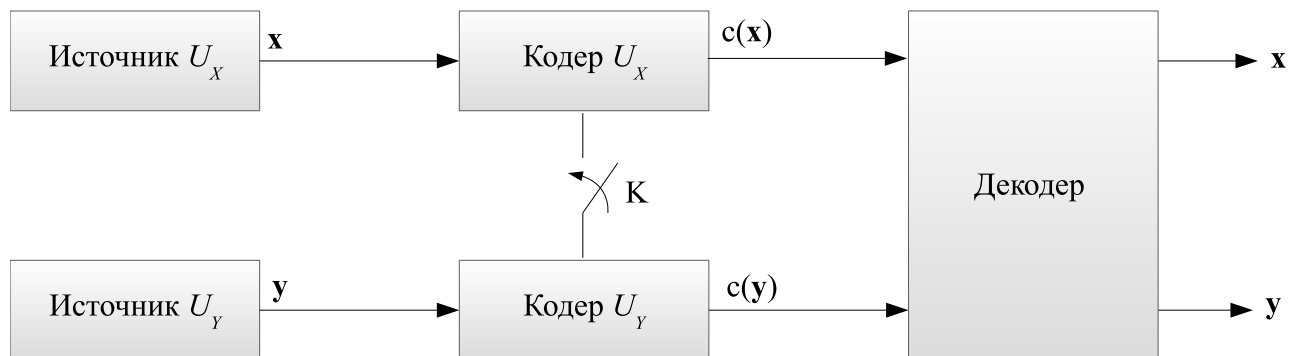


Рисунок 1.9 — Структурная схема обобщенной системы передачи информации от зависимых источников с общим декодером

1. Определим сколько бит нужно для кодирования всех кадров, когда ключ K замкнут. Пусть величина $r_{2|1}$ показывает сколько бит необходимо передать декодеру, чтобы он смог восстановить положение черной точки на втором кадре. Т.к. есть 8 возможных равновероятных сдвигов, то число бит, которое необходимо затратить в таком случае равно $\log_2 8 = 3$ бита. Аналогичные рассуждения позволяют рассчитать значение $r_{3|2}$. В результате:

$$r_{K=0} = r_1 + r_{2|1} + r_{3|2} = 0 + \log_2 8 + \log_2 8 = 6 \text{ бит.}$$

2. Определим сколько бит нужно для кодирования всех кадров, когда ключ K разомкнут. Опишем возможный способ сжатия видеопоследовательности в такой системе передачи информации.
 - (а) Кодер источника U_Y передает сжатую информацию о кадрах 1 и 3 совместному декодеру.
 - (б) Декодер по обратной связи передает кодеру U_X информацию о положении черного пикселя в кадре 3.
 - (в) Кодер источника U_X отправляет совместному декодеру некоторое количество вспомогательных бит, которые позволяют ему восстановить координаты черного пикселя на кадре 2.

Найдем среднее число бит для такого способа сжатия:

$$r_{K=1}^{(1)} = r_Y + r_X,$$

где r_Y - битовые затраты на передачу информации о положении точки в кадрах 1 и 3, r_X - битовые затраты на передачу информации о положении точки в кадре 2.

$$r_Y = r_1 + r_{3|1},$$

где $r_{3|1}$ - количество бит, необходимых для передачи информации о положении пикселя в кадре 3 при известном кадре 1. Вероятности появления пикселя в определенных позициях кадра 3 приведены в следующей матрице:

$$\mathbf{P}_{3|1} = \frac{1}{64} \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 2 & 4 & 2 & 2 \\ 3 & 4 & 8 & 4 & 3 \\ 2 & 2 & 4 & 2 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix}.$$

Таким образом,

$$r_Y = 0 + \sum_{i=1}^5 \sum_{j=1}^5 \mathbf{P}_{3|1}[i,j] \log_2 \mathbf{P}_{3|1}[i,j] = 4.4528 \text{ бит.}$$

Определим теперь среднее число бит, которые кодер источника U_X должен отправить совместному декодеру, зная положение черного пикселя в кадре 3. Это число определяется координатами пикселя в кадре 3 и количеством траекторий, ведущих из центра (положения пикселя в кадре 0) в эти координаты. Количество возможных траекторий приведено в матрице \mathbf{L} :

$$\mathbf{L} = \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 2 & 4 & 2 & 2 \\ 3 & 4 & 8 & 4 & 3 \\ 2 & 2 & 4 & 2 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix}.$$

Среднее число бит для последнего шага можно найти по следующей формуле:

$$r_X = \sum_{i=1}^5 \sum_{j=1}^5 \mathbf{P}_{3|1}[i,j] \log_2 \mathbf{L}[i,j] = 1.5472 \text{ бит.}$$

В таком случае среднее число бит, затрачиваемых на сжатие всей последовательности:

$$r_{K=1}^{(1)} = r_Y + r_X = 4.4528 + 1.5472 = 6 \text{ бит.}$$

Рассмотрим альтернативный способ сжатия видеопоследовательности.

- (a) Кодер источника U_Y передает сжатую информацию о кадрах 1 и 3 совместному декодеру.
- (b) Кодер источника U_X передает декодеру контрольную информацию, например CRC [13], рассчитанную для кадра 2.
- (c) Декодер строит некоторую оценку положения черной точки в кадре 2, считает контрольную информацию по этой оценке и сравнивает с данными, полученными от кодера. Если контрольная информация совпадает, то декодер считает, что ему удалось декодировать кадр 2. В противном случае декодер запрашивает дополнительную информацию у кодера.
- (d) При наличии запроса от декодера кодер источника U_X отправляет декодеру некоторое количество вспомогательных бит, которые позволяют ему восстановить промежуточное положение черного пикселя на кадре 2.

Определим среднее количество бит, которые нужно передать для этой процедуры, без учета контрольной информации.

Шаг a) совпадает с тем же шагом описанного ранее алгоритма и требует 4.4528 бит. Шаг d) схож с шагом c) предыдущего алгоритма, за исключением того, что нужно учесть то, что декодер может иногда угадать положение точки в кадре 2. Таким образом, формула для расчета среднего числа бит для шага d) выглядит следующим образом:

$$r_d = \sum_{i=1}^5 \sum_{j=1}^5 \mathbf{P}_{3|1}[i,j] p_f(i,j) \log_2 \mathbf{L}[i,j],$$

где через $p_f(i,j)$ обозначена вероятность события, что декодер не угадал положение точки в кадре 2, при условии того, что в кадре 3 точка нахо-

дится в клетке с координатами (i, j) :

$$p_f(i, j) = \frac{\mathbf{L}[i, j] - 1}{\mathbf{L}[i, j]}.$$

В результате:

$$r_d = \sum_{i=1}^5 \sum_{j=1}^5 \mathbf{P}_{3|1}[i, j] \frac{\mathbf{L}[i, j] - 1}{\mathbf{L}[i, j]} \log_2 \mathbf{L}[i, j] = 1.0884 \text{ бит.}$$

Для рассмотренного алгоритма можно сделать два вывода.

- (а) Если для передачи CRC в среднем использовать 0.4558 бит, то описанная процедура не будет проигрывать кодированию с учетом зависимости между кадрами.
- (б) С вероятностью p_s декодер может угадать положение черного пикселя на кадре 1:

$$p_s = \sum_{i=1}^5 \sum_{j=1}^5 \mathbf{P}_{3|1}[i, j] \left(1 - \frac{\mathbf{L}[i, j] - 1}{\mathbf{L}[i, j]} \right) = 0.3906.$$

Рассмотрим теперь существующие кодеки, основанные на принципах кодирования зависимых источников.

1.5 Классификация методов распределенного кодирования источников видеoinформации

Для удобства дальнейшего изложения введем в рассмотрение классификацию методов распределенного кодирования (рисунок 1.10). Данная классификация будет использоваться при описании основных концепций распределенного видеокодирования. Методы распределенного кодирования источников видеoinформации в зависимости от ряда признаков можно классифицировать следующим образом.

- По типу обрабатываемых данных.

- *Обработка кадров целиком.* В методах распределенного кодирования, основанных на обработке кадров целиком, выходами зависимых источников считаются кадры видеопоследовательности. Такой подход является обоснованным с точки зрения теории кодирования зависимых источников в силу того, что близкие во времени кадры как правило обладают высокой временной зависимостью. Таким образом, в первую очередь осуществляется устранение временной избыточности.
- *Обработка частей кадров.* Распределенное кодирование в подобных методах используется для устранения как межкадровой, так и внутрикадровой избыточности. Кадры разбиваются на непересекающиеся блоки, и для каждого блока принимается решение о принадлежности к одному из зависимых источников. Тип кодирования определяется в зависимости от принятого решения.
- По наличию обратной связи от декодера к кодеру.
 - *С обратной связью.* В подобных методах между кодером и декодером есть обратная связь, по которой декодер может запрашивать некоторую вспомогательную информацию, например, с целью управления битовой скоростью.
 - *Без обратной связи.* В методах данного класса декодер не может запрашивать у кодера дополнительную информацию. Следует отметить, что в таком случае, если декодеру не удастся успешно декодировать данные зависимого источника, сжатие возможно только с потерями.
- По области обработки.
 - *Обработка в пространстве пикселей* (в пиксельном домене). В подобных методах считается, что зависимые источники порождают непосредственно пиксели изображений. Схемы, основанные на сжатии в пространстве пикселей, характеризуются, как правило, очень низкой вычислительной сложностью на стороне кодера, т. к. сжатию подвергаются непосредственно значения интенсивностей цветовых компонент кадров. Существенным недостатком подобных подходов является низкая гибкость, заключа-

- ющаяся в невозможности эффективно выделять и устранять пространственную избыточность, присущую кадрам видео.
- *Обработка в преобразованном пространстве* (в преобразованном домене). В методах, принадлежащих данному классу, считается, что зависимые источники порождают сообщения, которые представляют собой некоторые функции от пикселей кадров. Наиболее распространенным подходом является обработка спектральных коэффициентов, рассчитанных с использованием дискретного косинусного преобразования. Подобный подход позволяет ранжировать (классифицировать) выходы зависимых источников по степени важности с точки зрения визуального восприятия. Каждый подкласс далее может кодироваться независимо, что позволяет осуществлять эффективное управление битовой скоростью за счет масштабирования сжатого потока.

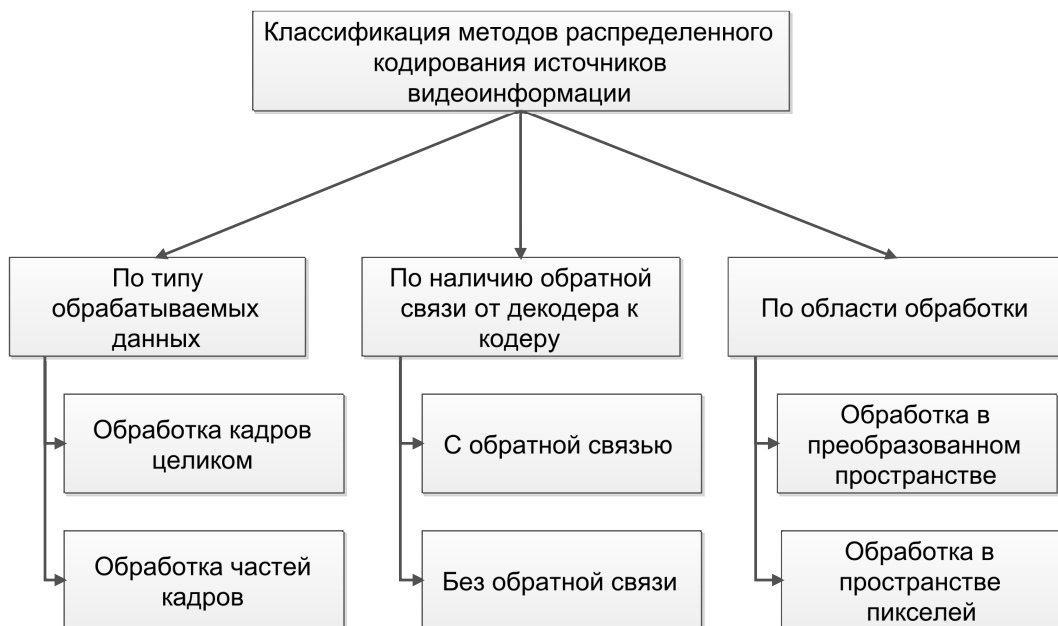


Рисунок 1.10 — Классификация методов распределенного кодирования видеоданных

1.6 Основные концепции распределенного кодирования источников видеoinформации

1.6.1 Концепция Стэнфорд

Концепция Стэнфорд была предложена в 2002 году для сжатия кадров в пиксельном домене [14], а затем была расширена для обработки спектральных коэффициентов [15]. В основе сжатия в схеме Стэнфорд лежит использование корреляции смежных кадров видеопоследовательности. С точки зрения классификации, приведенной в подразделе 1.5, данная концепция представляет собой метод сжатия кадров в преобразованном пространстве с использованием обратной связи. Схема распределенного кодека Стэнфорд приведена на рисунке 1.11. Приведем на качественном уровне описание работы кодека.

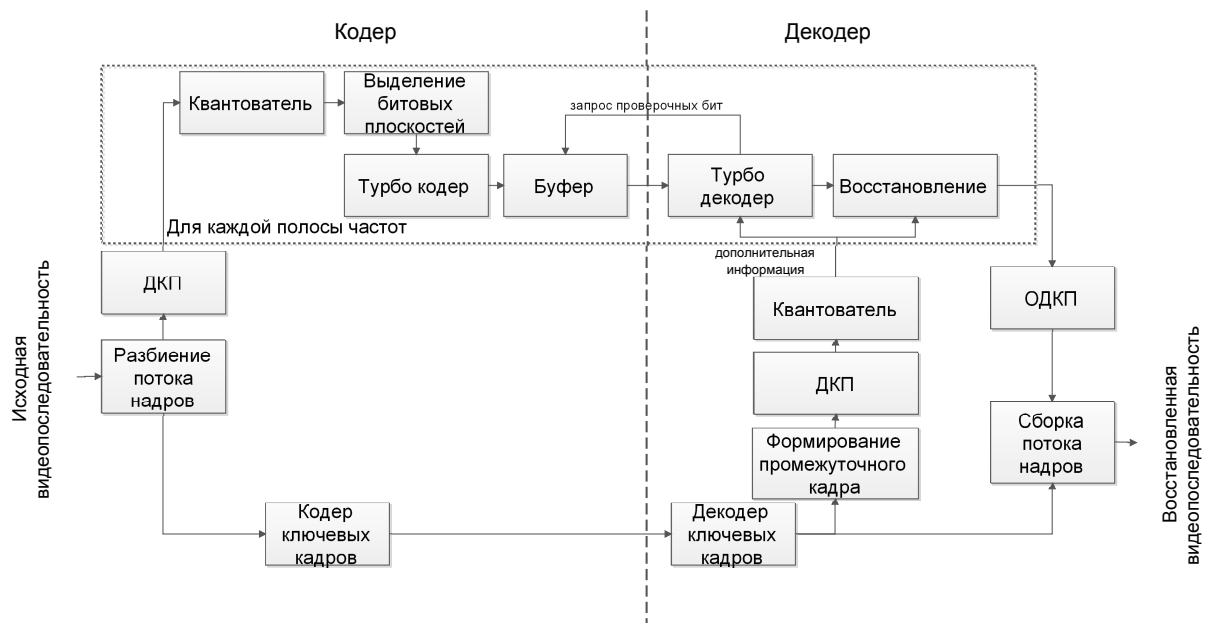


Рисунок 1.11 — Типовая схема кодека, основанного на концепции Стэнфорд

Кодер. Видеопоследовательность разбивается на промежуточные и базовые (ключевые) кадры в блоке классификации. Промежуточные кадры будем называть *ВЗ кадрами* (от Вайнер-Зив). Базовые кадры расположены в видеопоследовательности с некоторым интервалом, определяемым размером группы кадров (GOP, Group of Pictures). Ключевые кадры кодируются независимо, т. е. без устранения временной избыточности. Такой режим обработки будем называть режимом Intra.

К ВЗ кадрам применяется блочное спектральное преобразование, обычно дискретное косинусное преобразование. Спектральные коэффициенты со всего кадра затем группируются по следующему правилу: коэффициенты, находящиеся в разных блоках на одной и той же позиции попадают в одну группу.

Затем спектральные коэффициенты в каждой группе подвергаются равномерному квантованию, причем шаг квантования устанавливается таким, чтобы удовлетворять требуемому качеству (битовую скорость и качество восстановленного кадра с точки зрения критерия PSNR) сжатия. Квантованные спектральные коэффициенты в каждой группе далее разбиваются на битовые плоскости, которые независимо кодируются с использованием турбо кода. Кодирование начинается с наиболее значимой битовой плоскости. Проверочные биты турбо кода для каждой битовой плоскости сохраняются в промежуточном буфере, откуда могут отправляться частями декодеру при наличии запросов от декодера по обратной связи.

Декодер. Декодер, используя восстановленные ключевые кадры, осуществляет интерполяцию (или экстраполяцию) соответствующих ВЗ кадров. Будем называть кадры, полученные в результате данной операции *аппроксимирующими кадрами*. Аппроксимирующие кадры далее по аналогии с соответствующей операцией на кодере подвергаются дискретному косинусному преобразованию и квантованию, разбиваются на группы спектральных коэффициентов, из которых затем выделяются битовые плоскости. С точки зрения теории кодирования зависимых источников с дополнительной информацией на стороне декодера эти битовые плоскости как раз и формируют дополнительную информацию, которая используется при восстановлении спектральных коэффициентов ВЗ кадра. Процедура восстановления заключается в запросе по обратной связи проверочных бит из буфера кодера и исправлении ошибок в информационной части, сформированной из битовых плоскостей групп спектральных коэффициентов аппроксимирующего ВЗ кадра.

Для того, чтобы выполнить помехоустойчивое декодирование, декодер сначала оценивает параметры шума, искажившего ВЗ кадр в процессе аппроксимации. Процесс оценки параметров шума будем называть *моделированием ошибок межкадрового предсказания*. В этой процедуре используется информация об аппроксимации разностей между соответствующими спектральными коэффициентами ВЗ кадра и аппроксимирующего кадра. Делается допущение о том, что ста-

статистические характеристики этих разностей аппроксимируются распределением Лапласа, параметр которого можно в таком случае оценить по критерию максимума правдоподобия. Следует отметить, что описанный сценарий является неприменимым на практике, т. к. подразумевает наличие на стороне декодера информации об оригинальном ВЗ кадре.

После завершения процедуры оценки параметров ошибок декодер выполняет помехоустойчивое декодирование битовых плоскостей спектральных коэффициентов ВЗ кадра с использованием соответствующих битовых плоскостей спектральных коэффициентов аппроксимирующего кадра и проверочных бит, запрашиваемых у кодера. Декодирование начинается с наиболее значимой битовой плоскости наиболее значимой группы спектральных коэффициентов. Для определения успешного декодирования используется информация об оригинальном ВЗ кадре (отметим, что этот подход также является нереализуемым на практике.). Как только очередная битовая плоскость успешно декодирована, декодер приступает к декодированию следующей по значимости битовой плоскости. Декодирование следующей группы спектральных коэффициентов начинается после успешного декодирования всех битовых плоскостей текущей группы.

После завершения турбо декодирования, спектральные коэффициенты восстанавливаются из битовых плоскостей и выполняется обратное спектральное преобразование, в результате которого на стороне декодера восстанавливается ВЗ кадр.

Последней операцией на стороне декодера является восстановление порядка следования кадров: восстановленные ВЗ кадры помещаются в видеопоследовательность между соответствующими базовыми кадрами.

1.6.2 Концепция PRISM

Концепция *PRISM* (от Power-efficient, Robust, High-compression, Syndrome-based Multimedia coding) была предложена в Калифорнийском университете в Беркли почти одновременно с Стэнфорд [16], [17]. В отличие от Стэнфорд в PRISM обрабатываются не кадры целиком, а блоки кадров. Ещё одним существенным отличием является отсутствие обратной связи. Ключевой идеей PRISM явля-

ется использование корреляции старших спектральных коэффициентов соседних блоков на одном кадре. Структурная схема процесса обработки блоков с использованием методов распределенного кодирования в концепции PRISM приведена на рисунке 1.12.

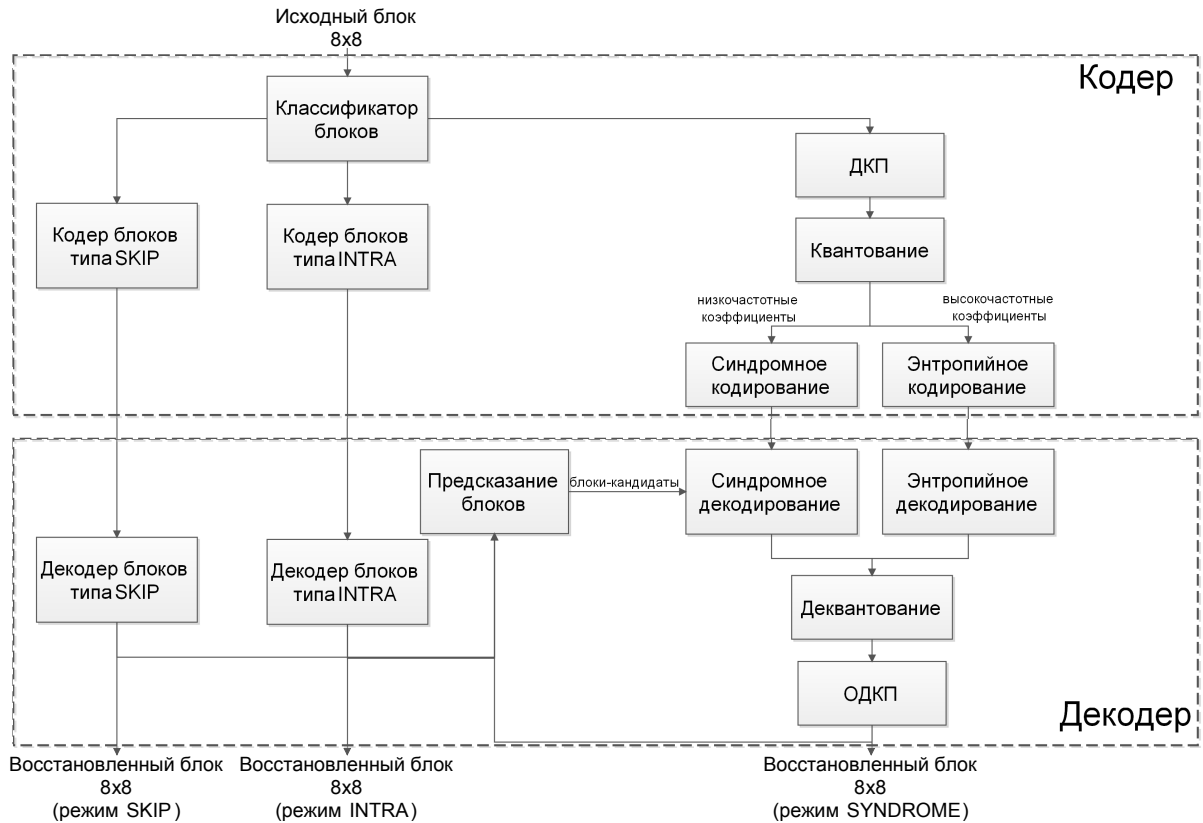


Рисунок 1.12 — Типовая схема кодека, основанного на концепции PRISM

Кодер. Каждый кадр видеопоследовательности разбивается на непересекающиеся блоки размером 8×8 пикселей. К каждому блоку применяется дискретное косинусное преобразование и равномерное квантование. Затем блоку ставится в соответствие метка класса, которая определяет способ кодирования блока. Метка класса зависит от степени корреляции между текущим блоком, который надо сжать, и блоком предсказания из базового кадра. Процедура предсказания, в зависимости от ограничений на сложность, может работать в одном из двух режимов:

- блок для предсказания ищется в предыдущем обработанном кадре с использованием низкосложностной процедуры оценки движения;
- блок для предсказания выбирается из одного из блоков, смежных с кодируемым блоком, на обрабатываемом кадре.

В зависимости от степени похожести сжимаемого блока на предсказанный блок, метка класса может принимать одно из трех значений.

- SKIP. В таком случае пиксели блока не кодируются (блок очень похож на предсказание).
- INTRA. Текущий блок сжимается независимо от остальных блоков (блок совсем не похож на предсказание).
- SYNDROME. Текущий блок сжимается с использованием вычислительно простой процедуры, основанной на распределенном кодировании (промежуточная степень схожести блока и предсказания).

Для блоков с меткой SYNDROME наиболее значимые биты квантованных спектральных коэффициентов кодируются с использованием синдромного кодирования, пример которого был приведен в подразделе 1.3. В качестве помехоустойчивого кода в PRISM используется код БЧХ. Наименее значимые биты кодируются обычным энтропийным кодом.

В дополнение к сжатым данным блока кодер формирует 16-битную контрольную сумму CRC, которая, как будет описано далее, используется декодером для выбора дополнительной информации.

Декодер. Декодер для каждого блока формирует набор блоков-кандидатов для дополнительной информации. Блоки-кандидаты выбираются по некоторому правилу из множества уже восстановленных блоков кадра. Каждый блок из этого набора интерпретируется как сторонняя информация, и выполняется синдромное декодирование. При этом наиболее значимые биты спектральных коэффициентов декодируются с использованием синдромных бит, а наименее значимые - с использованием энтропийного декодера. Затем для каждого восстановленного блока осуществляется расчет контрольной суммы CRC по аналогии с кодером. Блок, у которого рассчитанная контрольная сумма совпадет с контрольной суммой, полученной от кодера, выбирается в качестве декодированного блока.

После декодирования блока выполняется обратное дискретное косинусное преобразование, в результате которого на стороне декодера рассчитываются пиксели восстановленного блока.

Следует отметить, что в силу того, что в схеме PRISM нет обратной связи, возможна ситуация, когда блок на стороне декодера не удастся восстановить без потерь. Подобная ситуация может возникнуть в том случае, если в наборе блоков-кандидатов не найдется блока, достаточно похожего на сжатый блок, чтобы успешно исправить все ошибки в дополнительной информации.

1.7 Обоснование выбора концепции распределенного видеокодека

Следует отметить тот факт, что концепция Стэнфорд получила большее распространение, чем PRISM. Это можно объяснить следующими наблюдениями:

- схема кодека, работающего в соответствии с концепцией Стэнфорд, в большей степени, чем PRISM, соответствует существующей парадигме кодирования видеопоследовательностей, в рамках которой поток кадров разбивается на два подпотока: опорные и промежуточные;
- концепция Стэнфорд предоставляет большую гибкость при выборе сферы применения, т. к. в ней присутствует возможность разнесения кодеров опорных и промежуточных кадров в пространстве/времени;
- кодеры, основанные на концепции Стэнфорд обладают меньшей сложностью, по сравнению с PRISM, за счет отсутствия выбора режима кодирования блоков;
- наличие обратной связи от кодера к декодеру обеспечивает кодекам, основанным на концепции Стэнфорд, возможность восстановления данных на стороне декодера без потерь.

В связи с этим далее в данной работе основное внимание будет уделено схемам распределенного кодирования, основанным на концепции Стэнфорд.

1.8 Модель распределенного кодирования на базе проекта DISCOVER

В 2005 году ряд исследовательских групп из нескольких Европейских университетов запустили совместный проект DISCOVER (DIStributed COding for Video sERvices) [18], в рамках которого была разработана базовая модель видеокодека, основанного на концепции Стэнфорд. Структурная схема распределенного кодека, работающего в соответствии с моделью DISCOVER, приведена на рисунке 1.13. Здесь и далее будем называть данную модель *моделью кодека DISCOVER*.

Следует отметить, что по сравнению с концепцией Стэнфорд, модель кодека DISCOVER представляет собой существенно доработанное решение [19]. Одной из целей проекта была разработка методов, которые позволили бы обойти

ряд конструктивных особенностей концепции Стэнфорд, не позволяющих реализовать распределенный видеокodeк на практике в реальной системе. В частности, авторы DISCOVER разработали методы оценки параметров ошибок межкадрового предсказания и способы управления битовой скоростью, не требующие наличия информации о промежуточном кадре на стороне декодера.

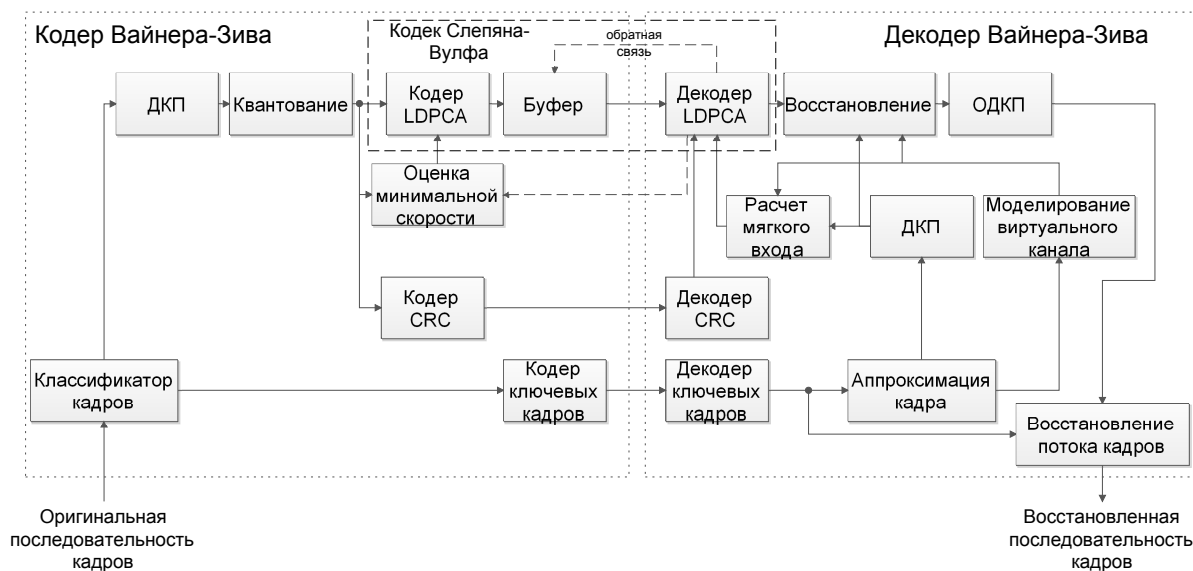


Рисунок 1.13 — Структурная схема модели кодека DISCOVER

Как и в концепции Стэнфорд в DISCOVER поток кадров разбивается на два подпотока: базовые (ключевые) кадры и промежуточные кадры. Ключевые кадры кодируются методом, аналогичным режиму Intra стандарта H.264 [20] без устранения временной избыточности. Приведем описание процесса обработки промежуточных кадров в DISCOVER. Межкадровое предсказание промежуточных кадров осуществляется с помощью процедуры временной интерполяции по паре ключевых кадров. Исправление ошибок временного предсказания происходит в спектральной области. Для этого исходный кадр на стороне кодера и интерполированный кадр на стороне декодера разбиваются на непересекающиеся блоки размером 4×4 пикселя, которые подвергаются дискретному косинусному преобразованию (ДКП) [21]. Спектральные коэффициенты всех блоков группируются по частотам по следующему правилу:

$$G_{k,l} = \{\mathbf{T}(k + 4i, l + 4j)\},$$

где \mathbf{T} – $W \times H$ матрица спектральных коэффициентов, $\mathbf{T}(k + 4i, l + 4j)$ – спектральный коэффициент при паре частот (k, l) для блока с номером (i, j) . В итоге

формируется шестнадцать матриц $G_{k,l}$ размером $(W/4) \times (H/4)$ (W и H – ширина и высота кадра в пикселях соответственно), состоящих из спектральных коэффициентов при паре частот (k,l) , которые в дальнейшем будем называть группой (band). Затем коэффициенты в каждой группе (k,l) квантуются с использованием скалярного квантования с шагом $Q_{k,l}$. Шаг квантования определяется по диапазону значений коэффициентов в соответствующей группе [22]. Для различных групп используются разные методы квантования:

- группа с коэффициентами DC: равномерное скалярное квантование;
- группы с коэффициентами AC: скалярное квантование с расширенной нулевой зоной.

Затем каждая группа квантованных коэффициентов разбивается на битовые плоскости $S_{k,l}^{(b)}(i,j)$, где b определяет номер бита в двоичном представлении $G'_{k,l}(i,j)$. Битовые плоскости нумеруются, начиная со старшей (наиболее значимой). Обработка битовых плоскостей также начинается со старшей плоскости. На кодере каждая битовая плоскость интерпретируется как искаженное кодовое слово кода LDPCA (Low-Density Parity Check Codes with Accumulation) [23]. Под искажением понимается тот факт, что синдром, рассчитанный для бит плоскости, не будет являться нулевым. Кодек ориентирован на работу с видеоданными форматов CIF (разрешение 352x288 пикселей) и QCIF (разрешение 176x144 пикселей), поэтому длина кода составляет 101376 бит для CIF и 25344 бит для QCIF. Данные параметры являются верхней оценкой для количества бит, которые декодер может запросить у кодера. Помимо бит синдрома, рассчитанных для каждой плоскости, кодер дополнительно формирует CRC (cyclic redundancy check) последовательность с использованием многочлена $f(x) = x^8 + x^2 + x + 1$ [22]. Эта последовательность в обязательном порядке входит в состав битового потока.

На приемной стороне декодер пытается восстановить различия в соответствующих битовых плоскостях исходных данных и результатов предсказания, используя для этого биты синдрома, сформированные кодером. Декодер DISCOVER работает по схеме с мягким входом (Soft Input), вычисляя для каждого бита кодового слова надежность [22]. Процедура декодирования основана на алгоритме распространения надежности. Она выполняется для каждой битовой плоскости, начиная со старшей. Для определения успешности декодирования декодер по аналогии с кодером рассчитывает контрольную сумму по информационной последо-

вательности и сверяет её с полученной от кодера. Если CRC не совпадают, декодер запрашивает у кодера новую порцию бит синдрома и продолжает декодирование.

Использование кодов LDPC обусловлено возможностью использования при декодировании только части бит синдрома. Таким образом, в схему DVC вводится дополнительная возможность управления битовой скоростью. В DISCOVER управление битовой скоростью осуществляется декодером при помощи канала обратной связи (feedback channel), по которому передаются запросы дополнительных бит, если декодирование с использованием имеющегося числа бит не удастся.

В результате можно сделать вывод, что на эффективность устранения временной избыточности на стороне декодера и, в конечном итоге, на сжатие будут влиять следующие факторы.

- Точность генерации дополнительной информации: чем меньше различий между дополнительной информацией декодера и исходными данными на стороне кодера, тем меньше проверочных бит необходимо затратить на их исправление.
- Эффективность исправления ошибок в дополнительной информации, на которую влияют:
 - модуль оценки параметров виртуального канала: надежности символов оказывают существенное влияние на эффективность исправления ошибок с использованием корректирующих кодов;
 - модуль помехоустойчивого кодирования: чем выше корректирующая способность кода, тем больше ошибок он позволяет исправить при фиксированной длине кода.

Рассмотрению вопросов, связанных с этими факторами, и посвящена настоящая диссертационная работа.

1.9 Области применения распределенного кодирования

В данном подразделе перечислены основные прикладные области, в которых может применяться распределенное кодирование видеоданных [24]. Удобно

ввести в рассмотрение следующую классификацию прикладных задач в зависимости от количества источников информации:

- информация поступает от одного источника;
- информация поступает от нескольких источников.

В задачах, относящихся к первому классу, кодер искусственным образом выделяет в поступающих ему на вход данных подпоток, которые затем обрабатываются как выходы зависимых источников. Основным выигрыш от применения распределенного кодирования в таких задачах заключается в существенном уменьшении сложности процедуры сжатия на стороне кодера и, как следствие, меньших габаритах кодирующего устройства и меньшем энергопотреблении. В рамках этого класса можно выделить следующие прикладные задачи:

- передача данных систем видеонаблюдения с низким энергопотреблением;
- организация мобильных видеоконференций;
- мобильная видеопочта;
- беспроводная капсульная эндоскопия;
- и т. д.

В задачах, относящихся ко второму классу, как правило натуральным образом присутствует несколько зависимых источников, причем их количество может быть очень большим. Кодек в таком случае может выделить несколько основных источников, данные от которых будут использоваться как сторонняя информация при декодировании остальных источников. Использование распределенного кодирования в таком случае может, помимо уменьшения габаритов и энергопотребления, повысить надежность восстановления данных, а также отказаться от использования связей между источниками при сжатии. Ко второму классу можно отнести следующие прикладные задачи:

- сжатие многомерных (3D и 4D) изображений;
- обработка информации в визуальных сенсорных сетях;
- сжатие многовидовых видеопоследовательностей;
- и т. д.

Однако следует отметить, что существующая инфраструктура передачи данных от мобильных видеоисточников на данный момент плохо приспособлена для использования методов распределенного кодирования на практике. В частности для систем с одним источником перенос сложности со стороны кодера на

декодер требует наличия в сети специальных узлов-транскодеров, которые будут выполнять перекодирование сжатых видеопотоков в формат, обеспечивающий вычислительно простую процедуру декодирования. Тем не менее рассмотрение методов распределенного кодирования в качестве перспективной технологии является актуальной задачей.

1.10 Выводы по разделу

В качестве выводов по разделу можно отметить, что задача распределенного кодирования видеоданных представляет существенный интерес, т. к. данный подход может найти применение во многих современных прикладных задачах, в которых требуется передать видеoinформацию от источников с ограниченными вычислительными ресурсами. В разделе приведено описание основных теоретических результатов и практических методов кодирования зависимых источников. Представлена классификация методов распределенного кодирования. Приведены основные концепции, лежащие в основе сжатия с использованием данного подхода. Далее было дано описание модели распределенного кодирования на базе проекта DISCOVER. В рамках модели выделены основные факторы, оказывающие влияние на сжатие:

- точность генерации дополнительной информации;
- эффективность помехоустойчивого кодирования.

Исследованию этих факторов и разработке новых алгоритмов обработки видеoinформации в рамках системы распределенного кодирования и посвящена данная работа. Несмотря на то, что настоящий раздел не содержит новых результатов, его материал является вводным для последующих разделов.

2 ГЕНЕРАЦИЯ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ

2.1 Вводные замечания

Настоящий раздел посвящен задаче генерации дополнительной информации декодера в распределенных видеокодеках. В начале приводится обзор типовых подходов к решению данной задачи и выделяется ключевая операция, связанная с оценкой движения между кадрами видеопоследовательности. Далее дается краткое описание метода оценки движения, используемого при генерации дополнительной информации в базовой модели распределенного кодирования на базе проекта DISCOVER, указываются его основные особенности и потенциальные недостатки. В настоящем разделе данный алгоритм считается базовым для последующего анализа и улучшений. Далее вводится в рассмотрение понятие «истинного движения» как векторного поля, отражающего реальное смещение объектов между парой кадров видеопоследовательности. Данное понятие затем формализуется с использованием оптимизационной задачи и приводится описание нового эвристического алгоритма оценки истинного движения, выполняющего поиск локального минимума введенной оптимизационной задачи. Объяснение сходимости алгоритма к некоторому локальному минимуму дается на качественном уровне. Вопросы, связанные с оценкой эффективности данного алгоритма, рассматриваются в разделе 4. Основные результаты данного раздела опубликованы в работах [4], [25], [26].

2.2 Обзор методов генерации дополнительной информации

Как было показано в подразделе 1.6.1, процедура генерации дополнительной информации выполняется декодером для формирования вспомогательных данных, которые интерпретируются как выход одного из зависимых источников в схеме распределенного кодирования. При обработке видеоданных ключевым методом процесса генерации дополнительной информации является выполнение на

стороне декодера предсказания промежуточного кадра по ранее декодированным кадрам видеопоследовательности [4]. В зависимости от того, какой подход лежит в основе процедуры предсказания, выделяют два класса методов генерации дополнительной информации:

- подходы, основанные на временной интерполяции [27];
- подходы, основанные на временной экстраполяции [28].

В схемах, основанных на экстраполяции, для формирования дополнительной информации используются только кадры видеопоследовательности, предшествующие во времени предсказываемому кадру. Существенным преимуществом такого подхода является то, что не увеличивается задержка декодирования, т. к. поток кадров декодируется в том же порядке, в котором он поступает на вход кодера. Однако, в силу того что при экстраполяции не доступна информация о реальном движении объектов на будущих кадрах, данный метод может в общем приводить к большему числу ошибок по сравнению с методами, основанными на интерполяции.

В подходах, основанных на интерполяции, для предсказания промежуточного кадра используются как кадры из прошлого, так и кадры из будущего. Таким образом, на декодере необходимо сначала восстановить так называемые опорные кадры, а затем по ним осуществить интерполяцию, что позволяет выполнять предсказание точнее по сравнению с экстраполяцией, но приводит к увеличению задержки декодирования. Однако, т. к. данный недостаток, как правило, не является критичным для типовых сценариев применения распределенного кодирования видеоданных, именно подходы использующие интерполяцию кадров получили наибольшее распространение.

Обобщенная схема типового метода межкадрового предсказания, основанного на временной интерполяции кадров, приведена на рисунок 2.1. Она включает два основных блока:

- блок оценки движения (Motion Estimation, ME), используется для оценки изменений между соседними кадрами;
- блок компенсации движения, используется для формирования нового кадра с использованием информации от предыдущего блока.

Перед тем, как переходить к описанию особенностей процедуры предсказания кадров с использованием временной интерполяции введем ряд необходимых определений.

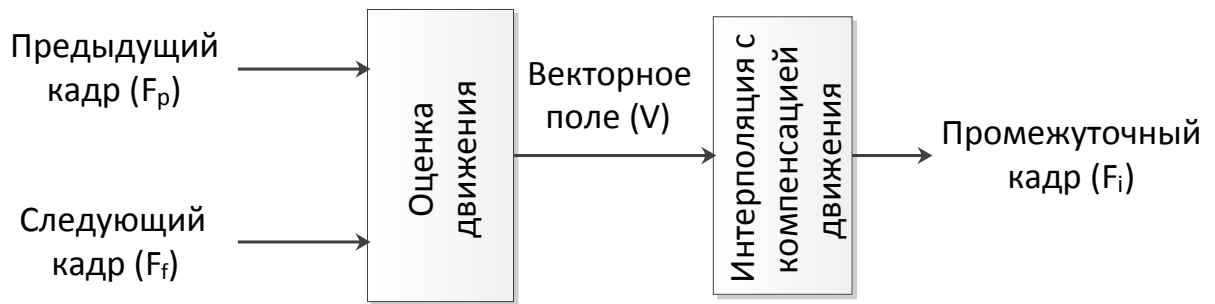


Рисунок 2.1 — Обобщенная схема типового алгоритма временной интерполяции

Определение 2.1. Будем называть вектором движения двумерный вектор, показывающий изменение координат пикселя между кадрами.

Определение 2.2. Векторное поле – матрица, содержащая векторы движения всех пикселей кадра, для которого выполняется процедура оценки движения.

В схеме предсказания кадров назначение блока оценки движения заключается в поиске такого векторного поля, которое отражает перемещение объектов на оригинальных кадрах видеопоследовательности. Эта задача известна как поиск *истинного движения* (True Motion Estimation) [29].

Определение 2.3. Векторное поле, соответствующее истинному движению, будем называть истинным векторным полем.

Истинное векторное поле используется в блоке компенсации движения для интерполяции объектов на промежуточных кадрах в тех координатах, в которые объекты должны сместиться в соответствии с траекторией их движения в видеопоследовательности.

Можно выделить две основные задачи, возникающие при временной интерполяции кадров:

1. поиск истинного движения [30];
2. обработка регионов на интерполированном кадре, не имеющих вектора движения или имеющих более одного вектора движения [31].

Следует отметить, что задача поиска истинного движения отличается от задачи оценки движения, возникающей при сжатии видеопоследовательностей. В связи с этим для формализации понятия истинного движения, как правило, используется ряд допущений о типовом векторном поле, отражающем специфику перемещений объектов на кадрах. Одним из наиболее распространенных допущений

является допущение о гладкости векторного поля, заключающееся в том, что векторы движения, соответствующие пространственно близким регионам, должны быть похожи. При этом под похожестью векторов понимается как похожесть амплитуды, так и направления. Как правило, для расчета гладкого векторного поля используются схемы, основанные на предсказании векторов движения, в которых поиск вектора для нового региона осуществляется с учетом векторов, сформированных ранее для соседних регионов. Процедура предсказания может осуществляться как в пространственной области, так и во временной. Однако, допущение о гладкости векторного поля, как правило, является неверным для регионов, содержащих границы объектов. При поиске истинного движения для таких регионов используется предсказание с учетом векторов, полученных по обе стороны границы объекта. Предсказанные направления затем используются для инициализации векторов, соответствующих региону, содержащему границу. Подобный подход используется в одном из первых алгоритмов поиска истинного движения – *трехмерном рекурсивном поиске* (3D Recursive Search – 3DRS) [30].

Вторая задача временной интерполяции связана с назначением векторов движения областям на интерполированном кадре. Существует два основных подхода к решению этой задачи:

1. однонаправленная оценка/компенсация движения [30];
2. двунаправленная (билатеральная) оценка/компенсация движения [32].

В соответствии с однонаправленным подходом один из смежных базовых кадров разбивается на непересекающиеся блоки, и для каждого блока ищется наиболее похожий блок в другом кадре. Вектор движения определяется как расстояние между координатами этих блоков на кадрах. Координаты блока на промежуточном кадре находятся посередине между координатами блоков на базовых кадрах. Интерполяция осуществляется усреднением пикселей, находящихся на совпадающих позициях в блоках и помещением нового «усредненного» блока в соответствующую позицию на промежуточном кадре. У такого подхода есть существенный недостаток, поскольку в промежуточном кадре возможно появление регионов, для которых интерполяция с использованием найденных векторов будет неоднозначна:

- области, не ассоциированные ни с одним из найденных векторов (так называемые «дыры»);

- регионы, которые ассоциированы с несколькими найденными векторами (так называемые «наложения»).

Двунаправленный подход позволяет избежать этих проблем. Основным допущением билатерального подхода является допущение о равномерности и прямолинейности движения объектов. В таком случае на блоки разбивается не один из опорных кадров, а промежуточный кадр. Для каждой фиксированной позиции блока на промежуточном кадре осуществляется поиск похожих блоков на оригинальных кадрах, при котором векторы смещения коллинеарны, разнонаправлены и откладываются относительно блока в промежуточном кадре. Этот подход не решает проблемы, возникающие из-за появления/исчезновения/наложения объектов на кадрах, но, в отличие от однонаправленного поиска, предоставляет завершённую процедуру обработки регионов возле этих объектов.

Следует отметить, что процедура временной интерполяции является только частью генерации дополнительной информации. Полученная аппроксимация промежуточного кадра далее, по аналогии с кодером, подвергается спектральному преобразованию и квантованию. Квантованные коэффициенты затем разбиваются на битовые плоскости, и именно битовые плоскости являются дополнительной информацией. Тем не менее, далее в данном разделе рассматривается только задача межкадрового предсказания на основе временной интерполяции, т. к. основная причина наличия ошибок в дополнительной информации связана именно с ошибками аппроксимации.

2.3 Аппроксимация промежуточных кадров в модели DISCOVER

2.3.1 Основные определения и обозначения

При описании алгоритма оценки движения будем рассматривать пару соседних кадров оригинальной видеопоследовательности \mathbf{F}_p и \mathbf{F}_f , между которыми необходимо построить векторное поле \mathbf{V} , описывающее смещение объектов между этими кадрами. Кадры \mathbf{F}_p и \mathbf{F}_f будем называть опорными или базовыми кадрами. Матрица векторов движения \mathbf{V} (векторное поле) используется для интер-

поляции нового кадра F_i , расположенного во времени между базовыми кадрами. Кадр F_i будем называть интерполированным или промежуточным кадром. Здесь и далее будем считать, что обрабатываемые кадры являются многокомпонентными (или цветными) изображениями, представленными одной яркостной компонентой Y и двумя хроматическими компонентами C_b и C_r . Однако следует отметить, что описанные далее алгоритмы легко обобщаются на изображения с произвольным числом компонент.

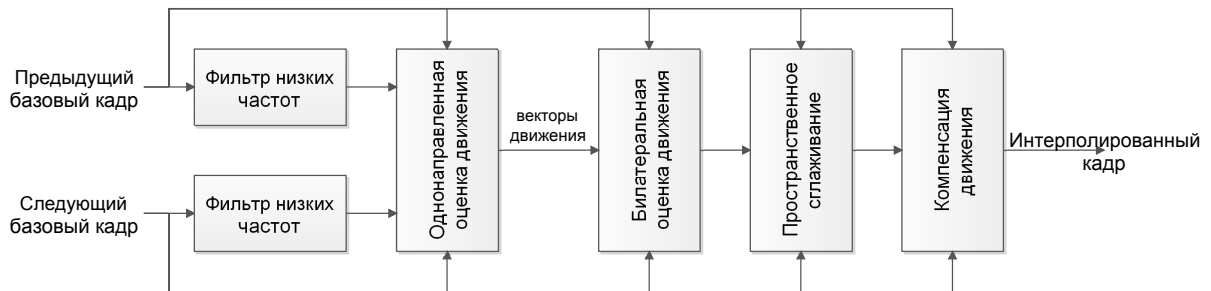


Рисунок 2.2 — Схема алгоритма межкадрового предсказания в модели DISCOVER

Перед тем, как приводить подробное описание процедуры аппроксимации, рассмотрим её основные этапы и их назначение. В модели DISCOVER оценка движения реализуется за два шага (рисунок 2.2) [33], [34]. На первом шаге выполняется однонаправленная оценка движения, на втором – билатеральная. Перед первым шагом оба базовых кадра подвергаются низкочастотной фильтрации для уменьшения шумов. Считается что такого рода предобработка способствует повышению точности оценки движения. После однонаправленной оценки выполняется уточняющая билатеральная оценка с меньшим радиусом поиска. На всех шагах используется блоковая оценка движения, т.е. один из кадров разбивается на непересекающиеся квадратные блоки пикселей. Для каждого такого блока определяется *вектор движения* $\mathbf{v} = (v_y, v_x)^T$, соответствующий изменению координат y и x этого блока между ключевыми кадрами. В качестве критерия, по которому оценивается изменение местоположения блока, в DISCOVER используется среднеквадратичное искажение (MSE, Mean Squared Error).

Введем необходимые обозначения. Для упрощения изложения будем полагать, что выполняются следующие соотношения:

$$\frac{w}{b}, \frac{h}{b} \in \mathbb{N},$$

где w и h – ширина и высота кадров соответственно; b – размер блока, для которого выполняется оценка движения.

Назовем множество координат пикселей на кадре сеткой пикселей. Формально сетку пикселей можно определить как

$$\mathcal{P} = \{\mathbf{p} = (y, x)^T | y \in \mathcal{Y}, x \in \mathcal{X}\},$$

где $\mathcal{X} = \{1, 2, \dots, w\}$, $\mathcal{Y} = \{1, 2, \dots, h\}$. В общем случае значение пикселя в координате $\mathbf{p} = (y, x)^T$ на кадре \mathbf{F} задается как триплет $\mathbf{F}(\mathbf{p}) = (\mathbf{Y}(\mathbf{p}), \mathbf{Cb}(\mathbf{p}), \mathbf{Cr}(\mathbf{p}))$, однако, т. к. в модели DISCOVER обрабатываются только последовательности с кадрами в градациях серого, будем рассматривать только обработку яркостной компоненты, т. е. будем считать, что $\mathbf{F}(\mathbf{p}) = \mathbf{Y}(\mathbf{p})$. По аналогии с сеткой пикселей определим сетку блоков, как множество координат блоков (координат левого верхнего пикселя в блоке) на кадре:

$$\mathcal{G} = \{\mathbf{g}_{i,j} = (i, j)^T | i \in \mathcal{I}, j \in \mathcal{J}\},$$

где $\mathcal{I} = \{1, 2, \dots, \frac{h}{b}\}$, $\mathcal{J} = \{1, 2, \dots, \frac{w}{b}\}$ – высота и ширина сетки блоков соответственно.

Множество координат пикселей, находящихся в блоке с индексом $(i, j)^T$ определим как

$$\mathcal{B}_{i,j} = \{(y, x)^T \in \mathcal{P} | y \in \mathcal{Y}_i, x \in \mathcal{X}_j\},$$

где $\mathcal{Y}_i = \{(i-1)b + 1, \dots, ib\}$; $\mathcal{X}_j = \{(j-1)b + 1, \dots, jb\}$.

Приведем описание алгоритма межкадрового предсказания в модели DISCOVER с учетом введенных обозначений.

2.3.2 Однонаправленная оценка движения

Поиск векторов движения осуществляется как оценка смещения в последующем ключевом кадре \mathbf{F}_f относительно предыдущего \mathbf{F}_p . Для каждого блока на предыдущем ключевом кадре ищется вектор движения с минимальной ошибкой сопоставления блоков:

$$\mathbf{v}_{i,j}^{(u)} = \arg \min_{\mathbf{v} \in \mathcal{V}_s^{(u)}} \frac{1}{b^2} \sum_{\mathbf{p} \in \mathcal{B}_{i,j}} (\mathbf{F}_p(\mathbf{p}) - \mathbf{F}_f(\mathbf{p} + \mathbf{v}))^2,$$

где $\mathcal{V}_s^{(u)} = \{(-r, -r)^T, (-r + \delta, -r)^T, (-r, -r + \delta)^T, \dots, (r, r)^T\}$ – множество возможных векторов движения; r – радиус поиска, определяющий максимальную длину вектора, δ – расстояние между векторами.

Таким образом, на первом шаге для каждого блока на предыдущем кадре ищется с помощью перебора векторов в некотором радиусе оптимальный вектор движения, соответствующий минимальному среднему квадрату ошибки сопоставления блоков. Данный вектор соответствует смещению всех пикселей в блоке между кадрами. Координаты пикселя на интерполированном кадре \mathbf{F}_i рассчитываются по правилу:

$$\mathbf{p}_i = \text{round}(\mathbf{p}_p + \alpha \mathbf{v}),$$

где $\mathbf{p}_i = (y, x)_i^T$ и $\mathbf{p}_p = (y, x)_p^T$ – координаты пикселей в интерполированном и предыдущем ключевом кадрах соответственно, $\alpha = \frac{d_p}{d}$, d – расстояние (количество кадров) между ключевыми кадрами, d_p – расстояние между предыдущим опорным кадром и интерполированным кадром. Здесь и далее для упрощения обозначений будем рассматривать случай, когда $\alpha = \frac{1}{2}$, т. е. промежуточный кадр находится посередине между базовыми.

Результатом данного поиска является векторное поле, причем векторы движения поставлены в соответствие пикселям одного из базовых кадров, что может приводить к «дырам» и «коллизиям» на промежуточном кадре. Для устранения этих эффектов на втором шаге алгоритма выполняется дополнительная билатеральная оценка движения.

2.3.3 Билатеральная оценка движения

Первой операцией билатеральной оценки является преобразование однонаправленного векторного поля, полученного с предыдущего шага алгоритма, в билатеральное. Для этого промежуточный кадр разбивается на непересекающиеся блоки и для каждого блока осуществляется выбор билатерального вектора из мно-

жества однонаправленных векторов, пересекающих этот блок. Выбор заключается в поиске такого однонаправленного вектора, который «покрывает» наибольшее число позиций пикселей в соответствующем блоке на промежуточном кадре.

Таким образом, в результате данного преобразования на промежуточном кадре строится сетка непересекающихся блоков, причем каждому блоку поставлен в соответствие билатеральный вектор движения, т. е. построено начальное билатеральное векторное поле. Данное поле уточняется в процессе билатеральной оценки движения.

В процессе билатеральной оценки движения на ключевых кадрах ищутся похожие блоки относительно координат \mathbf{p}_t для блоков в интерполированном, анализируя разности:

$$\mathbf{v}_{i,j}^{(b)} = \arg \min_{\mathbf{v} \in \mathcal{V}_{i,j}^{(b)}} \frac{1}{b^2} \sum_{\mathbf{p} \in \mathcal{B}_{i,j}} (\mathbf{F}_p(\mathbf{p} - \mathbf{v}) - \mathbf{F}_f(\mathbf{p} + \mathbf{v}))^2, \quad (2.1)$$

где $\mathcal{V}_{i,j}^{(b)} = \{\mathbf{v}'_{i,j} + (-r', -r')^T, \mathbf{v}'_{i,j} + (-r' + 1, -r')^T, \mathbf{v}'_{i,j} + (-r', -r' + 1)^T, \dots, \mathbf{v}'_{i,j} + (r', r')^T\}$ – множество возможных билатеральных векторов движения, учитывающее результат однонаправленной оценки $\mathbf{v}'_{i,j}$.

Поиск векторов движения при билатеральной оценке осуществляется в радиусе r' относительно начального однонаправленного смещения, причем $r' < r$.

После билатеральной оценки у каждой координаты пикселя на промежуточном кадре будет оцененный вектор движения, т. е. на интерполированном кадре не будет регионов с «коллизиями» и «дырами», которые возможны после первого шага.

2.3.4 Сглаживание векторного поля

Следующим шагом алгоритма является пространственное сглаживание *векторного поля*. Для этого используется взвешенная медианная фильтрация. Для блока с координатой $\mathbf{g}_{i,j}$ и вектором движения $\mathbf{v}_{i,j}$ эта операция определена как:

$$\mathbf{v}_{i,j}^* = \arg \max_{\mathbf{v} \in \{\mathbf{v}_1, \dots, \mathbf{v}_n\}} \left(\sum_{k=1}^n w_k (\|\mathbf{v}_{i,j} - \mathbf{v}_k\|_L - \|\mathbf{v} - \mathbf{v}_k\|_L) \right), \quad (2.2)$$

где \mathbf{v}_i^* – результат фильтрации, $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ – множество участвующих в фильтрации \mathbf{v}_i векторов, в это множество входят векторы соседних блоков, а также вектор для блока на позиции $\mathbf{g}_{i,j}$, найденный для предыдущего интерполированном кадра; w_k – весовые коэффициенты, определяемые как:

$$w_k = \frac{\text{MSE}(\mathbf{v}_{i,j}, \mathbf{g}_{i,j})}{\text{MSE}(\mathbf{v}_k, \mathbf{g}_{i,j})}, \quad (2.3)$$

где $\text{MSE}(\mathbf{v}, \mathbf{p})$ – функция расчета среднеквадратичной ошибки (2.1) при применении вектора \mathbf{v} к блоку на позиции \mathbf{p} . Полученное сглаженное билатеральное векторное поле \mathbf{V} затем используется на последнем шаге алгоритма для выполнения компенсации движения и формирования интерполированного кадра.

2.3.5 Интерполяция с компенсацией движения

Перед этим шагом каждому пикселю на промежуточном кадре поставлен в соответствие билатеральный вектор движения, полученный в результате оценки движения для блока, в котором этот пиксель находится. Интерполяция с компенсацией движения заключается в усреднении интенсивностей соответствующих пикселей в базовых кадрах и помещении полученного значения в соответствующую координату пикселя на промежуточном кадре:

$$\mathbf{F}_i(\mathbf{p}) = \frac{1}{2} [\mathbf{F}_p(\mathbf{p} - \mathbf{V}(\mathbf{p})) + \mathbf{F}_f(\mathbf{p}_i + \mathbf{V}(\mathbf{p}))], \forall \mathbf{p} \in \mathcal{P}. \quad (2.4)$$

2.3.6 Параметры процедуры аппроксимации кадров

В завершении описания алгоритма укажем параметры, используемые в DISCOVER [34]:

- сглаживание выполняется с использованием усредняющего фильтра размером 3×3 пикселя;
- размер блока на при однонаправленной оценке движения: $b = 16$;

- радиус поиска при однонаправленной оценке движения: $r = 32$;
 - размер блока на при билатеральной оценке движения: $b = 8$;
 - расстояние между перебираемыми векторами: $\delta = 2$;
 - радиус поиска при билатеральной оценке движения: $r' = 2$.
- Оценка движения выполняется с полупиксельной точностью.

2.3.7 Недостатки базового алгоритма

Одним из существенных недостатков используемого в DISCOVER метода генерации дополнительной информации является то, что в нем используется обычная блоковая оценка движения с полным перебором всех возможных векторов движения в некотором радиусе. Учет корреляции векторов соседних блоков осуществляется только при билатеральном допоиске и взвешенной медианной фильтрации векторного поля, но никак не используется в самой процедуре оценки движения. Таким образом, данный алгоритм в явном виде не учитывает допущения об истинном движении, приведенные в подразделе 2.2, и можно добиться существенного повышения качества аппроксимации, если учесть эти допущения при оценке движения.

2.4 Модель истинного движения в задаче временной интерполяции кадров

Перед тем, как приводить описание нового алгоритма поиска векторного поля, рассмотрим введенное ранее понятие истинного движения более формально. Будем считать, что истинное векторное поле может быть получено в результате оптимизации следующей функции при фиксированных кадрах \mathbf{F}_p и \mathbf{F}_f [35]:

$$\mathbf{V}^* = \arg \min_{\mathbf{V} \in \mathcal{V}} E(\mathbf{F}_p, \mathbf{F}_f, \mathbf{V}) \quad (2.5)$$

где \mathcal{V} – пространство всех возможных векторных полей \mathbf{V} и

$$E(\mathbf{F}_p, \mathbf{F}_f, \mathbf{V}, \alpha) = E_1(\mathbf{F}_p, \mathbf{F}_f, \mathbf{V}) + \alpha E_2(\mathbf{V}). \quad (2.6)$$

Слагаемое $E_1(\cdot, \cdot, \cdot)$ соответствует энергии разностного кадра между кадрами \mathbf{F}_p и \mathbf{F}_f с использованием векторного поля \mathbf{V} ; слагаемое $E_2(\cdot)$ отражает гладкость (похожесть соседних векторов) поля \mathbf{V} ; $\alpha \geq 0$ – коэффициент регуляризации между энергией разностного кадра и гладкостью векторного поля. Следует отметить, что обычно E_2 является обратным значением к гладкости, т.е. меньшее значение E_2 соответствует более гладкому полю \mathbf{V} .

Сформулированную оптимизационную задачу можно пояснить следующим образом: алгоритм оценки движения должен найти такое гладкое векторное поле, которое минимизирует энергию разностного кадра. Коэффициент α служит для балансировки между гладкостью и энергией (ошибкой сопоставления блоков). В том случае, если значение α близко к нулю, оптимальное векторное поле будет доставлять глобальный минимум энергии разностного кадра, но векторы движения при этом могут быть хаотичны. С другой стороны, большое значение α приводит к тому, что оптимальное векторное поле может быть чрезмерно сглаженным и не обеспечивать хорошего совпадения базовых кадров. Промежуточные значения α соответствуют сбалансированному гладкому векторному полю, обеспечивающему приемлемое совпадение базовых кадров. Следует отметить, что α является одним из параметров данной модели. Поиск значения α , соответствующего истинному движению является сложной задачей, т.к. в общем случае нельзя сказать, что зависимость значения целевой функции от α является унимодальной.

Предположим, что в пространстве \mathcal{V} введено отношение соседства. Обозначим через $N(\mathbf{V})$ множество соседей \mathbf{V} в \mathcal{V} . Говорят, что векторное поле \mathbf{V}^* доставляет локальный минимум $E(\cdot, \cdot, \cdot, \cdot)$ при фиксированных кадрах \mathbf{F}_p и \mathbf{F}_f тогда и только тогда, когда для всех $\mathbf{V} \in N(\mathbf{V})$ выполняется следующее соотношение:

$$E(\mathbf{F}_p, \mathbf{F}_f, \mathbf{V}^*) \leq E(\mathbf{F}_p, \mathbf{F}_f, \mathbf{V}). \quad (2.7)$$

В том случае, если неравенство выполняется для всех $\mathbf{V} \in \mathcal{V}$, говорят, что векторное поле \mathbf{V} доставляет глобальный минимум E .

Следует отметить, что существуют алгоритмы, которые для заданных α в некоторых случаях позволяют искать минимумы функции (2.5) в явном виде [36], [37]. Так как функция (2.5) зависит от большого числа параметров и не является унимодальной, найденные минимумы являются, как правило, локальными. Однако следует отметить, что с помощью формулы (2.5) описывается только модель движения, и векторные поля, соответствующие локальным и глобальным

минимумам этой модели, не обязательно точно отражают истинное движение объектов на кадрах. В частности, модель не учитывает неравномерность движения на границах объектов, где векторное поле, как правило, не является гладким. Поэтому найденное для данной модели решение не гарантирует высокого визуального качества в результате интерполяции. В связи с этим более хорошие результаты с точки зрения визуального качества достигаются с использованием подоптимальных алгоритмов, рассматривающих оптимизацию (2.5) неявно. Такие алгоритмы, как правило, с одной стороны основаны на эвристических подходах, нацеленных на повышение визуального качества интерполированных кадров, а с другой стороны, неявно косвенно минимизируют (2.5), т.е. выдают результат, согласованный с моделью истинного движения. В связи с этим далее в данном разделе приводится описание алгоритма оценки движения, принадлежащего именно к такому классу алгоритмов.

Процедура поиска оптимального вектора движения, лежащая в основе всех методов оценки движения, также может быть реализована с использованием либо полного перебора всех возможных векторов (оптимальный поиск), либо с использованием некоторой под-оптимального алгоритма. Полный перебор оптимален в том смысле, что он гарантирует минимум ошибки E_1 в 2.6. В ряде работ [38] указывается, что полный перебор плохо согласуется с поиском истинного движения, в особенности на текстурных регионах. Кроме того, процедура полного перебора всех векторов обладает высокой вычислительной сложностью. В связи с этим поиск оптимального вектора в задаче оценки истинного движения, как правило, осуществляется с использованием под-оптимальных методов, например, градиентного спуска [39].

2.5 Описание разработанного алгоритма аппроксимации промежуточных кадров с учетом истинного движения

2.5.1 Обоснование разработанного алгоритма

В основе разработанного алгоритма оценки движения (рисунок 2.3) лежит многоуровневая иерархическая билатеральная процедура блочной оценки движения с дополнительным итеративным поиском [26], [25]. На каждом уровне иерархии для фиксированного размера блока выполняются три основные операции:

1. инициализация уровня иерархии для подготовки кадров к оценке движения;
2. начальная оценка движения, используемая для поиска предварительного векторного поля, доставляющего локальный минимум E_1 в (2.6);
3. итеративный дополнительный поиск, повышающий гладкость векторного поля (уменьшение E_2 в (2.6)).

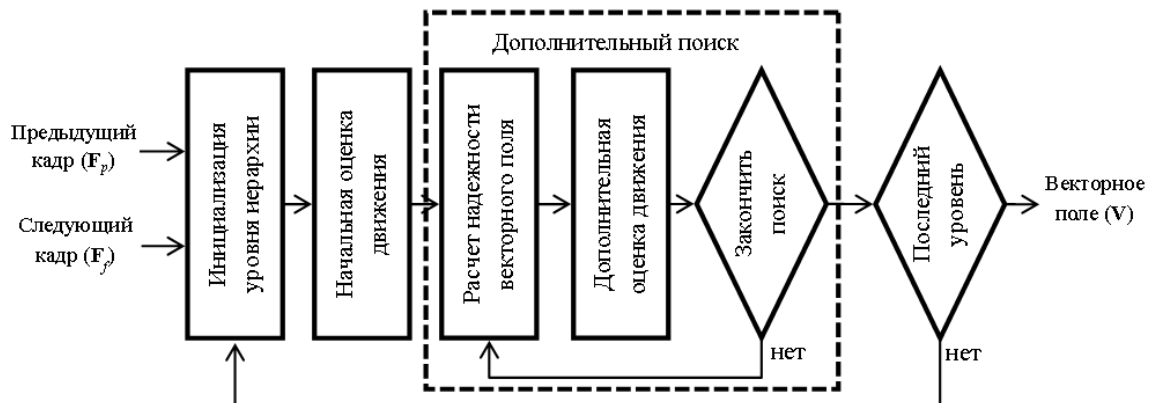


Рисунок 2.3 — Схема разработанного алгоритма оценки движения

На каждом уровне иерархии выполняется оценка движения для блоков фиксированного размера. Здесь и далее будем называть процесс обработки одного уровня стадией. Выходом стадии является векторное поле V^* , доставляющее локальный минимум энергии в выражении (2.6). Полученные векторы движения используются в качестве начального смещения на следующей стадии.

Иерархическая оценка реализована с использованием многосеточного подхода (multigrid), размер блока уменьшается с номером стадии [40], [41]. Основная идея такого подхода заключается в том, что сначала выполняется оценка движения для больших областей на базовых кадрах, затем эта оценка уточняется для меньших регионов и т. д. Подобный подход позволяет передавать информацию о глобальном движении между стадиями, приводя к тому, что векторы движения уточняются от больших объектов к маленьким.

Начальная оценка движения является первой процедурой поиска векторов на стадии, при этом все блоки обрабатываются независимо друг от друга. После завершения начальной оценки каждому блоку на промежуточном кадре поставлен в соответствие вектор движения, минимизирующий ошибку сопоставления блоков (block matching) при градиентном спуске. Затем применяется дополнительный поиск для повышения гладкости полученного векторного поля. Процедура дополнительного поиска реализована как итеративный поиск, основанный на предсказании с учетом надежности векторов. Вектор считается надежным, если соответствующая ему билатеральная ошибка сопоставления мала (малая величина слагаемого E_1) и его значение коррелировано с соседями (малая величина слагаемого E_2) [42], т. е. вектор согласован с моделью истинного движения. Процесс предсказания оперирует с надежными векторами из множества соседей текущего блока на данном уровне иерархии и с надежными векторами на предыдущих уровнях иерархии. Эти векторы формируют множество кандидатов для интерполяции текущего блока. Кандидаты используются для инициализации градиентного спуска при поиске оптимального вектора для текущего блока. Следует отметить, что подобная процедура должна быть повторена несколько раз для того, чтобы исправить все ненадежные вектора. Таким образом, итеративность дополнительного поиска позволяет распространить влияние надежных векторов в векторном поле. Суммарное значение E в (2.6) уменьшается по следующим двум причинам:

- кандидаты для инициализации градиентного спуска берутся из «гладкого» множества надежных векторов соседних блоков;
- вектор для текущего блока замещается на результат градиентного спуска только в том случае, если ошибка сопоставления, полученная после поиска оптимального вектора с использованием векторов из множества кандидатов, не превышает начальную ошибку сопоставления.

Следует отметить, что начальная оценка движения на следующей стадии может привести к увеличению слагаемого E_2 в выражении (2.6). Как правило, это не влияет на стабильность векторного поля в том случае, когда оценка движения выполняется для больших блоков, так как они соответствуют смещению больших объектов на кадрах. Однако при оценке движения маленькими блоками данный эффект может привести к существенному увеличению «шума» в векторном поле, который возможно не удастся сгладить при дополнительном поиске. Для обработки таких ситуаций разработанный алгоритм оценки движения, по аналогии с [30], использует градиентный спуск, добавляющий пенальти при сравнении блоков. Пенальти – это константное слагаемое, добавляемое к ошибке сопоставления блоков для векторов, отклоняющихся от предсказанного значения. Значение пенальти инициализируется небольшим значением и растет с увеличением номера стадии, что позволяет избежать существенных отклонений от предсказанных значений векторов при оценке движения малыми блоками.

В следующих разделах приведено полное формальное описание разработанного алгоритма оценки движения.

2.5.2 Оценка движения

2.5.2.1 Формирование иерархической схемы

Каждый уровень иерархии инициализируется перед выполнением оценки движения. Для упрощения обозначений, как и при описании метода из модели DISCOVER, будем полагать, что выполняются следующие соотношения:

$$\frac{w}{2^{l_{max}}}, \frac{h}{2^{l_{max}}} \in \mathbb{N},$$

где w и h – ширина и высота кадров видеопоследовательности соответственно; l_{max} – индекс максимального уровня иерархии.

При инициализации уровня иерархии выполняются следующие операции:

1. подготовка сетки блоков на промежуточном кадре;
2. расширение базовых кадров.

Подготовка сетки блоков заключается в разбиении множества координат пикселей промежуточного кадра на подмножества непересекающихся блоков фиксированного размера. Для упрощения изложения будем считать, что используются квадратные блоки, и размер блока на уровне с номером l определяется как $2^l \times 2^l$.

Определим сетку блоков на уровне l как

$$\mathcal{G}^{(l)} = \{\mathbf{g}_{i,j}^{(l)} = (i,j)^T \mid i \in \mathcal{I}^{(l)}, j \in \mathcal{J}^{(l)}\},$$

где $\mathcal{I}^{(l)} = \{1, 2, \dots, h^{(l)}\}$, $\mathcal{J}^{(l)} = \{1, 2, \dots, w^{(l)}\}$, $h^{(l)} = \frac{h}{2^l}$, $w^{(l)} = \frac{w}{2^l}$ – высота и ширина сетки блоков соответственно. Отметим, что для $l = 0$ сетка блоков совпадает с сеткой пикселей. Множество координат пикселей, находящихся в блоке с индексом $(i, j)^T$ на уровне l , определим как

$$\mathcal{B}_{i,j}^{(l)} = \{(y,x)^T \in \mathcal{P} \mid y \in \mathcal{Y}_i^{(l)}, x \in \mathcal{X}_j^{(l)}\},$$

где $\mathcal{Y}_i^{(l)} = \{(i-1)2^l + 1, \dots, i2^l\}$; $\mathcal{X}_j^{(l)} = \{(j-1)2^l + 1, \dots, j2^l\}$. Следует отметить, что описанная процедура сопоставления координат пикселей блокам уменьшает размер блока (число пикселей, входящих в блок) с увеличением номера уровня иерархии. При этом каждый блок разбивается на четыре блока меньшего размера на следующем уровне. Индексы блоков, включающих одни и те же координаты пикселей на смежных уровнях иерархии, рассчитаются по формуле

$$\mathbf{g}_{i,j}^{(l)} \leftrightarrow \{\mathbf{g}_{2i-1,2j-1}^{(l+1)}, \mathbf{g}_{2i-1,2j}^{(l+1)}, \mathbf{g}_{2i,2j-1}^{(l+1)}, \mathbf{g}_{2i,2j}^{(l+1)}\}. \quad (2.8)$$

Множество координат блоков на вышележащих уровнях для блока $\mathbf{g}_{i,j}^{(l)}$ вычисляется по формуле

$$\mathcal{H}_{i,j}^{(l)} = \left\{ \mathbf{g}_{k,l}^{(m)} \mid m = \overline{1, l-1}, \mathbf{g}_{k,l}^{(m)} = \left(\left[\frac{i}{d} \right], \left[\frac{j}{d} \right] \right)^T, d = \frac{2^l}{2^m} \right\}. \quad (2.9)$$

Приведенные соотношения завершают описание процедуры подготовки сетки блоков на промежуточном кадре. Вторая процедура инициализации уровня иерархии применяется только к базовым кадрам. Она заключается в расширении этих кадров за счет добавления дополнительных m_b столбцов и m_b строк к верхней, нижней, левой и правой границам кадра, m_b – параметр алгоритма. Это делается

для улучшения поиска на краях изображений. Проведенные эксперименты показывают, что в среднем лучшие результаты достигаются с использованием $m_b = 4$. Расширение кадров осуществляется размножением пикселей с соответствующей границы кадра. Обозначим множество координат пикселей в расширенном базовом кадре на уровне l через $\mathcal{P}^{(l)}$.

2.5.2.2 Начальная оценка движения.

Задача начальной оценки движения заключается в поиске билатерального вектора движения для каждого блока на промежуточном кадре. Все блоки на данном этапе обрабатываются независимо друг от друга. В качестве процедуры поиска вектора движения в разработанном алгоритме используется градиентный спуск с штрафа, который можно формально описать как последовательность следующих шагов (обновлений):

$$\mathbf{v}^{(k)}(\mathbf{g}_{i,j}^{(l)}) = \arg \min_{\mathbf{c} \in \mathcal{C}_{i,j}^{(k)}} (E(\mathbf{c}, \mathbf{g}_{i,j}^{(l)})),$$

где через $k = 1, 2, \dots, k_{max}$ обозначен номер шага; $\mathbf{v}^{(k)}(\mathbf{g}_{i,j}^{(l)})$ – вектор движения для блока $\mathbf{g}_{i,j}^{(l)}$; l – номер текущего уровня иерархии; \mathbf{c} – вектор-кандидат; $\mathcal{C}_{i,j}^{(k)}$ – множество векторов-кандидатов, определенное как

$$\mathcal{C}_{i,j}^{(k)} = \{\mathbf{c} | \mathbf{c} = \mathbf{c}^{(k-1)}(\mathbf{g}_{i,j}^{(l)}) + \mathbf{u}\},$$

где $\mathbf{u} \in \mathcal{U}$ – векторы обновления, $\mathcal{U} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \pm 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix} \right\}$. Начальный вектор для градиентного спуска определяется как

$$\mathbf{v}^{(0)}(\mathbf{g}_{i,j}^{(l)}) = \mathbf{v} \left(\left\lceil \frac{\mathbf{g}_{i,j}^{(l-1)}}{2} \right\rceil \right),$$

где $\mathbf{v}(\mathbf{g}_{i,j}^{(l-1)})$ – вектор с предыдущего уровня иерархии (для первого уровня используется нулевое смещение). $E(\cdot, \cdot)$ – функция, описывающая ошибку сопоставления блока. В данной диссертационной работе предлагается использовать следу-

ющую функцию, основанную на сумме абсолютных разностей (Sum of Absolute Differences – SAD):

$$E(\mathbf{c}, \mathbf{g}_{i,j}^{(l)}) = \left[\frac{1}{|\mathcal{B}_{i,j}^{(l)}|} \sum_{\mathbf{p} \in \mathcal{B}_{i,j}^{(l)}} \text{WD}(\mathbf{p}, \mathbf{c}) \right] + \beta^{(l)} \|\mathbf{c} - \mathbf{v}^{(k-1)}(\mathbf{g}_{i,j}^{(l)})\|,$$

где β^h – параметр алгоритма, используемый для добавления штрафа к ошибке сопоставления для векторов, отличающихся от предсказанного значения и

$$\text{WD}(\mathbf{p}, \mathbf{c}) = \begin{cases} \langle \mathbf{w}, D(\mathbf{p}, \mathbf{c}) \rangle, & \text{если } \mathbf{p} \pm \mathbf{c} \in \mathcal{P}^{(l)} \\ \infty, & \text{иначе} \end{cases},$$

где через $\langle \cdot, \cdot \rangle$ обозначено скалярное произведение векторов и

$$D(\mathbf{p}, \mathbf{c}) = |\mathbf{F}_f(\mathbf{p} - \mathbf{c}) - \mathbf{F}_p(\mathbf{p} + \mathbf{c})|.$$

Значение коэффициента $\mathbf{w} = (1, 2, 2)$ (аналогично работе [43]). Градиентный спуск выполняется для каждого блока в сетке независимо. Поиск оптимального вектора для блока заканчивается тогда, когда либо достигается максимальное количество шагов, либо вектор движения, рассчитанный на текущем шаге, совпадает с результатом предыдущего шага, формально:

$$GSStop = (k = k_{max}) \text{ или } (\mathbf{v}^{(k)}(\mathbf{g}_{i,j}^{(l)}) = \mathbf{v}^{(k-1)}(\mathbf{g}_{i,j}^{(l)})).$$

Обозначим описанную процедуру градиентного спуска для блока, находящегося в сетке блоков на позиции $\mathbf{g}_{i,j}^{(l)}$, через $GS(\mathbf{g}_{i,j}^{(l)})$. Выходом $GS(\mathbf{g}_{i,j}^{(l)})$ является вектор движения $\mathbf{V}(\mathbf{g}_{i,j}^{(l)})$, оптимальный для критерия ошибки сопоставления блока $E(\mathbf{v}(\mathbf{g}_{i,j}^{(l)}), \mathbf{g}_{i,j}^{(l)})$. Для упрощения обозначений далее будем сокращать $E(\mathbf{v}(\mathbf{g}_{i,j}^{(h)}), \mathbf{g}_{i,j}^{(l)})$ как $e_{i,j}^{(l)}$, $\mathbf{v}(\mathbf{g}_{i,j}^{(l)})$ как $\mathbf{v}_{i,j}^{(l)}$.

Результатом процедуры начальной оценки движения являются векторное поле $\mathbf{V}^{(l)} = \{\mathbf{v}_{i,j}^{(l)}\}$ и множество ошибок сопоставления блоков $\mathbf{E}^{(h)} = \{e_{i,j}^{(h)}\}$, $i = \overline{1, h^{(l)}}, j = \overline{1, w^{(l)}}$.

2.5.2.3 Итеративный дополнительный поиск.

Дополнительный поиск реализован как последовательная итеративная процедура, на каждой итерации которой осуществляются три действия.

1. Расчет надежностей векторов движения.
2. Дополнительная оценка движения.
3. Анализ критериев завершения дополнительного поиска.

На каждой итерации n вычисляется уточненная версия $\mathbf{V}_n^{(l)}$ векторного поля с предыдущей итерации $\mathbf{V}_{n-1}^{(l)}$ (или поля с начальной оценки движения, если итерация первая).

Расчет надежностей векторов движения используется для сопоставления каждого вектора с некоторой меткой класса, показывающей «качество» вектора относительно его ошибки сопоставления и локальной гладкости векторного поля. В разработанном алгоритме используются метки классов 0, 1, 2 и 3, где класс 3 соответствует наиболее надежному вектору.

Метод расчета надежностей основан на подходе, описанном в работе [44], единственное отличие заключается в том, что в разработанном алгоритме предлагается всегда использовать метку класса 0 для блоков, находящихся на границе кадра. Таким образом, расчет метки надежности для вектора $\mathbf{v}_{i,j}^{(l)}$ осуществляется по следующему правилу:

$$r_{i,j}^{(h)} = \begin{cases} 0, & \text{если } i \in \{1, h^{(l)}\} \text{ или } j \in \{1, w^{(l)}\} \\ 1, & \text{если } e_{i,j}^{(l)} > t_1 \\ 2, & \text{если } s_{i,j}^{(l)} > sa_{i,j}^{(l)} \text{ и } s_{i,j}^{(l)} > t_2 \\ 3, & \text{иначе} \end{cases} .$$

где $s_{i,j}^{(h)}$ и $sa_{i,j}^{(h)}$ являются характеристиками локальной гладкости векторного поля и рассчитываются в соответствии с работой [45].

Дополнительная оценка движения выполняется для каждого блока на промежуточном кадре. Для блока $\mathbf{g}_{i,j}^{(l)}$ строится множество векторов-кандидатов:

$$\mathcal{CS}_{i,j}^{(l)} = \left\{ \mathbf{v}_{k,m}^{(l)} \mid r_{k,m}^{(l)} \geq r_{i,j}^{(l)}, \mathbf{g}_{k,m}^{(l)} \in \mathcal{N}(\mathbf{g}_{i,j}^{(l)}) \right\} \cup \left\{ \mathbf{v}_{k,m}^{(p)} \mid r_{k,m}^{(p)} \geq r_{i,j}^{(l)}, \mathbf{g}_{k,m}^{(p)} \in \mathcal{H}_{i,j}^{(l)} \right\},$$

где $\mathcal{N}(\mathbf{g}_{i,j}^{(l)})$ – множество блоков, соседних с $\mathbf{g}_{i,j}^{(l)}$:

$$\mathcal{N}(\mathbf{g}_{i,j}^{(l)}) = \{\mathbf{g}_{k,m}^{(l)} \in \mathcal{G}^{(l)} \mid \|\mathbf{g}_{k,m}^{(l)} - \mathbf{g}_{i,j}^{(l)}\|_2 \leq t, t > 0\},$$

где t – некоторый заранее заданный порог, определяющий размер множества соседних блоков. В данной работе используется значение $t = \sqrt{2}$.

$\mathcal{CS}_{i,j}^{(l)}$ включает надежные векторы из множества соседей блока $\mathbf{g}_{i,j}^{(l)}$ на уровне l , а также надежные векторы, соответствующие данному блоку, на предыдущих уровнях иерархии.

Затем для блока $\mathbf{g}_{i,j}^{(l)}$ выполняется процедура градиентного спуска с использованием векторов из $\mathcal{CS}_{i,j}^{(l)}$ в качестве начальных смещений. Вектор с минимальной ошибкой сопоставления выбирается в качестве оптимального для блока $\mathbf{g}_{i,j}^{(l)}$. Если есть несколько векторов, дающих одинаковое значение минимума при градиентном спуске, решение принимается на основе анализа локальной гладкости: вектор, обеспечивающий наиболее гладкое поле, выбирается как оптимальный. После выбора вектора происходит обновление надежности для блока $\mathbf{g}_{i,j}^{(l)}$.

Расчет надежностей и дополнительный поиск повторяются несколько раз, до тех пор, пока либо не будет достигнуто максимальное число итераций, либо векторное поле не будет сильно изменяться между итерациями:

$$AS = (n = n_{max}^{(l)}) \text{ или } \left(\sum_{i=1}^{h^{(l)}} \sum_{j=1}^{w^{(l)}} \mathbb{1}(d_{i,j} > t_{s1}^{(l)} < t_s^{(l)}) \right),$$

где $d_{i,j} \in |\mathbf{V}_{\mathbf{n}}^{(h)} - \mathbf{V}_{\mathbf{n}-1}^{(h)}|$ и $\mathbb{1}(\cdot)$ – индикаторная функция, определенная как

$$\mathbb{1}(X) = \begin{cases} 1, & \text{если } X \text{ истинно} \\ 0, & \text{иначе} \end{cases}.$$

Результатом дополнительного поиска является уточненное векторное поле $\mathbf{V}^{(l)}$, которое используется для инициализации векторов движения на следующем уровне иерархии.

2.5.3 Компенсация движения

Полученное в результате оценки движения векторное поле $\mathbf{V} = \{\mathbf{v}_{k,m}\}$, $k = \overline{1, \frac{h}{2^{l_{max}}}}$, $m = \overline{1, \frac{w}{2^{l_{max}}}}$ далее используется в процедуре компенсации движения для аппроксимации промежуточного кадра.

Описанный ранее в данном разделе алгоритм оценки движения принадлежит к классу блоковых методов. Использование блоковой оценки движения может приводить к наличию блочной структуры на промежуточном кадре. Одним из распространенных подходов, позволяющим уменьшить подобные визуальные проявления, является *компенсация движения с перекрытиями* (ОВМС, Overlapped Block Motion Compensation) [46]. При таком способе компенсации результат интерполяции для каждого пикселя рассчитывается как взвешенная сумма, учитывающая векторы движения соседних пикселей. Подобную процедуру можно рассматривать как сглаживание кадра с учетом движения, в результате которого уменьшается визуальное проявление границ между блоками.

Рассмотрим применение ОВМС для интерполяции блока, верхний левый угол которого находится на позиции $\mathbf{p} = (y, x)^T$ в сетке пикселей на промежуточном кадре \mathbf{F}_i . Пусть данному блоку в результате оценки движения поставлен в соответствие вектор $\mathbf{v}_{k,l}$, где k и l – координаты блока в сетке блоков. Тогда расчет значений интерполированных пикселей осуществляется в соответствии со следующим правилом:

$$\mathbf{B} = \sum_{(k',l')^T \in \mathcal{N}((k,l)^T)} \frac{\mathbf{W}_{k-k',l-l'}}{2} \otimes [\mathbf{F}_p(\mathbf{p} + \mathbf{v}_{k',l'}) + \mathbf{F}_f(\mathbf{p} - \mathbf{v}_{k',l'})],$$

где \mathbf{B} – блок пикселей, который затем помещается в соответствующую позицию блока на промежуточном кадре, $\mathcal{N}((k,l)^T) = \{(k,l)^T, (k+1,l)^T, (k-1,l)^T, (k,l+1)^T, (k,l-1)^T\}$ – множество позиций векторов в сетке блоков, участвующих в усреднении при интерполяции текущего блока, $\mathbf{W}_{0,0}$, $\mathbf{W}_{0,1}$, $\mathbf{W}_{1,0}$, $\mathbf{W}_{0,-1}$, $\mathbf{W}_{-1,0}$ – матрицы весовых коэффициентов, размер матриц совпадает с размерами блоков, причем

$$\mathbf{W}_{0,0} + \mathbf{W}_{0,1} + \mathbf{W}_{1,0} + \mathbf{W}_{0,-1} + \mathbf{W}_{-1,0} = \mathbf{J},$$

где \mathbf{J} - матрица, каждый элемент которой равен единице. Пример матриц для блока размером 4×4 пикселя:

$$\mathbf{W}_{0,0} = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix};$$

$$\mathbf{W}_{-1,0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}; \mathbf{W}_{1,0} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 \end{pmatrix};$$

$$\mathbf{W}_{0,1} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \mathbf{W}_{0,-1} = \begin{pmatrix} 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

В результате компенсации движения получается аппроксимирующий кадр, который затем обрабатывается на стороне декодера так же, как оригинальный промежуточный кадр на кодере. Вопросы, связанные с оценкой эффективности описанной схемы, рассматриваются в разделе 4.

2.6 Выводы по разделу

В данном разделе была рассмотрена задача генерации дополнительной информации декодера в системах распределенного кодирования видеоданных. Показано, что ключевым блоком модуля генерации дополнительной информации является блок межкадрового предсказания, выполняющий аппроксимацию промежуточного кадра на стороне декодера. Приведена классификация методов аппроксимации промежуточных кадров и разработан новый метод, использующий при интерполяции промежуточного кадра алгоритм оценки истинного движения объектов в видеопоследовательности.

Основные результаты раздела можно сформулировать следующим образом:

- проанализирован алгоритм генерации дополнительной информации в модели распределенного кодирования DISCOVER и выделена ключевая операция, связанная с оценкой движения;
- проанализирована модель истинного движения объектов в видеопоследовательности;
- разработан метод межкадрового предсказания для процедуры генерации дополнительной информации, основанный на новом эвристическом алгоритме поиска истинного движения.

3 ОЦЕНКА ПАРАМЕТРОВ ОШИБОК В ВИРТУАЛЬНОМ КАНАЛЕ

3.1 Вводные замечания

Интерпретация процесса сжатия видеоданных в распределенных видеокодеках с использованием понятия виртуального канала была рассмотрена в разделе 1. На качественном уровне была обоснована важность задачи оценки параметров ошибок, возникающих в виртуальном канале. В данном разделе рассматриваются вопросы, связанные с оценкой параметров ошибок межкадрового предсказания на практике. Отметим, что в англоязычной литературе эта задача также иногда называется *моделированием виртуального канала* (Virtual Dependency Channel Modeling) или *моделированием корреляционного шума* (Correlation Noise Modeling).

Рассмотрим формирование ошибок межкадрового предсказания в рамках распределенного видеокодирования в спектральной области с использованием концепции Стэнфорд. Обозначим результат предсказания промежуточного кадра на стороне декодера через \mathbf{F}_a . Следует отметить, что т. к. кодек DISCOVER обрабатывает только однокомпонентные видеопоследовательности, здесь и далее будем считать, что \mathbf{F}_a содержит яркостную составляющую. Декодер, аналогично кодеру, выполняет блоковое спектральное преобразование и квантование для пикселей кадра \mathbf{F}_a , в результате чего для каждой группы спектральных коэффициентов формируется набор индексов квантов $\hat{\mathbf{q}}^{(b)} = (\hat{q}_1^{(b)}, \hat{q}_2^{(b)}, \dots, \hat{q}_m^{(b)})$, где b – номер группы, m – число спектральных коэффициентов в группе. Отметим, что кодер, выполнив аналогичные операции, получает набор индексов квантов $\mathbf{q}^{(b)} = (q_1^{(b)}, q_2^{(b)}, \dots, q_m^{(b)})$ для оригинального промежуточного кадра \mathbf{F}_i . Так как результат предсказания является некоторой аппроксимацией промежуточного кадра, $\mathbf{q}^{(b)}$ и $\hat{\mathbf{q}}^{(b)}$ можно рассматривать как выходы зависимых источников. Таким образом, $\hat{\mathbf{q}}^{(b)}$ является дополнительной информацией декодера и ошибка межкадрового предсказания в спектральной области $\mathbf{n}^{(b)} = (n_1^{(b)}, n_2^{(b)}, \dots, n_m^{(b)})$ для каждой группы определяется как

$$\mathbf{n}^{(b)} = \mathbf{q}^{(b)} - \hat{\mathbf{q}}^{(b)}.$$

Наличие различий между квантованными спектральными коэффициентами на стороне кодера и декодера, объясняется в первую очередь несовершенством алгоритмов аппроксимации промежуточного кадра. Отметим также, что так как оригинал кадра доступен только кодеру, декодер не может в явном виде рассчитать $\mathbf{n}^{(b)}$.

Учитывая вышесказанное, задачу декодера при оценке параметров ошибок межкадрового предсказания можно сформулировать следующим образом: не зная точные значения ошибок $n_i^{(b)}$, оценить для каждого квантованного спектрального коэффициента $\hat{q}_i^{(b)}$ набор некоторых характеристик, описывающих шум, возникающий в виртуальном канале при «передаче» данного коэффициента. Данные характеристики далее используются декодером для формирования мягкого входа помехоустойчивого декодера в кодеке Слепяна-Вулфа. В рамках приведенной задачи можно выделить следующие подзадачи:

- аппроксимация шума $\mathbf{n}^{(b)}$ на стороне декодера;
- определение параметров ошибки для каждого квантованного спектрального коэффициента $\hat{q}_i^{(b)}$.

Рассмотрению этих вопросов и посвящен данный раздел. В подразделе 3.2 приводится набор общепринятых допущений об ошибках, возникающих в виртуальном канале, и рассматривается алгоритм оценки параметров ошибок межкадрового предсказания, используемый в кодеке DISCOVER. Указаны основные недостатки данного алгоритма и предложена модифицированный алгоритм, лучше выполняющий учет особенностей ошибок межкадрового предсказания. В подразделе 3.4 предложен новый, расширенный набор допущений, на основе которого разработана обобщенная модель виртуального канала, основанная на представлении ошибок как случайного процесса, задаваемого Марковской сетью со скрытыми состояниями. Для этой модели показано, что задачу оценки параметров ошибок можно свести к задаче согласованной маркировки (раскрашивания) вершин некоторого графа, причем решение этой задачи соответствует оценке по методу максимального правдоподобия. Показано, что в рамках введенной модели результат модифицированного алгоритма лучше, чем у базового алгоритма. Основные результаты данного раздела опубликованы в работах [47], [48], [49].

3.2 Существующие методы оценки параметров ошибок межкадрового предсказания для систем распределенного кодирования

3.2.1 Описание общепринятых допущений о виртуальном канале

В ранних работах по распределенному кодированию источников видеoinформации ошибка межкадрового предсказания рассматривалась как аддитивный случайный стационарный процесс с нулевым математическим ожиданием [50]. Это допущение известно как *стационарный виртуальный канал*. Термин «стационарность» в данном контексте означает, что вероятностные характеристики ошибок не меняются в рамках одной группы спектральных коэффициентов для одного кадра. Одним из существенных преимуществ такой модели шума является простота её реализации на практике. Однако, в силу того, что алгоритмы аппроксимации промежуточных кадров в распределенном видеокодировании основаны на различных методах временной интерполяции/экстраполяции, вероятностные характеристики ошибок межкадрового предсказания на практике могут существенно различаться. Например, интенсивность ошибок в областях на кадрах, где появляются новые объекты или наблюдается сложное движение существующих объектов, существенно превышает интенсивность ошибок в областях с малоподвижным фоном (рисунок 3.1). Таким образом, модель, основанная на допущении о стационарном виртуальном канале, плохо подходит для описания ошибок межкадрового предсказания. В связи с этим в современных работах по распределенному видеокодированию рассматриваются модели шума, основанные на допущении о *нестационарном виртуальном канале*.

Рассмотрим набор общепринятых допущений о нестационарном виртуальном канале [51].

Допущение 3.1. Для каждой группы спектральных коэффициентов декодер может выполнить аппроксимацию ошибок $\hat{\mathbf{n}}^{(b)} = (\hat{n}_1^{(b)}, \hat{n}_2^{(b)}, \dots, \hat{n}_m^{(b)})$, причем $\hat{n}_i^{(b)} = n_i^{(b)} + e_i^{(b)}$, где $e_i^{(b)}$ – случайное искажение, имеющее некоторый закон распределения с нулевым математическим ожиданием и конечной дисперсией.

Будем называть $\hat{\mathbf{n}}^{(b)}$ *аппроксимацией* ошибки межкадрового предсказания на стороне декодера.

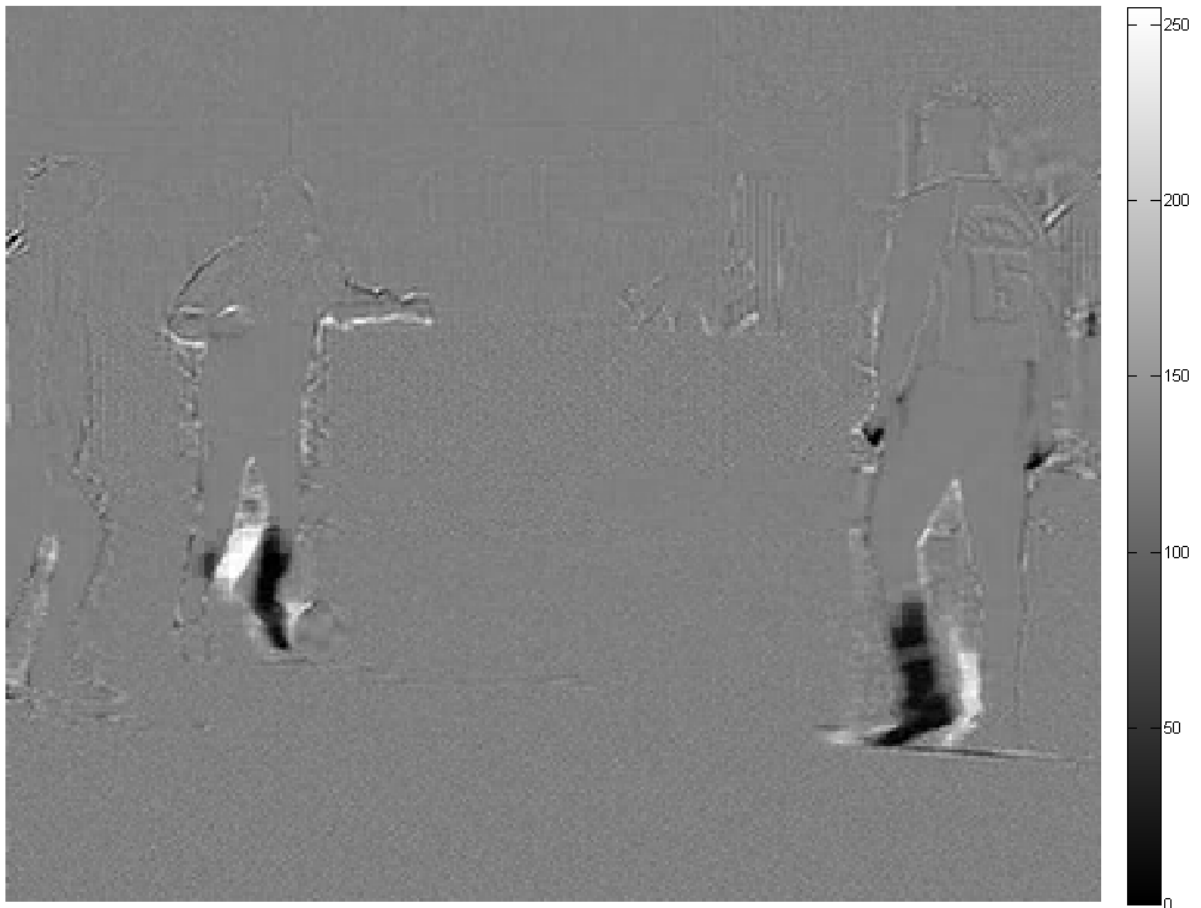


Рисунок 3.1 — Пример ошибок межкадрового предсказания для кадра из последовательности SOCCER (значение 128 соответствует нулевой ошибке)

Допущение 3.2. Виртуальный канал является каналом с аддитивным нестационарным Лапласовым шумом, т. е. каждое значение $n_i^{(b)}$ является реализацией случайной величины $N_i^{(b)}$, имеющей закон распределения Лапласа с нулевым математическим ожиданием:

$$N_i^{(b)} \sim \text{Lap}(x|\alpha_i^{(b)}, 0),$$

где

$$\text{Lap}(x|\alpha_i^{(b)}, \mu_i^{(b)}) = \frac{\alpha_i^{(b)}}{2} e^{-\alpha_i^{(b)}|x-\mu_i^{(b)}|}.$$

Отметим, что допущение 3.2 позволяет рассматривать последовательность ошибок $n_i^{(b)}$ как реализацию множества в общем случае зависимых случайных величин, хотя понятие зависимости при этом никак не формализуется. Единственное, что утверждается, это то, что условная функция плотности вероятности для каждой ошибки является Лапласовой.

С учетом приведенных допущений, задача оценки параметров ошибок межкадрового предсказания заключается в оценке для каждого квантованного спектрального коэффициента параметра масштаба распределения Лапласа, описывающего интенсивность искажающего воздействия, «возникающего» в виртуальном канале при передаче данного спектрального коэффициента. Для решения данной задачи в модели Discover предлагается эвристический алгоритм, описанный в следующем подразделе.

3.2.2 Описание базового алгоритма оценки параметров ошибок межкадрового предсказания

Рассмотрим алгоритм моделирования виртуального канала, разработанный в рамках проекта DISCOVER [52]. Здесь и далее будем придерживаться обозначений, введенных в разделе 2.

Процедура оценки параметров ошибок межкадрового предсказания в кодеке DISCOVER представляет собой следующую последовательность действий.

1. Расчет аппроксимации ошибки в пространственной области. В базовом алгоритме ошибку аппроксимации в пространственной области предлагается рассчитывать как

$$\mathbf{R}(\mathbf{p}) = \frac{1}{2} [\mathbf{F}_p(\mathbf{p} - \mathbf{V}(\mathbf{p})) + \mathbf{F}_f(\mathbf{p}_i + \mathbf{V}(\mathbf{p}))], \forall \mathbf{p} \in \mathcal{P},$$

где \mathbf{F}_p , \mathbf{F}_f – опорные кадры, находящиеся во времени соответственно перед и после промежуточного кадра, $\mathbf{V}(\mathbf{p})$ – билатеральное векторное поле, полученное в результате процедуры оценки движения (подраздел 2.3), \mathcal{P} – множество позиций пикселей.

2. Расчет оценки шума $\hat{\mathbf{n}}^{(b)}$. Разностный кадр \mathbf{R} подвергается блоковому дискретному косинусному преобразованию с размером блока 4×4 пикселя. Затем, по аналогии с обработкой аппроксимирующего кадра \mathbf{F}_a , полученные спектральные коэффициенты собираются в группы $\mathbf{n}^{(b)}$ (детальное описание процесса приведено в подразделе 3.1). Делается допущение, что рассчитанные таким образом данные являются аппроксимацией ошибок межкадрового предсказания на стороне декодера.

3. Расчет выборочного среднего для абсолютных значений оцененных ошибок:

$$\forall b \in \{1, 2, \dots, B\} : \hat{\mu}^{(b)} = \frac{1}{m} \sum_{i=1}^m |\hat{n}_i^{(b)}|.$$

4. Расчет для каждого оцененного значения шума квадратичного отклонения от выборочного среднего

$$\forall b \in \{1, 2, \dots, B\}, \forall i \in \{1, 2, \dots, m\} : d_i^{(b)} = \left(|\hat{n}_i^{(b)}| - \hat{\mu}^{(b)} \right)^2.$$

5. Расчет выборочной дисперсии для абсолютных значений оцененных ошибок:

$$\forall b \in \{1, 2, \dots, B\} : \left(\hat{\sigma}^{(b)} \right)^2 = \frac{1}{m} \sum_{i=1}^m d_i^{(b)}.$$

6. Оценка параметра распределения Лапласа для группы:

$$\forall b \in \{1, 2, \dots, B\} : \hat{\alpha}^{(b)} = \frac{\sqrt{2}}{\hat{\sigma}^{(b)}}$$

7. Оценка параметра распределения Лапласа для каждого коэффициента:

$$\forall b \in \{1, 2, \dots, B\}, \forall i \in \{1, 2, \dots, m\} : \hat{\alpha}_i^{(b)} = \begin{cases} \hat{\alpha}^{(b)}, & \text{если } d_i^{(b)} \leq \sigma^{(b)} \\ \frac{\sqrt{2}}{d_i^{(b)}}, & \text{иначе} \end{cases}.$$

В дальнейшем будем называть данный алгоритм *базовым алгоритмом* оценки параметров ошибок межкадрового предсказания. Отметим, что в базовом алгоритме нестационарность ошибок в виртуальном канале учитывается с помощью адаптации оценки параметра распределения Лапласа для каждого спектрального коэффициента с учетом статистики всех коэффициентов в группе.

3.2.3 Недостатки базового алгоритма

Рассмотрим пример, в рамках которого продемонстрируем работу базового алгоритма, а также выявим ряд его потенциальных недостатков. Данный пример будет использован в следующем подразделе для демонстрации особенностей

предложенной модели шума. Рассмотрим процесс восстановления промежуточного кадра на стороне декодера. Декодер, восстановив пару смежных опорных кадров, выполняет по ним аппроксимацию промежуточного кадра (например, с использованием алгоритма, описанного в подразделе 2.3 и рассчитывает оценку ошибки межкадрового предсказания (например, с использованием подхода, описанного в подразделе 3.2). Результат данных операций представлен на рисунке 3.2. Также, для того, чтобы продемонстрировать тот факт, что результат оценки ошибки межкадрового предсказания коррелирует с реальной ошибкой, на данном рисунке приведена разность между восстановленным промежуточным кадром на стороне кодера и его аппроксимацией на стороне декодера.

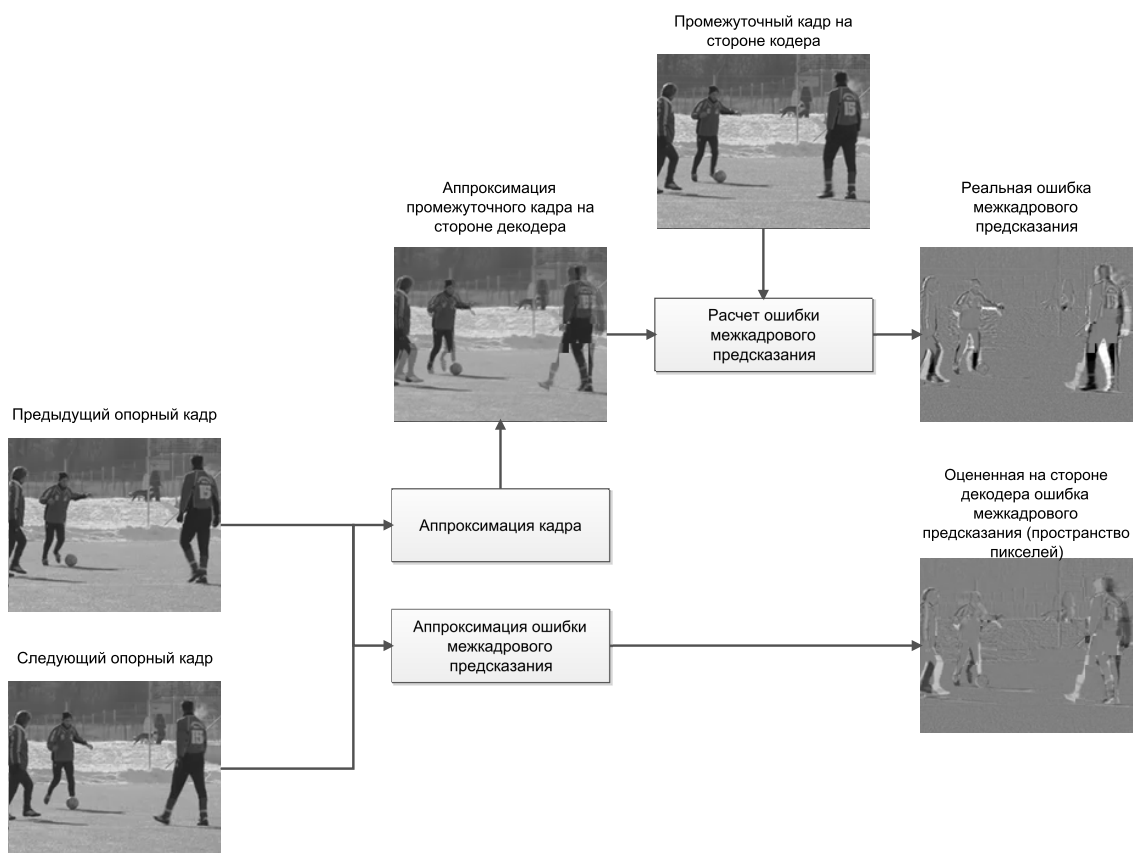


Рисунок 3.2 — Пример оцененной на стороне декодера ошибки межкадрового предсказания в пространстве пикселей

Аппроксимация ошибки в спектральной области для старшего коэффициента, рассчитанная по оцененной на стороне декодера ошибке межкадрового предсказания, представлена на рисунке 3.3. Видно, что особенность шума, связанная с группировкой ошибок в пространстве пикселей, сохранилась и в спектральной области. Отметим, что т. к. базовый алгоритм работает только с абсолютными значениями коэффициентов, на рисунке приведены абсолютные величины оце-

ненных ошибок. Для того, чтобы продемонстрировать работу базового алгоритма, рассмотрим процесс оценки параметров ошибок для части спектральных коэффициентов, отмеченных на рисунке 3.3. В целях демонстрации будем считать, что в аппроксимации шума для данной группы частот содержатся только эти коэффициенты, т. е.

$$\hat{\mathbf{n}}^{(1)} = \begin{pmatrix} 2 & 1 & 3 & 5 \\ 0 & 0 & 4 & 83 \\ 7 & 24 & 157 & 208 \\ 1 & 1 & 30 & 132 \end{pmatrix}. \quad (3.1)$$

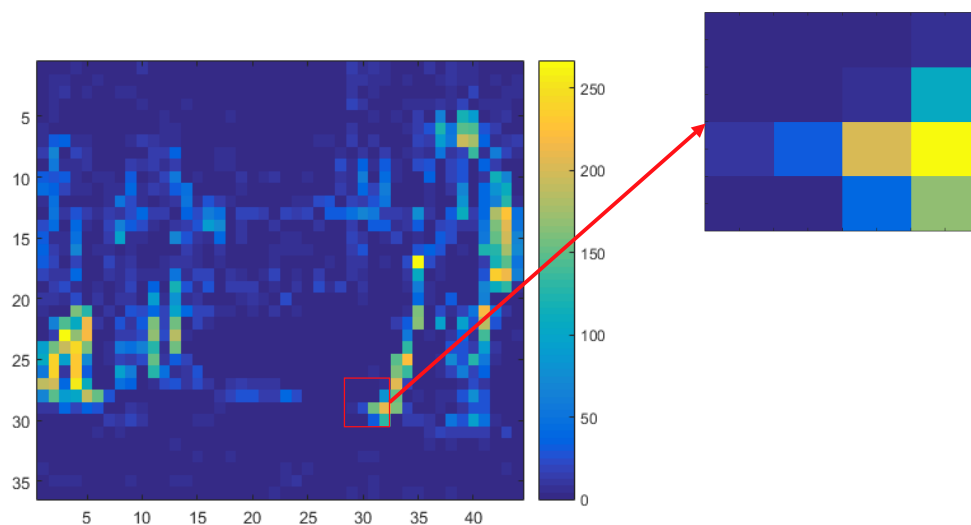


Рисунок 3.3 — Пример аппроксимации ошибок межкадрового предсказания в спектральной области (коэффициент DC).

Приведем шаги алгоритма для данных значений. Т. к. шаги 1 и 2 связаны с получением аппроксимации шума, начнем пример с шага 3.

- Шаг 3. Расчет выборочного среднего для абсолютных значений оцененных ошибок: $\hat{\mu}^{(1)} = 41.125$.
- Шаг 4. Расчет для каждого оцененного значения шума квадратичного отклонения от выборочного среднего:

$$\begin{pmatrix} 1530.8 & 1610 & 1453.5 & 1305 \\ 1691.3 & 1691.3 & 1378.3 & 1753.5 \\ 1164.5 & 293.27 & \mathbf{13427} & \mathbf{27847} \\ 1610 & 1610 & 123.77 & \mathbf{8258.3} \end{pmatrix}. \quad (3.2)$$

- Шаг 5. Расчет выборочной дисперсии для абсолютных значений оцененных ошибок: $(\hat{\sigma}^{(1)})^2 = 4449.85$.
- Шаг 6. Оценка параметра распределения Лапласа для группы: $\hat{\alpha}^{(b)} = 0.0212$.
- Шаг 7. Оценка параметра распределения Лапласа для каждого коэффициента:

$$\begin{pmatrix} 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & \mathbf{0.012} & \mathbf{0.008} \\ 0.021 & 0.021 & 0.021 & \mathbf{0.016} \end{pmatrix}. \quad (3.3)$$

В получившейся матрице полужирным шрифтом выделены те позиции спектральных коэффициентов, которые были помечены ненадежными по сравнению с остальными коэффициентами в группе. Функции плотности вероятности для распределения Лапласа с различными значениями параметров масштаба приведены на рисунке 3.4. Из графиков видно, что чем выше надежность коэффициентов, тем более острый пик имеет распределение Лапласа. Однако следует отметить тот факт, что несмотря на существенную разницу в оцененных значениях ошибок, значения надежностей различаются не столь значительно. Данный факт объясняется недостаточным учетом корреляции между ошибками, оцененными для различных спектральных коэффициентов. Кроме того, недостатком базового алгоритма является то, что не учитываются возможные ошибки аппроксимации шума, когда большие значения ошибок оценены для корректно аппроксимированных регионов кадра и наоборот.

Для того, чтобы обойти указанные недостатки в подразделе 3.3 приводится описание модифицированного алгоритма оценки параметров ошибок межкадрового предсказания.

3.3 Предложенный модифицированный алгоритм оценки параметров ошибок межкадрового предсказания

Основная идея предложенного модифицированного алгоритма оценки параметров ошибок межкадрового предсказания заключается в выполнении разби-

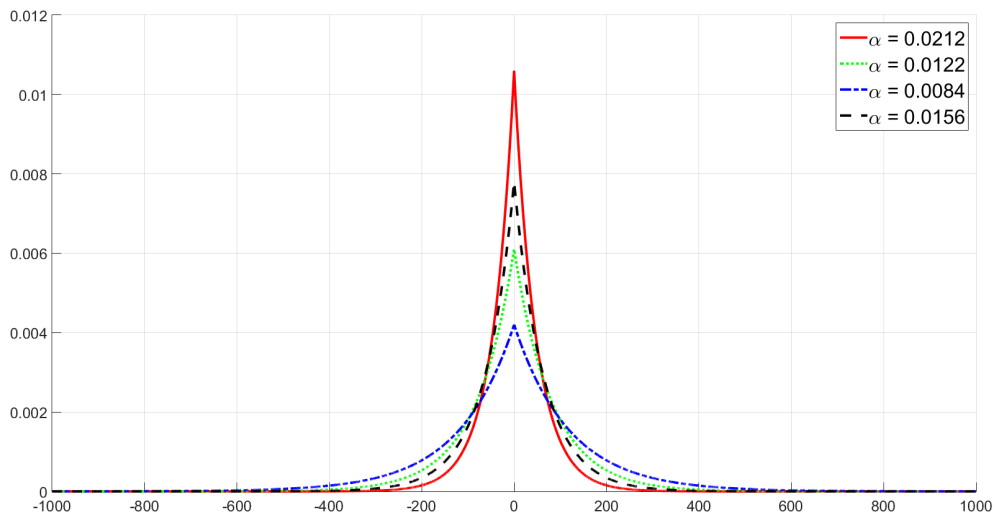


Рисунок 3.4 — Функции плотности вероятности для различных значений параметра масштаба.

ения множества $\hat{\mathbf{n}}^{(b)}$ на подмножества и последующей оценке параметров ошибок в каждом подмножестве независимо. Обозначим число подмножеств через $k^{(b)}$. Для выполнения разбиения предлагается использовать подход, основанный на EM-алгоритме [53]. Основная идея EM-алгоритма заключается в следующем. Для каждой случайной величины в выборке вводится скрытая переменная, значение которой может быть вычислено, если известны параметры распределений в смеси. При этом скрытые переменные вводятся таким образом, что, если они известны, можно выполнить пересчет параметров распределений в смеси с использованием принципа максимума правдоподобия.

Приведем описание EM-алгоритма для задачи разбиения множества ошибок на $k^{(b)}$ подмножеств. Будем считать, что смесь имеет вид

$$f(x|\mathcal{D}^{(b)}) = \sum_{i=1}^{k^{(b)}} w_i^{(b)} \text{Lap}(x|\alpha_i^{(b)}, 0). \quad (3.4)$$

Обозначим множество скрытых переменных через $\mathcal{A}^{(b)}$. EM-алгоритм состоит из итеративного повторения двух шагов.

- E-шаг. На этом шаге выполняется расчет ожидаемого значения скрытых случайных величин $A_i^{(b)} \in \mathcal{A}^{(b)}$. Обозначим $\Pr[A_i^{(b)} = a | N_i^{(b)} = \hat{n}_i^{(b)}]$ через

$g_{i,a}$. Тогда на данном шаге выполняются следующие действия:

$$\forall a \in \{1, 2, \dots, k^{(b)}\}, \forall i \in \{1, 2, \dots, m\} :$$

$$g_{i,a} = \frac{\Pr\{A_i^{(b)} = a\} \Pr\{N_i^{(b)} = \hat{n}_i^{(b)} | A_i^{(b)} = a\}}{\sum_{k=1}^{k^{(b)}} \Pr\{A_i^{(b)} = k\} \Pr\{N_i^{(b)} = \hat{n}_i^{(b)} | A_i^{(b)} = k\}} = \frac{w_a^{(b)} \mathbf{p}(\hat{n}_i^{(b)} | a)}{\sum_{k=1}^{k^{(b)}} w_k^{(b)} \mathbf{p}(\hat{n}_i^{(b)} | k)}, \quad (3.5)$$

где $w_a^{(b)} = \Pr\{A_i^{(b)} = a\}$ – априорное распределение на индексах подмножеств.

– М-шаг. На этом шаге выполняется максимизация правдоподобия параметров $\alpha_j^{(b)*}$ и $w_j^{(b)}$ с учетом $\mathcal{A}^{(b)}$:

$$\forall a \in \{1, 2, \dots, k^{(b)}\} : w_a^{(b)} = \frac{1}{m} \sum_{i=1}^m \mathbf{p}(a | \hat{n}_i^{(b)}), \quad (3.6)$$

$$\forall a \in \{1, 2, \dots, k^{(b)}\} : \alpha_a^{(b)*} = \arg \max_{\alpha} \sum_{i=1}^m g_{i,a} \ln \left(\text{Lap}(\hat{n}_i^{(b)} | \alpha, 0) \right). \quad (3.7)$$

Рассмотрим оптимизацию данного выражения. Отметим, что решение оптимизационной данной задачи распадается на $k^{(b)}$ независимых задач. Рассмотрим максимизацию для некоторого индекса a .

$$\begin{aligned} \alpha_a^{(b)*} &= \arg \max_{\alpha} \sum_{i=1}^m g_{i,a} \ln \left(\text{Lap}(\hat{n}_i^{(b)} | \alpha, 0) \right) = \\ &= \arg \max_{\alpha} \sum_{i=1}^m g_{i,a} \ln \left(\frac{\alpha}{2} e^{-\alpha |\hat{n}_i^{(b)}|} \right) = \\ &= \arg \max_{\alpha} \ln(\alpha) \sum_{i=1}^m g_{i,a} - \alpha \sum_{i=1}^m g_{i,a} |\hat{n}_i^{(b)}|. \end{aligned}$$

Для поиска оптимального значения параметра распределения приравняем производную от получившегося выражения к нулю.

$$\frac{d}{d\alpha} \left(\ln(\alpha) \sum_{i=1}^m g_{i,a} - \alpha \sum_{i=1}^m g_{i,a} |\hat{n}_i^{(b)}| \right) = \frac{1}{\alpha} \sum_{i=1}^m g_{i,a} - \sum_{i=1}^m g_{i,a} |\hat{n}_i^{(b)}| = 0.$$

Отсюда получим выражения для $\alpha_a^{(b)*}$:

$$\alpha_a^{(b)*} = \frac{\sum_{i=1}^m g_{i,a}}{\sum_{i=1}^m g_{i,a} |\hat{n}_i^{(b)}|}. \quad (3.8)$$

Величина 3.8 рассчитывается на М-шаге EM-алгоритма независимо для каждой из компонент смеси.

Е и М шаги повторяются до тех пор, пока либо значения скрытых переменных $A_i^{(b)}$ не перестанут существенно меняться между итерациями, либо не будет достигнуто максимальное число итераций. Условия сходимости EM-алгоритма рассматриваются в работе [54]. Выходом данной процедуры являются параметры распределений в множестве $\mathcal{D}^{(b)}$, а также веса компонент w_j . Для расчета индексов подмножеств предлагается проанализировать вероятности скрытых переменных и выбирать для каждой ошибки то распределение, которое соответствует максимуму апостериорной вероятности события, что ошибка в позиции i получена из распределения с номером j :

$$\forall i \in \{1, 2, \dots, m\} : a_i^{(b)} = \arg \max_{a \in \{1, 2, \dots, k^{(b)}\}} p(a | \hat{n}_i^{(b)}). \quad (3.9)$$

Полученное в результате применения данного алгоритма сопоставление между ошибками и распределениями, задаваемое множеством индексов $\{a_1^{(b)}, a_2^{(b)}, \dots, a_m^{(b)}\}$ не учитывает тот факт, что ошибки являются зависимыми. Для учета пространственной зависимости предлагается выполнить сглаживающую двумерную фильтрацию данного множества с помощью медианного фильтра, подавив тем самым возможные ошибки сопоставления. В результате, значения $\{a_1^{(b)}, a_2^{(b)}, \dots, a_m^{(b)}\}$ будут задавать разбиение множества ошибок на подмножества. Далее для оценки параметров ошибок для каждого квантованного спектрального коэффициента предлагается воспользоваться методом из базового алгоритма 3.2. Таким образом, модифицированный алгоритм оценки параметров ошибок межкадрового предсказания для группы спектральных коэффициентов с номером b представляет собой следующую последовательность действий.

Входные данные:

- аппроксимация ошибок межкадрового предсказания в спектральной области – $\hat{\mathbf{n}}^{(b)}$;

- начальное множество параметров распределений в смеси $\mathcal{L}_{in}^{(b)} = \{\alpha_1, \alpha_2, \dots, \alpha_{k^{(b)}}\}$;
- начальные значения весов $w_1^{(b)}, w_2^{(b)}, \dots, w_{k^{(b)}}^{(b)}$ для компонент смеси;
- максимальное число итераций EM-алгоритма t_1 ;
- порог на изменение множества индексов t_a ;
- начальные значения скрытых переменных $g_{i,a}$, для $i = 1, 2, \dots, m$ и $a = 1, 2, \dots, k^{(b)}$;
- число итераций пространственного сглаживания t_2 ;
- радиус сглаживающего фильтра r_f .

Выходные данные:

- оцененные параметры шума для каждого спектрального коэффициента в группе.

Алгоритм.

1. Инициализация:

- $\mathcal{L}^{(b)} = \mathcal{L}_{in}^{(b)}$;
- номер итерации EM-алгоритма $t = 1$;
- начальная маркировка $a_i = 1$, для $i = \overline{1, m}$.

2. Инициализация скрытых переменных:

$$\forall i \in \{1, 2, \dots, m\}, \forall a \in \{1, 2, \dots, k^{(b)}\} : g_{i,a}^{(0)} = g_{i,a}.$$

3. Расчет ожидаемого значения скрытых случайных величин $g_{i,a}$ с использованием формулы (3.5).
4. Расчет новых весов компонент смеси и параметров распределений с использованием формул (3.6) и (3.7).
5. Увеличение номера итерации: $t = t + 1$.
6. Расчет критерия остановки предварительной маркировки:

$$Stop = (t > t_1) \text{ или } (\max_{i,a} |g_{i,a} - g_{i,a}^{(0)}| < t_a).$$

7. Если $Stop$ принимает истинное значение, то переход к шагу 8, иначе переход к шагу 2.
8. Определение маркировки узлов:

$$a_i = \arg \max_{a=\{1,2,\dots,k^{(b)}\}} g_{i,a}. \quad (3.10)$$

9. Повторять t_2 раз:

$$\begin{aligned} \forall i \in \{1, 2, \dots, m\} : a_i^{(filt)} &= \text{median}(\{a_j\} : d(\text{map}^{-1}(j), \text{map}^{-1}(i)) < r_f) \\ \forall i \in \{1, 2, \dots, m\} : a_i &= a_i^{(filt)} \end{aligned}$$

где $\text{map}^{-1}(\cdot)$ – функция, отображающая одномерные индексы в множестве ошибок в двумерные индексы соответствующих блоков на кадре; $\text{median}(\cdot)$ – функция взятия срединного элемента (медианы) множества.

10. Расчет выборочных средних:

$$\forall j \in \{1, 2, \dots, k^{(b)}\} : \mu_j^{(b)} = \frac{1}{\sum_{i=1}^m \mathbb{1}(a_i = j)} \sum_{i=1}^m |\hat{n}_i^{(b)}| \mathbb{1}(a_i = j),$$

где $\mathbb{1}(\cdot)$ – индикаторная функция, определенная как

$$\mathbb{1}(X) = \begin{cases} 1, & \text{если } X \text{ принимает истинное значение} \\ 0, & \text{иначе} \end{cases}.$$

11. Расчет выборочных среднеквадратичных отклонений:

$$\forall j \in \{1, 2, \dots, k^{(b)}\} : \sigma_j^{(b)} = \sqrt{\frac{1}{\sum_{i=1}^m \mathbb{1}(a_i = j)} \sum_{i=1}^m (|\hat{n}_i^{(b)}| - \mu_j^{(b)})^2 \mathbb{1}(a_i = j)}$$

12. Расчет отклонения от выборочного среднего:

$$\forall i \in \{1, 2, \dots, m\} : d_i^{(b)} = (|\hat{n}_i^{(b)}| - \hat{\mu}_{a_i}^{(b)})^2.$$

13. Оценка параметров для каждой ошибки:

$$\forall i \in \{1, 2, \dots, m\} : \alpha_i^{(b)} = \begin{cases} \frac{\sqrt{2}}{\sigma_{a_i}^{(b)}}, & \text{если } d_i^{(b)} < (\sigma_j^{(b)})^2 \\ \frac{\sqrt{2}}{d_i^{(b)}}, & \text{иначе} \end{cases},$$

В завершение описания алгоритма приведем связь предложенного метода с базовым подходом из модели DISCOVER. Пусть для любой группы спектральных коэффициентов начальное множество распределений состоит из одного ве-

роятностного закона, т. е. $\mathcal{L}_{in}^{(b)} = \{\alpha_1\}$. Тогда на шагах 2-8 будет осуществляться оценка параметра масштаба α по выборке ошибок $\hat{\mathbf{n}}^{(b)}$. Т. к. на шаге 9 фактически выполняется медианная фильтрация, конфигурация поля распределений меняться не будет. Очевидно, что в таком случае оценка параметров на шагах 10-13 будет в точности совпадать с оценкой в базовом алгоритме. В том случае, когда начальное множество распределений $\mathcal{L}_{in}^{(b)}$ состоит из $k^{(b)}$ законов, алгоритм будет выполнять разбиение выборки $\hat{\mathbf{n}}^{(b)}$ на максимум $k^{(b)}$ подвыборок, причем по сравнению с базовым алгоритмом оценка интенсивности шума будет снижена для регионов с высоким качеством интерполяции и повышена для регионов с низким качеством интерполяции. Продемонстрируем данное утверждение на примере из подраздела 3.2.3. Напомним, что оценка шума для данного примера имеет следующий вид:

$$\hat{\mathbf{n}}^{(1)} = \begin{pmatrix} 2 & 1 & 3 & 5 \\ 0 & 0 & 4 & 83 \\ 7 & 24 & 157 & 208 \\ 1 & 1 & 30 & 132 \end{pmatrix}. \quad (3.11)$$

Матрица коэффициентов масштаба распределения Лапласа, полученная с использованием базового алгоритма:

$$\begin{pmatrix} 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & \mathbf{0.012} & \mathbf{0.008} \\ 0.021 & 0.021 & 0.021 & \mathbf{0.016} \end{pmatrix}. \quad (3.12)$$

Продемонстрируем работу предложенного алгоритма. Зададим входные параметры:

- начальное множество параметров распределений $\mathcal{L}_{in}^{(b)} = \{0.47, 0.94\}$;
- начальные значения весов для компонент смеси: $w_1^{(b)} = 0.5, w_2^{(b)} = 0.5$;
- максимальное число итераций EM-алгоритма $t_1 = 10000$;
- порог на изменение поля распределений $t_a = 10^{-4}$;
- начальные значения скрытых переменных $g_{i,a} = 0.5$, для $i = 1, 2, \dots, 16$ и $a = 1, 2$;
- число итераций пространственного сглаживания $t_2 = 1$;
- радиус сглаживающего фильтра $r_f = 1$.

В результате применения EM-алгоритма (шаги 2-7 предложенного алгоритма) через 12 итераций получим следующее разбиение множества ошибок (шаг 8):

$$\begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \end{pmatrix}.$$

После медианной фильтрации:

$$\begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \end{pmatrix}.$$

После применения базового алгоритма независимо в областях, помеченных индексами 1 и 2, получаются следующие оценки параметров распределений Лапласа.

$$\begin{pmatrix} 0.206 & 0.206 & 0.206 & 0.206 \\ 0.206 & 0.206 & 0.206 & 0.021 \\ 0.206 & 0.072 & 0.021 & 0.017 \\ 0.206 & 0.206 & 0.015 & 0.021 \end{pmatrix}.$$

Из полученных результатов видно, что по сравнению с базовым, предложенный алгоритм позволяет существенно увеличить разницу в надежности спектральных коэффициентов с малыми и большими оцененными значениями ошибок. Например, коэффициенту со значением 0, находящемуся на позиции (2,2), базовый алгоритм назначил значение масштаба, равное 0.021. В то же время, результат предложенного алгоритма для данного коэффициента равен 0.206.

3.4 Описание предложенной аналитической модели виртуального канала

В начале данного подраздела необходимо отметить, что в настоящее время виртуальный канал рассматривается как некоторое абстрактное понятие. Не суще-

ствуется строгого математического описания модели ошибок в виртуальном канале, что усложняет анализ существующих алгоритмов оценки параметров ошибок межкадрового предсказания и ухудшает понимание специфики и особенностей данной задачи в целом. В связи с этим в данном подразделе вводится в рассмотрение модель искажения промежуточного кадра в процессе его аппроксимации на стороне декодера. В рамках данной модели выполняется формализация понятия виртуального канала и формулируется возможная целевая функция для алгоритмов оценки параметров ошибок межкадрового предсказания.

Рассмотрим расширенный набор допущений о виртуальном канале.

Допущение 3.2 (*). *Виртуальный канал является каналом с аддитивным нестационарным шумом.*

Данное допущение обобщает допущение 3.2, позволяя рассматривать не только закон распределения Лапласа. Для определенности будем полагать, что ошибки в виртуальном канале являются случайными величинами, закон распределения каждой из которых принадлежит множеству $\mathcal{D}^{(b)} = \{D_i^{(b)}(x|\theta_i^{(b)})\}$, $i = \overline{1, k^{(b)}}$, где $D_i^{(b)}(x|\theta_i^{(b)})$ задает вероятностный закон и его параметры для распределения с индексом i . Элементы в множестве $\mathcal{D}^{(b)}$ могут описывать один закон распределения, но в таком случае должны различаться по параметрам. Данное допущение обобщает соответствующее допущение из базового набора, позволяя рассматривать произвольные законы распределений в смеси, а не только распределение Лапласа.

Напомним, что допущение 3.2 оперирует с понятием случайной величины $N_i^{(b)}$, реализацией которой является значение ошибки для спектрального коэффициента, находящегося на позиции i в группе спектральных коэффициентов b .

Определение 3.1. Будем называть множество случайных величин $\mathcal{N}^{(b)} = \{N_1^{(b)}, N_2^{(b)}, \dots, N_m^{(b)}\}$ полем ошибок.

Определение 3.2. Будем называть множество случайных величин $\mathcal{A}^{(b)} = \{A_1^{(b)}, A_2^{(b)}, \dots, A_m^{(b)}\}$, где $A_j^{(b)} \in \{1, 2, \dots, k^{(b)}\}$, полем распределений.

Отметим, что для стационарного виртуального канала совместная функция плотности вероятности случайных величин из множества $\mathcal{N}^{(b)}$ задается как

$$f(x|\mathcal{D}^{(b)}) = \sum_{i=1}^{k^{(b)}} w_i^{(b)} D_i^{(b)}(x|\boldsymbol{\theta}_i^{(b)}). \quad (3.13)$$

Для сокращения обозначений далее не будем указывать формальный параметр x при написании закона распределения.

В процессе моделирования виртуального канала каждой случайной величине из $\mathcal{N}^{(b)}$ должно быть сопоставлено распределение из $\mathcal{D}^{(b)}$. Будем обозначать событие, что $N_i^{(b)}$ имеет закон распределения $D_j^{(b)}(\boldsymbol{\theta}_j^{(b)})$, как $N_i^{(b)} \sim D_j^{(b)}(\boldsymbol{\theta}_j^{(b)})$. Событие, заключающееся в сопоставлении законов распределений из $\mathcal{D}^{(b)}$ со случайными величинами из $\mathcal{N}^{(b)}$, можно рассматривать как реализацию случайного поля $\mathcal{A}^{(b)}$. Нестационарность шума в виртуальном канале означает, что

$$\Pr \left[\bigcap_{i=1}^m A_i^{(b)} = a_i \right] \neq \prod_{i=1}^m \Pr\{A_i^{(b)} = a_i\}$$

Допущение 3.3. Поле распределений является Марковским случайным полем.

Определение 3.3. Марковское случайное поле, или сеть Маркова, – это случайное поле, обладающее Марковским свойством, описанным неориентированным графом.

Допущения 3.2 и 3.3 позволяют рассматривать процесс появления шума с точки зрения вероятностной порождающей модели (PGM, Probabilistic Generative Model). На качественном уровне появление шума в данной модели можно пояснить следующим образом. Следующие шаги выполняются для каждого спектрального коэффициента.

1. С помощью некоторой процедуры выбирается распределение из $\mathcal{D}^{(b)}$ так, чтобы учитывать пространственную зависимость законов распределений ошибок спектральных коэффициентов соседних блоков;
2. В соответствии с выбранным законом генерируется случайная величина, соответствующая ошибке в данном спектральном коэффициенте.

Подобная модель позволяет свести задачу оценки параметров ошибок межкадрового предсказания к согласованному сопоставлению ошибок и распределений из $\mathcal{D}^{(b)}$.

Напомним, что для того, чтобы задать МСП необходимо определить множество случайных величин (узлы графа) и вероятностные зависимости (дуги графа) между ними (приложение А).

Зададим МСП над множеством случайных величин $\mathcal{A}^{(b)}$. Для этого введем в рассмотрение регулярную прямоугольную решетку узлов $\mathcal{S}^{(b)}$ как:

$$\mathcal{S}^{(b)} = \{(i,j)^T | i = \{1,2,\dots,h_b\}, j = \{1,2,\dots,w_b\}\},$$

где $w_b = \frac{w}{4}$ и $h_b = \frac{h}{4}$, w и h ширина и высота кадра \mathbf{F}_a соответственно. Делитель 4 задается размером блока в дискретном косинусном преобразовании, т. е. w_b и h_b определяют число спектральных коэффициентов в одной группе:

$$m = w_b \times h_b.$$

Здесь и далее будем использовать понятия «узел» и «позиция узла» как синонимы. Следует отметить, что узлы в множестве $\mathcal{S}^{(b)}$ задают на плоскости прямоугольную регулярную решетку, координаты в которой можно рассматривать как индексы блоков при дискретном косинусном преобразовании. Отметим также, что так как число спектральных коэффициентов не меняется между группами, координаты в решетке не привязаны к конкретному номеру b . Следуя данному соглашению, сопоставим элементы множества $\mathcal{A}^{(b)}$ с узлами решетки, для определенности предполагая, что спектральные коэффициенты в группе формировались с помощью обхода блоков по столбцам:

$$\begin{aligned} A_1^{(b)} &\leftrightarrow (1,1)^T \\ A_2^{(b)} &\leftrightarrow (2,1)^T \\ &\dots \\ A_{h_b}^{(b)} &\leftrightarrow (h_b,1)^T \quad . \\ A_{h_b+1}^{(b)} &\leftrightarrow (1,2)^T \\ &\dots \\ A_m^{(b)} &\leftrightarrow (h_b,w_b)^T \end{aligned} \tag{3.14}$$

Отметим, что сопоставление 3.14 задает взаимно однозначное отображение множества одномерных индексов случайных величин из \mathcal{A} на множество узлов двумерной решетки. Для приведенного порядка обхода данное отображение мож-

но определить как следующую функцию:

$$\text{map}(i, j) = i + (j - 1)h_b. \quad (3.15)$$

Далее определим свойство Марковости для случайного поля \mathcal{A} . Для этого зададим отношение соседства над узлами из $\mathcal{S}^{(b)}$:

$$\mathcal{R}^{(b)} = \{\mathcal{R}_{i,j}^{(b)} \mid \forall (i,j)^T \in \mathcal{S}^{(b)}\},$$

где $\mathcal{R}_{i,j}^{(b)}$ – множество соседей узла $(i,j)^T$. На регулярной решетке это множество можно задать следующим образом:

$$\mathcal{R}_{i,j}^{(b)} = \{(i',j')^T \in \mathcal{S}^{(b)} \mid 0 < \mathbf{d}((i,j)^T, (i',j')^T) \leq r\},$$

где r – параметр, определяющий количество соседей; функция $\mathbf{d}((i,j)^T, (i',j')^T)$ задает Евклидово расстояние в двумерном пространстве:

$$\mathbf{d}((i,j)^T, (i',j')^T) = \sqrt{(i - i')^2 + (j - j')^2}.$$

Таким образом, соседями узла $(i,j)^T$ являются те узлы, которые находятся от него на расстоянии не больше, чем r . Для Марковского случайного поля это означает, что

$$A_{i,j} \perp\!\!\!\perp \mathcal{A} \setminus \mathcal{R}_{i,j}^{(b)} \mid \mathcal{R}_{i,j}^{(b)}. \quad (3.16)$$

Множество $\mathcal{R}^{(b)}$ определяет вероятностные связи между случайными величинами из $\mathcal{A}^{(b)}$. На графической вероятностной модели для случайного поля эта зависимость выражается в наличии дуг между вершиной $(i,j)^T$ и её соседями из $\mathcal{R}_{i,j}$. Таким образом упорядоченная пара множеств $(\mathcal{S}^{(b)}, \mathcal{R}^{(b)})$ задает граф $\mathcal{G}^{(b)}$, соответствующий МСП над множеством случайных величин $\mathcal{A}^{(b)}$.

Рассмотрим связь описанного МСП с полученной на стороне декодера аппроксимацией шума. Конкретная реализация МСП задает разбиение множества случайных величин $\mathcal{N}^{(b)}$ на несколько подмножеств, причем в каждом подмножестве случайные величины имеют один закон распределения. В таком случае шум представляет собой реализацию поля $\mathcal{N}^{(b)}$ с учетом скрытых случайных величин $\mathcal{A}^{(b)}$. Таким образом, модель шума описывается скрытым Марковским случайным полем (СМСП), где наблюдаемым случайными величинами являются $N_1^{(b)}$,

$N_2^{(b)}, \dots, N_m^{(b)}$, а скрытыми – $A_1^{(b)}, A_2^{(b)}, \dots, A_m^{(b)}$. Пример вероятностной графической модели для $r = \sqrt{2}$ приведен на рисунке 3.5.

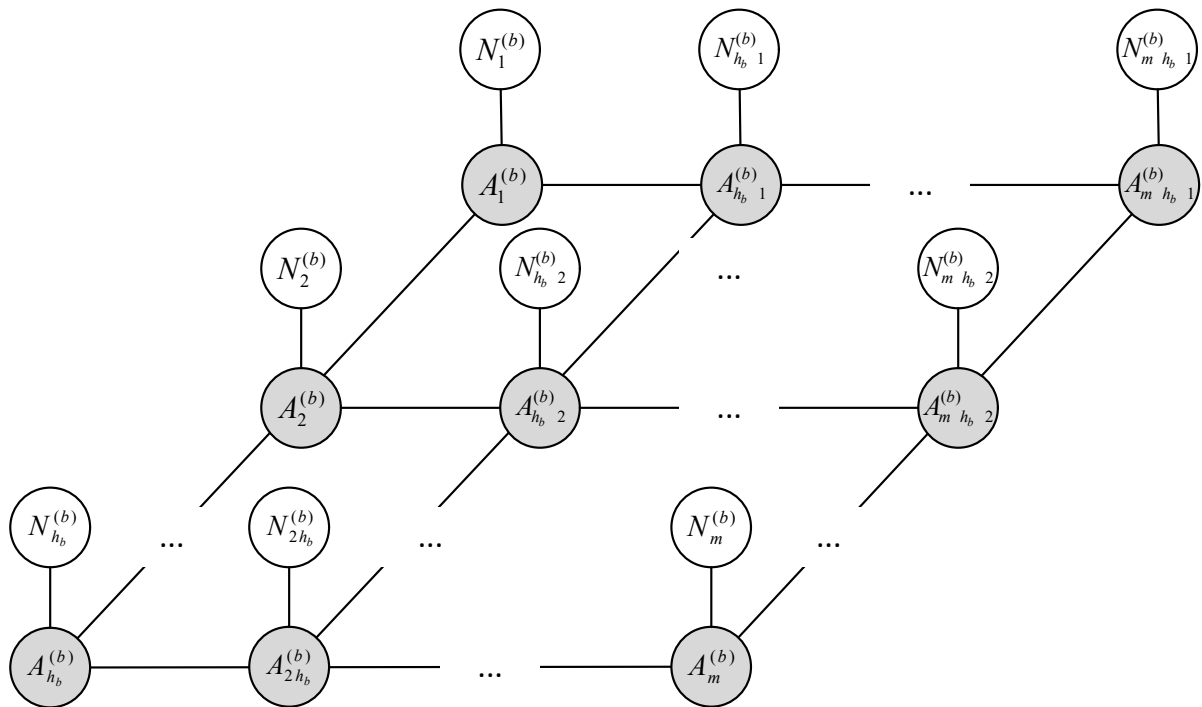


Рисунок 3.5 — Пример вероятностной графической модели для скрытого Марковского случайного поля.

С учетом приведенной модели задачу моделирования виртуального канала для группы спектральных коэффициентов с номером b можно свести к решению следующей оптимизационной задачи:

$$\begin{aligned}
 \mathbf{a}^{(b)*} &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} p(\mathbf{a} | \hat{\mathbf{n}}^{(b)}) = \\
 &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \frac{p(\hat{\mathbf{n}}^{(b)} | \mathbf{a}) p(\mathbf{a})}{p(\hat{\mathbf{n}}^{(b)})} =, \\
 &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} p(\hat{\mathbf{n}}^{(b)} | \mathbf{a}) p(\mathbf{a})
 \end{aligned} \tag{3.17}$$

где

$$p(\hat{\mathbf{n}}^{(b)} | \mathbf{a}) = \Pr \left\{ \bigcap_{i=1}^m N_i^{(b)} = \hat{n}_i^{(b)} \mid \bigcap_{i=1}^m A_i^{(b)} = a_i^{(b)} \right\}, \tag{3.18}$$

$$p(\mathbf{a}) = \Pr \left\{ \bigcap_{i=1}^m A_i^{(b)} = a_i^{(b)} \right\}. \tag{3.19}$$

Т. е. $p(\hat{\mathbf{n}}^{(b)}|\mathbf{a})$ является правдоподобием поля распределений для аппроксимации шума $\hat{\mathbf{n}}^{(b)}$, а $p(\mathbf{a})$ описывает априорное распределение вероятностей над множеством реализаций поля распределений.

Учитывая, что случайные величины $N_i^{(b)}$ независимы при известной реализации поля распределений $\mathcal{A}^{(b)}$, а также воспользовавшись свойствами условной независимости случайных величин в поле распределений, упростим выражение (3.18):

$$p(\hat{\mathbf{n}}^{(b)}|\mathbf{a}) = \prod_{i=1}^m p(\hat{n}_i^{(b)}|\mathbf{a}) = \prod_{i=1}^m p(\hat{n}_i^{(b)}|a_i). \quad (3.20)$$

Рассмотрим теперь выражение для априорного распределения вероятностей над множеством реализаций поля распределений. Согласно теореме Хаммерсли-Клиффорда [55], случайное поле $\mathcal{A}^{(b)}$ является полем Гиббса, и, следовательно, совместная плотность случайных величин из $\mathcal{A}^{(b)}$ может быть факторизована на кликах графа $\mathcal{G}^{(b)}$:

$$p(\mathbf{a}) = \frac{1}{z} \exp\left(-\frac{1}{t}U(\mathbf{a})\right), \quad (3.21)$$

где t – параметр распределения Гиббса; z – коэффициент нормировки, определяемый как

$$z = \sum_{\mathbf{a} \in \{1,2,\dots,k^{(b)}\}^m} \exp\left(-\frac{1}{t}U(\mathbf{a})\right); \quad (3.22)$$

$U(\mathbf{a})$ – сумма потенциальных функций на кликах графа $\mathcal{G}^{(b)}$. Обозначим множество клик $\mathcal{G}^{(b)}$ через \mathcal{C} , тогда

$$U(\mathbf{a}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a}), \quad (3.23)$$

где $\psi_c(\cdot)$ – потенциалы клик.

Учитывая вышеприведенные соотношения, перепишем оптимизационную задачу (3.17):

$$\begin{aligned} \mathbf{a}^{(b)*} &= \arg \max_{\mathbf{a} \in \{1,2,\dots,k^{(b)}\}^m} p(\hat{\mathbf{n}}^{(b)}|\mathbf{a})p(\mathbf{a}) \\ &= \arg \max_{\mathbf{a} \in \{1,2,\dots,k^{(b)}\}^m} \prod_{i=1}^m p(\hat{n}_i^{(b)}|a_i) \frac{1}{z} \exp\left(-\frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a})\right). \\ &= \arg \max_{\mathbf{a} \in \{1,2,\dots,k^{(b)}\}^m} \prod_{i=1}^m p(\hat{n}_i^{(b)}|a_i) \exp\left(-\frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a})\right) \end{aligned} \quad (3.24)$$

Выражение (3.24) является общей целевой функцией при решении задачи моделирования виртуального канала. Отметим, что в данном выражении в явном виде не задаются ни вероятности, связывающие наблюдаемые ошибки $\hat{n}_i^{(b)}$ с индексами распределений a_i , из которых они получены, ни конкретный вид функций $\psi_c(\mathbf{a})$.

Рассмотрим один частный случай оптимизации (3.24) в предположении, что правдоподобие поля $\mathcal{A}^{(b)}$ задается параметрическим семейством экспоненциальных распределений, т. е.

$$\forall j \in \{1, 2, \dots, k^{(b)}\} : D_j^{(b)}(x | \boldsymbol{\theta}_j^{(b)}) = \zeta(\boldsymbol{\theta}_j^{(b)}) \exp\left(-\phi_j(x, \boldsymbol{\theta}_j^{(b)})\right),$$

где $\zeta(\boldsymbol{\theta}_j^{(b)})$ – некоторый коэффициент, зависящий от параметров распределения; $\phi_i(\cdot, \cdot)$ – некоторая функция, обратно пропорциональная правдоподобию распределения из $\mathcal{D}^{(b)}$. Напомним, что к семейству экспоненциальных распределений относятся все дискретные, а также многие непрерывные распределения (например, распределения Гаусса, Лапласа, Дирихле, Бэта и т. д.).

Используя выражение (3.4), оптимизационную задачу (3.24) можно упростить следующим образом:

$$\begin{aligned} \mathbf{a}^{(b)*} &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \prod_{i=1}^m p(\hat{n}_i^{(b)} | a_i) \exp\left(-\frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a})\right) = \\ &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \left[\prod_{i=1}^m \zeta(\boldsymbol{\theta}_{a_i}^{(b)}) \exp\left(-\phi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)})\right) \right] \exp\left(-\frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a})\right) = \\ &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \left[\prod_{i=1}^m \zeta(\boldsymbol{\theta}_{a_i}^{(b)}) \right] \exp\left(-\sum_{i=1}^m \phi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}) - \frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a})\right) = \\ &= \arg \max_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \sum_{i=1}^m \ln \zeta(\boldsymbol{\theta}_{a_i}^{(b)}) - \sum_{i=1}^m \phi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}) - \frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a}) = \\ &= \arg \min_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \sum_{i=1}^m \left(\phi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}) - \ln \zeta(\boldsymbol{\theta}_{a_i}^{(b)}) \right) + \frac{1}{t} \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a}). \end{aligned} \tag{3.25}$$

Обозначим $\left[\phi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}) - \ln \zeta(\boldsymbol{\theta}_{a_i}^{(b)}) \right]$ через $\varphi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)})$ и положим $t = 1$, тогда

$$\mathbf{a}^{(b)*} = \arg \min_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \sum_{i=1}^m \varphi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}) + \sum_{c \in \mathcal{C}} \psi_c(\mathbf{a}). \quad (3.26)$$

Рассмотрим получившееся выражение. Для определенности зададим $r = \sqrt{2}$, т. е. множество соседей для каждого узла включает только те узлы, которые находятся на расстоянии не больше $\sqrt{2}$ от него. Данная конфигурация соответствует графу $\mathcal{G}^{(b)}$ с кликами, состоящими либо из одной вершины, либо из пары вершин (рисунок 3.5). Тогда выражение 3.26 можно переписать следующим образом:

$$\mathbf{a}^{(b)*} = \arg \min_{\mathbf{a} \in \{1, 2, \dots, k^{(b)}\}^m} \left[E_{data}(\mathbf{a}, \hat{\mathbf{n}}^{(b)}) + E_{smooth}(\mathbf{a}) + E_{label}(\mathbf{a}) \right], \quad (3.27)$$

где

$$E_{data}(\mathbf{a}, \hat{\mathbf{n}}^{(b)}) = \sum_{i=1}^m \varphi_i(\hat{n}_i^{(b)}, \boldsymbol{\theta}_{a_i}^{(b)}),$$

$$E_{smooth}(\mathbf{a}) = \sum_{i=1}^m \sum_{j \in \mathcal{R}_i} \psi_{i,j}(a_i, a_j),$$

$$E_{label}(\mathbf{a}) = \sum_{i=1}^m \psi_i(a_i).$$

Таким образом, в процессе минимизации выражения 3.27 осуществляется поиск такой конфигурации поля распределений, которая одновременно:

- уменьшает невязку наблюдаемых случайных величин $\hat{n}_i^{(b)}$ с сопоставленными им законами распределений (минимизация слагаемого E_{data});
- учитывает пространственную зависимость распределений, уменьшая рассогласование между метками соседних коэффициентов (минимизация слагаемого E_{smooth});
- позволяет задавать априорную информацию о маркерах распределений (минимизация слагаемого E_{label}).

Следует отметить, что данная теоретическая модель обладает рядом особенностей.

- В общем случае для произвольных функций $\varphi_i(\cdot)$, $\psi_i(\cdot)$ и $\psi_{i,j}(\cdot, \cdot)$, минимизация (3.27) является NP-полной задачей и требует перебора $(k^{(b)})^m$

различных реализаций поля распределений для поиска глобального минимума. Однако, для некоторых классов функций потенциалов, в частности, когда данные функции являются метриками или полу-метриками, существуют алгоритмы, которые позволяют находить локальный минимум выражения (3.27).

- Параметризация данной модели, а именно задача, связанная с определением вида функций $\varphi_i(\cdot)$, $\psi_i(\cdot)$ и $\psi_{i,j}(\cdot, \cdot)$, является в настоящее время открытой. В общем случае, не существует алгоритма подбора функций, описывающих зависимости между случайными величинами в Марковском случайном поле. Более того, данная задача является задачей без ограничений в бесконечномерном пространстве функций, что исключает возможность её решения без наложения ограничений, например, на семейство функций. Следует отметить, что при наличии ограничений такого рода, задача может быть решена, например, с использованием алгоритмов обучения графических моделей [56].

В завершение описания модели приведем значения целевой функции, которые получаются при использовании базового и модифицированного алгоритма. Расчет будем проводить для примера, разобранный в подразделах 3.2.3 и 3.4. Для определенности будем полагать, что Марковское случайное поле описывается четырехсвязным графом (рисунок 3.5). Совместную плотность для «смежных» случайных величин зададим как экспоненциальное распределение от разности между параметрами масштаба, рассчитанными для распределения Лапласа в позициях соответствующих коэффициентов:

$$p(a_i, a_j) = \lambda e^{(-\lambda|\alpha_{a_i} - \alpha_{a_j}|)}, \forall i, j : i \neq j \text{ и } |i^2 - j^2| < \sqrt{2}.$$

Зададим значение коэффициента $\lambda = 1$. Значения оцененных ошибок межкадрового предсказания:

$$\hat{\mathbf{n}}^{(1)} = \begin{pmatrix} 2 & 1 & 3 & 5 \\ 0 & 0 & 4 & 83 \\ 7 & 24 & 157 & 208 \\ 1 & 1 & 30 & 132 \end{pmatrix}.$$

Результат базового алгоритма:

$$\begin{pmatrix} 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & 0.021 & 0.021 \\ 0.021 & 0.021 & 0.012 & 0.008 \\ 0.021 & 0.021 & 0.021 & 0.016 \end{pmatrix}.$$

Результат модифицированного алгоритма:

$$\begin{pmatrix} 0.206 & 0.206 & 0.206 & 0.206 \\ 0.206 & 0.206 & 0.206 & 0.021 \\ 0.206 & 0.072 & 0.021 & 0.017 \\ 0.206 & 0.206 & 0.015 & 0.021 \end{pmatrix}.$$

Значение логарифма от целевой функции для базового алгоритма составляет -83.85 , для модифицированного алгоритма -70.3 , т. е. с точки зрения рассмотренной модели модифицированный алгоритм выполняет более вероятное назначение распределений оцененным ошибкам межкадрового предсказания.

3.5 Выводы

В данном разделе были рассмотрены вопросы, связанные с оценкой параметров ошибок в виртуальном канале в рамках систем распределенного кодирования видеоданных. Основные результаты раздела можно сформулировать следующим образом:

- проанализирован алгоритм оценки параметров ошибок межкадрового предсказания, используемый в базовой модели распределенного кодирования на основе проекта DISCOVER, выделены его основные особенности и недостатки;
- предложена модификация алгоритма оценки параметров ошибок межкадрового предсказания, учитывающая свойство группировки ошибок с большой амплитудой;
- введен новый, расширенный, набор допущений о виртуальном канале;

- предложена аналитическая модель виртуального канала, основанная на моделировании ошибок межкадрового предсказания как случайного процесса, описываемого скрытым Марковским случайным полем.

Вопросы, связанные со сравнительной оценкой модифицированного и базового алгоритмов, рассматриваются в разделе 4 данной диссертационной работы.

4 СРАВНИТЕЛЬНЫЙ АНАЛИЗ РАЗРАБОТАННЫХ АЛГОРИТМОВ

4.1 Вводные замечания

В разделах 2 и 3 данной работы было приведено описание двух новых алгоритмов обработки видеинформации в рамках системы распределенного видеокодирования. Следует отметить, что задача сравнительной оценки эффективности данных алгоритмов является сложной задачей ввиду того, что в открытом доступе нет исходного кода распределенного видеокodeка, который можно было бы принять за «эталонный». Кодек, разработанный в рамках проекта DISCOVER [18], не предоставляет возможности внесения изменений, т. к. распространяется только в виде бинарного исполняемого файла. Кроме проекта DISCOVER существует три открытых исследовательских проекта, посвященных распределенному кодированию.

- Проект BLAST-DVC (BitpLANe SelecTive Distributed Video Coding) [57]. Основная цель данного проекта заключается в разработке распределенного видеокodeка, расширяющего модель DISCOVER возможностью оценки важности битовых плоскостей с точки зрения кривой «скорость-искажение». В схеме BLAST-DVC декодер запрашивает у кодера не все битовые плоскости квантованных спектральных коэффициентов, начиная с наиболее значимого спектрального коэффициента, а только те, которые ему необходимо восстановить для обеспечения заданного качества восстановления. Результат проекта доступен в виде исполняемого файла.
- Проект OpenDVC [58]. В результате данного проекта предполагалась разработка открытой реализации распределенного видеокodeка. В качестве базовых модулей использовались алгоритмы, схожие с алгоритмами из базовой модели DISCOVER. Однако цель данного проекта не была достигнута, в настоящее время проект больше не развивается, хотя доступны исходные коды промежуточных версий кодека. Следует отметить, что исходный код OpenDVC характеризуется наличием большого числа ошибок и неточностей. Этот факт, а также отсутствие какой-либо документа-

ции и научных статей, описывающих OpenDVC, не позволяют использовать результаты данного проекта в качестве эталонного кодека.

- Проект Stanford DVC [59]. В данном проекте основное внимание уделялось разработке адаптивных по скорости помехоустойчивых кодов для распределенного кодирования видеоданных, в частности, LPDCA кодов. Результаты данного проекта используются в кодеках DISCOVER, OpenDVC и BLAST-DVC для блока помехоустойчивого кодирования.

Таким образом, можно сделать вывод о том, что в настоящее время не существует открытой завершенной реализации распределенного видеокодека, и для проведения сравнительной экспериментальной оценки требуется в первую очередь разработать эталонный программный комплекс, реализующий основные концепции распределенного видеокодирования и позволяющий исследовать различные алгоритмы обработки видеоинформации в рамках данного подхода. В связи с этим настоящий раздел начинается с описания ключевых алгоритмов, особенностей и возможностей разработанной программной модели распределенного видеокодека. Далее приводится описание проведенных экспериментов для демонстрации выигрыша от применения разработанных алгоритмов и основные практические результаты, полученные с использованием данной программной модели. Перед тем, как переходить к описанию программного комплекса, введем в рассмотрение основные критерии сравнения алгоритмов сжатия видеоинформации, а также опишем тестовое множество видеопоследовательностей, которое будет использоваться в дальнейшем при сравнении.

4.2 Критерии сравнения алгоритмов сжатия видеоданных

Для количественного оценивания искажений, внесенных в изображение в результате обработки, как правило, используют критерий PSNR (пиковое отношение сигнал-шум), рассчитанный по яркостной компоненте. Для 8-битных компонент Y и \hat{Y} значение PSNR рассчитывается как

$$\text{PSNR}(Y, \hat{Y}) = 10 * \log_{10} \frac{255^2}{\text{MSE}(Y, \hat{Y})}, \quad (4.1)$$

где $\text{MSE}(\cdot, \cdot)$ – среднеквадратичное отклонение, определяемое как

$$\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{w * h} \sum_{i=1}^h \sum_{j=1}^w \left(\mathbf{Y}[i,j] - \hat{\mathbf{Y}}[i,j] \right)^2, \quad (4.2)$$

w, h – количество столбцов и строк в компоненте соответственно.

Чем выше значение PSNR, тем больше похожи компоненты \mathbf{Y} и $\hat{\mathbf{Y}}$. Считается, что $\text{PSNR} > 40$ соответствует очень высокой схожести изображений, когда искажения почти неразличимы человеческим глазом. Следует отметить, что в настоящее время существует большое число работ, в которых показывается, что данный критерий во многих случаях слабо коррелирует с человеческим восприятием искажений. В связи с этим ведутся активные исследования в области разработки новых критериев, например SSIM, MS-SSIM. Однако, несмотря на критику, PSNR до сих пор является наиболее распространенным критерием в различных прикладных задачах цифровой обработки изображений. По этой причине в данной диссертационной работе будут использоваться критерии основанные в первую очередь на расчете значения PSNR между парой изображений (компонент изображений).

В рамках задачи сравнительной оценки алгоритмов временной интерполяции, как правило, используются следующие критерии:

- графики покадрового значения PSNR;
- среднее значение PSNR, рассчитанное по всем интерполированным кадрам.

Графики покадрового PSNR позволяют выделить кадры, характеризующиеся, например, низким качеством интерполяции. Данные кадры могут быть использованы для определения узких мест алгоритма и повышения качества интерполяции в целом.

Среднее значение PSNR позволяет судить об эффективности алгоритма в целом. Именно этот критерий является наиболее распространенным при принятии решения о том, какой алгоритм интерполяции лучше.

Основным методом сравнения алгоритмов сжатия видеоданных является анализ кривых «скорость-искажение», показывающих для заданной битовой скорости среднее значение PSNR для восстановленных кадров, а также оценка численных критериев, основанных на них. Наиболее распространенным методом сравнения кривых «скорость-искажение» является анализ критерия Бьёнтегаар-

да [60], позволяющего оценить численно расстояние между двумя кривыми. Расстояние может выражаться в децибелах (BD-PSNR), показывая среднее различие по PSNR, или в процентах (BD-RATE), показывая относительное различие в битовой скорости. В настоящее время критерий BD-RATE является общепринятым при сравнительном анализе алгоритмов сжатия в процессе разработки новых стандартов кодирования видеoinформации.

4.3 Формирование тестового множества видеопоследовательностей

Кодек DISCOVER поддерживает обработку видеопоследовательностей с разрешением CIF (352×288) или QCIF (176×144) в формате YUV420, т. е. каждый кадр представлен одной яркостной составляющей Y и двумя хроматическими составляющими U, V. Сжатию подвергается только компонента Y.

Стандартное тестовое множество кодека DISCOVER включает в себя четыре последовательности с разрешением QCIF [61].

- Coastguard. В последовательности преобладает горизонтальное равномерное смещение. Кадры характеризуются большим количеством незначимых для визуального восприятия текстурных элементов.
- Foreman. Первая половина последовательности содержит изображение объекта, незначительно перемещающегося в центре сцены. Перемещение характеризуется наличием существенных деформаций формы объекта. Во второй половине происходит быстрое горизонтальное смещение с изображения объекта на статичную сцену.
- Hall. Представляет собой видео, снятое с камеры видеонаблюдения, установленной в коридоре. Характеризуется наличием заметных шумов, отсутствием смещения камеры и небольшим движением в центре сцены на протяжении почти всей последовательности.
- Soccer. Последовательность характеризуется быстрым и сложным движением небольшого числа объектов.

Т. к. видео с разрешением QCIF в настоящее время можно считать мало репрезентативным, в данной диссертационной работе предлагается при проведении экспериментов использовать данные последовательности с максимально поддер-

живаемым кодеком DISCOVER разрешением – CIF. Кроме того, предлагается расширить тестовое множество, добавив в него одну последовательность с быстрым движением – Football.

4.4 Описание разработанной программной модели распределенного видеокodeка

В основе разработанной программной модели распределенного видеокodeка лежит архитектура Стэнфорд, т. е. в качестве единицы обрабатываемых кодеком Слепяна-Вулфа данных выступает кадр целиком. Модель codeка представляет собой реконфигурируемый программный комплекс, предоставляющий возможность замены модулей при условии сохранения интерфейса взаимодействия отдельных блоков системы. В качестве базовой реализации для всех модулей применяются методы, основанные на подходах из базовой модели DISCOVER. Однако, в силу того что в открытой литературе нет достаточно подробного описания всех особенностей реализации данных алгоритмов в DISCOVER (более того, в некоторых статьях дано различное описание одних и тех же блоков), верификация разработанной программной модели не представляется возможной. Как будет продемонстрировано далее, результаты полученные с использованием разработанного программного комплекса не совпадают с результатами DISCOVER. Тем не менее адекватность применения разработанного программного комплекса для решения задачи оценки эффективности различных алгоритмов распределенного видеокodeирования является обоснованной, в силу того, что производительность данной модели с точки зрения критерия «скорость-искажение» сравнима (или даже превосходит) с производительностью codeка на основе проекта DISCOVER.

Приведем краткое описание процесса сжатия видеоинформации в разработанной программной модели распределенного видеокodeка. Видеопоследовательность, представляющая собой последовательность кадров в формате YUV 4:2:0, разбивается на две подпоследовательности:

- базовые (ключевые, опорные) кадры;
- промежуточные кадры.

В рамках разработанной программной модели поддерживается только обработка группы кадров длины 2 ($GOP=2$). Т. е. если нумерация кадров в последовательности начинается с 1, то каждый второй кадр является промежуточным.

4.4.1 Обработка базовых кадров

Для обработки базовых кадров в разработанном кодеке используется эталонная реализация стандарта H.264/AVC – JM 9.5 [62]. Выбор версии 9.5 обусловлен тем фактом, что именно данная версия программного кода используется в кодеке DISCOVER. Обработка осуществляется в режиме INTRA, т. е. без устранения межкадровой избыточности.

Возможна обработка 8-битных видеопоследовательностей двух возможных разрешений: CIF (352×288) и QCIF (176×144).

Для обеспечения воспроизводимости полученных результатов, параметры кодека базовых кадров приведены в приложении Б.

По аналогии с кодеком DISCOVER в разработанном программном комплексе зафиксировано 8 возможных уровней качества сжатия, параметры квантования (QP) для каждой тестовой последовательности для каждого уровня качества заданы в соответствии с таблицей 4.1. Данные параметры используются для всех тестовых разрешений. Следует отметить, что в кодеке DISCOVER указаны значения шагов квантования для последовательностей Coastguard, Foreman, Hall и Soccer. Для последовательности Football предлагается использовать те же параметры, что и для Soccer.

4.4.2 Кодер промежуточных кадров

Каждый промежуточный кадр кодируется независимо от остальных, обрабатывается только яркостная составляющая. На стороне кодера выполняются следующие действия:

Параметры квантования (QP) ключевых кадров для тестовых последовательностей

Последовательность	Уровень качества							
	1	2	3	4	5	6	7	8
Coastguard	38	37	37	34	33	31	30	26
Foreman	40	39	38	34	34	32	29	25
Hall	37	36	36	33	33	31	29	24
Soccer	44	43	41	36	36	34	31	25
Football	44	43	41	36	36	34	31	25

1. **Блочное дискретное косинусное преобразование.** Кадр разбивается на непересекающиеся блоки размером 4×4 пикселя. Каждый блок $\mathbf{B}^{(p)}$, где p – индекс блока, подвергается целочисленному преобразованию, определяемому как:

$$\mathbf{T}^{(p)} = (\mathbf{C}\mathbf{B}^{(p)}\mathbf{C}^T) \circ \mathbf{E},$$

где \circ – поэлементное умножение матриц и

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix},$$

$$a = 0.5,$$

$$b = \sqrt{\frac{2}{5}}.$$

Полученные коэффициенты далее собираются в 16 групп:

$$\mathbf{g}^{(k,l)} = \left(\mathbf{T}^{(p)}[k,l] \right), p = 1, 2, \dots, \frac{wh}{16},$$

где $\mathbf{T}^{(p)}[k,l]$ – элемент, находящийся на позиции (k,l) в блоке спектральных коэффициентов $\mathbf{T}^{(p)}$, w и h – ширина и высота кадра соответственно. Таким образом, для кадра с разрешением CIF в каждой группе содержится 6336 спектральных коэффициентов, а для QICF – 1584 спектральных коэффициента.

2. **Квантование.** Полученные спектральные коэффициенты далее подвергаются скалярному квантованию. Шаг квантования рассчитывается для каждой группы исходя из заранее заданной матрицы квантования (4.3) для данного уровня качества и динамического диапазона данных.

$$\begin{aligned}
 \mathbf{Q}^{(1)} &= \begin{bmatrix} 16 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{Q}^{(2)} &= \begin{bmatrix} 32 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{Q}^{(3)} &= \begin{bmatrix} 32 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{Q}^{(4)} &= \begin{bmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix} & \mathbf{Q}^{(5)} &= \begin{bmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \\ 4 & 4 & 0 & 0 \end{bmatrix} & \mathbf{Q}^{(6)} &= \begin{bmatrix} 64 & 16 & 8 & 8 \\ 16 & 8 & 8 & 4 \\ 8 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{bmatrix} \\
 \mathbf{Q}^{(7)} &= \begin{bmatrix} 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \end{bmatrix} & \mathbf{Q}^{(8)} &= \begin{bmatrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{bmatrix} & & (4.3)
 \end{aligned}$$

Коэффициенты DC подвергаются равномерному скалярному квантованию в диапазоне $[0, 2^{11})$ (для 8-битного входа) с числом квантов, равным левому верхнему значению в соответствующей матрице квантования 4.3. Для AC коэффициентов применяется скалярное квантование с удвоенным интервалом вокруг нуля. Шаг квантования в группе (i,j) для уровня качества q определяется как

$$\Delta^{(k,l)} = \left\lceil \frac{2m_{k,l}}{\mathbf{Q}^{(q)}[k,l]} \right\rceil, \text{ если } \mathbf{Q}^{(q)}[k,l] \neq 0.$$

где $m_{k,l}$ – максимальное по модулю значение в группе (k,l) . Если $\mathbf{Q}^{(q)}[k,l] = 0$, то спектральные коэффициенты в группе не квантуются и данные о них далее на стороне кодера не обрабатываются.

Таким образом, индексы квантов $q_i^{(k,l)}$ в группе (k,l) после квантования занимают $\log_2(\mathbf{Q}^{(q)}[k,l])$ бит.

3. **Формирование битовых плоскостей.** Индексы квантов $q_{(k)}^{(i,j)}$ разбиваются на битовые плоскости таким образом, что биты одной значимости попадают в одну плоскость. Обозначим j -й по значимости бит спектрального коэффициента, находящегося на позиции i в группе (k,l) , через $b_j^{(k,l,i)}$. Для определенности будем считать, что наиболее значимая битовая плоскость имеет индекс 1.
4. **Помехоустойчивое кодирование.** Биты индексов квантов кодируются LDPCA кодом, начиная с наиболее значимой битовой плоскости в позиции наиболее значимого спектрального коэффициента. Накопленный синдром сохраняется в промежуточном буфере.

Таким образом, кодер промежуточных кадров должен передать декодеру следующую информацию.

- Один раз для всей последовательности: размеры кадра, параметр качества.
- Для каждого кадра: динамический диапазон данных в каждой группе спектральных коэффициентов, кроме DC.

Следует отметить, что т. к. в данной работе не проводились исследования с целью снижения количества запросов декодером проверочных бит, в разработанном кодеке не используется модуль управления битовой скоростью на стороне кодера, присутствующий в схеме DISCOVER.

4.4.3 Декодер промежуточных кадров

В разработанном программном комплексе на стороне декодера в процессе восстановления промежуточных кадров выполняются следующие действия.

1. **Генерация дополнительной информации.** Алгоритм генерации дополнительной информации включает в себя следующие действия.

- (a) Выполняется процедура предсказания промежуточного кадра с использованием алгоритма временной интерполяции кадров, описанного в подразделе 2.3.
- (b) К аппроксимирующему кадру применяется блоковое дискретное косинусное преобразование (аналогично шагу 1 на кодере).
- (c) Квантование спектральных коэффициентов (аналогично шагу 2 на кодере). Следует отметить, что т. к. после дискретного косинусного преобразования спектральные коэффициенты могут выйти за границы динамического диапазона, рассчитанного на кодере, перед квантованием выполняется клипирование данных:

$$\hat{g}_i^{k,l} = \begin{cases} m_{k,l}, & \text{если } \hat{g}_i^{k,l} \geq m_{k,l} \\ -m_{k,l}, & \text{если } \hat{g}_i^{k,l} < -m_{k,l} \\ \hat{g}_i^{k,l}, & \text{иначе} \end{cases},$$

где $m_{k,l}$ – число, определяющее динамический диапазон данных (получено от кодера), $\hat{g}_i^{k,l}$ – оцененный декодером спектральный коэффициент, находящийся на позиции i в группе (k,l) .

- (d) Выделение битовых плоскостей (аналогично шагу 3 на кодере).

Таким образом, дополнительной информацией декодера при восстановлении промежуточного кадра являются:

- спектральные коэффициенты $\hat{g}_i^{k,l}$, рассчитанные по аппроксимирующему кадру;
- биты $\hat{b}_j^{(k,l,i)}$ квантованных спектральных коэффициентов $\hat{g}_i^{k,l}$.

2. **Оценка параметров ошибок межкадрового предсказания.** Для каждого квантованного спектрального коэффициента $\hat{g}_i^{k,l}$ выполняется оценка параметра масштаба распределения Лапласа $\hat{\alpha}_i^{(k,l)}$. В рамках программного комплекса используется алгоритм моделирования виртуального канала из кодека DISCOVER (раздел 3.2).

3. **Помехоустойчивое декодирование.** В данном блоке предлагается использовать результаты проекта Stanford DVC. Декодер LDPCA кода работает по схеме с мягким входом. Расчет мягкого входа осуществляется для каждого бита в каждой битовой плоскости с учетом ранее декодированных бит и значений спектральных коэффициентов дополнительной информации. Пусть битовые плоскости с номерами $1, 2, \dots, j-1$ уже декодированы. Мягкий вход декодера для бита, находящегося на позиции i в битовой плоскости j рассчитывается как логарифм отношения правдоподобия

$$\begin{aligned} LLR_j^{(k,l,i)} &= \ln \left(\frac{P \left\{ b_j^{(k,l,i)} = 0 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\}}{P \left\{ b_j^{(k,l,i)} = 1 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\}} \right) = \\ &= \ln \left(\frac{P \left\{ b_j^{(k,l,i)} = 0 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\}}{1 - P \left\{ b_j^{(k,l,i)} = 0 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\}} \right). \end{aligned}$$

Таким образом, для того, чтобы рассчитать $LLR_j^{(k,l,i)}$, необходимо определить условную вероятность $P \left\{ b_j^{(k,l,i)} = 0 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\}$. Эта вероятность рассчитывается следующим образом:

$$P \left\{ b_j^{(k,l,i)} = 0 \mid b_{j-1}^{(k,l,i)}, \dots, b_1^{(k,l,i)}, \hat{g}_i^{(k,l)} \right\} = \frac{\int_{l_0}^{l_1} f_j^{(k,l)}(x) dx}{\int_{l_0}^{l_2} f_j^{(k,l)}(x) dx},$$

где $f(x)$ – функция плотности вероятности распределения ошибок, центрированная вокруг $\hat{g}_i^{(k,l)}$; значения l_0 , l_1 и l_2 определяют границы интегрирования с учетом ранее декодированных бит в позиции данного коэффициента:

$$l_0 = \Delta^{(k,l)} \mathbf{Q}^{(a)}[k,l] \sum_{j'=1}^{j-1} \frac{b_{j'}^{(k,l,i)}}{2^{j'}} - m_{k,l},$$

$$l_1 = l_0 + \Delta^{(k,l)} \sum_{j'=j+1}^{\log_2(\mathbf{Q}^{(a)}[k,l])} 2^{j'},$$

$$l_2 = l_1 + \Delta^{(k,l)} 2^j.$$

Для модели шума, основанной на распределении Лапласа, расчет интегралов можно выполнить численно с использованием следующих соотношений:

$$\int_{l_b}^{l_t} f(x) dx = \begin{cases} 0.5 \left(e^{\hat{\alpha}_i^{(k,l)}(l_t - \hat{g}_i^{(k,l)})} - e^{\hat{\alpha}_i^{(k,l)}(l_b - \hat{g}_i^{(k,l)})} \right), & \text{если } l_t \leq \hat{g}_i^{(k,l)} \\ 0.5 \left(e^{\hat{\alpha}_i^{(k,l)}(\hat{g}_i^{(k,l)} - l_t)} - e^{\hat{\alpha}_i^{(k,l)}(\hat{g}_i^{(k,l)} - l_b)} \right), & \text{если } l_b \geq \hat{g}_i^{(k,l)} \\ 1 - 0.5 \left(e^{\hat{\alpha}_i^{(k,l)}(l_b - \hat{g}_i^{(k,l)})} + e^{\hat{\alpha}_i^{(k,l)}(\hat{g}_i^{(k,l)} - l_t)} \right), & \text{иначе} \end{cases}$$

4. **Восстановление индексов квантов.** Декодированные биты собираются по плоскостям, и формируются квантованные спектральные коэффициенты $q_i^{(k,l)}$:

$$q_i^{(k,l)} = \mathbf{Q}^{(q)}[k,l] \sum_{j=1}^{\log_2(\mathbf{Q}^{(q)}[k,l])} b_j^{(k,l,i)} 2^{-j}, \quad \forall (k,l) : \mathbf{Q}^{(q)}[k,l] \neq 0.$$

5. **Деквантование спектральных коэффициентов.** В данном модуле выполняется операция деквантования с учетом спектральных коэффициентов, рассчитанных по аппроксимирующему кадру [63]. Обозначим значение деквантованного спектрального коэффициента, находящегося на позиции i в группе (k,l) через $\tilde{g}_i^{(k,l)}$, тогда

$$\tilde{g}_i^{(k,l)} = \begin{cases} z_{q_i^{(k,l)}} + \frac{1}{\hat{\alpha}_i^{(k,l)}} + \frac{\Delta^{(k,l)}}{1 - e^{\hat{\alpha}_i^{(k,l)} \Delta^{(k,l)}}}, & \text{если } \hat{g}_i^{(k,l)} < z_{q_i^{(k,l)}} \\ z_{q_i^{(k,l)}+1} - \frac{1}{\hat{\alpha}_i^{(k,l)}} - \frac{\Delta^{(k,l)}}{1 - e^{\hat{\alpha}_i^{(k,l)} \Delta^{(k,l)}}}, & \text{если } \hat{g}_i^{(k,l)} \geq z_{q_i^{(k,l)}+1} \\ \hat{g}_i^{(k,l)} + \frac{\left(\gamma + \frac{1}{\hat{\alpha}_i^{(k,l)}} \right) e^{-\hat{\alpha}_i^{(k,l)} \gamma} - \left(\delta + \frac{1}{\hat{\alpha}_i^{(k,l)}} \right) e^{-\hat{\alpha}_i^{(k,l)} \delta}}{2 - \left(e^{-\hat{\alpha}_i^{(k,l)} \gamma} + e^{-\hat{\alpha}_i^{(k,l)} \delta} \right)}, & \text{иначе} \end{cases},$$

где $\gamma = \hat{g}_i^{(k,l)} - z_{q_i^{(k,l)}}$, $\delta = z_{q_i^{(k,l)}+1} - \hat{g}_i^{(k,l)}$, $z_{q_i^{(k,l)}}$ и $z_{q_i^{(k,l)}+1}$ – границы кванта с номером $q_i^{(k,l)}$, $\hat{g}_i^{(k,l)}$ – соответствующий спектральный коэффициент, рассчитанный по аппроксимирующему кадру.

В том случае, если данные о группе (k,l) декодеру не передаются, то используется значение коэффициента дополнительной информации:

$$\tilde{g}_i^{(k,l)} = \hat{g}_i^{(k,l)}, \quad \forall (k,l) : \mathbf{Q}^{(q)}[k,l] = 0.$$

Из восстановленных таким образом 16 групп спектральных коэффициентов далее формируются 4×4 блоки $\tilde{\mathbf{T}}^{(p)}$.

6. **Обратное дискретное косинусное преобразование.** Восстановленные на предыдущем шаге спектральные коэффициенты далее подвергаются обратному дискретному косинусному преобразованию с целью восстановления промежуточного кадра. Из каждого блока спектральных коэффициентов $\tilde{\mathbf{T}}^{(p)}$ восстанавливается соответствующий блок кадра $\tilde{\mathbf{V}}^{(p)}$ как

$$\tilde{\mathbf{V}}^{(p)} = \mathbf{C}^T (\tilde{\mathbf{T}}^{(p)} \circ \mathbf{E}) \mathbf{C},$$

где \circ – поэлементное умножение матриц и

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0.5 & -0.5 & -1 \\ 1 & -1 & -1 & 1 \\ 0.5 & -1 & 1 & -0.5 \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix},$$

$$a = 0.5,$$

$$b = \sqrt{\frac{2}{5}}.$$

Последней операцией на стороне декодера является восстановление порядка следования кадров в выходной видеопоследовательности.

4.4.4 Сравнение разработанного эталонного распределенного кодека с кодеком на основе базовой модели DISCOVER

Перед тем, как переходить к сравнению разработанных алгоритмов обработки видеоинформации в рамках реализованной программной модели, приведем результаты её сравнения с кодеком DISCOVER. Как было отмечено ранее, получение результатов, в точности совпадающих с результатами DISCOVER, не представляется возможным в силу того, что нет полного и точного описания алгоритмов, реализованных в данном кодеке, и их параметров. Однако, т. к. разработанная программная модель максимально близко повторяет реализацию DISCOVER, данные кодеки должны демонстрировать схожую эффективность сжатия.

Сравнение проводилось с использованием стандартного тестового множества видеопоследовательностей, используемого в проекте DISCOVER. Результаты сравнения приведены на рисунке 4.1. Из приведенных графиков видно, что в целом кодеки показывают приблизительно одинаковые результаты. На трех последовательностях из четырех разработанная программная модель незначительно превзошла кодек DISCOVER, на одной последовательности наблюдается небольшой проигрыш. Таким образом, можно сделать вывод, что кодек, описанный в подразделе 4.4, можно использовать в качестве базового для проведения сравнительного анализа алгоритмов, разработанных в данной диссертационной работе.

4.5 Схема эксперимента для сравнительной оценки разработанных алгоритмов

4.5.1 Общие замечания по сравнению алгоритмов обработки видеоинформации в схеме распределенного видеокодирования

В настоящее время для проведения сравнительной оценки различных алгоритмов в схеме DVC используется следующий подход [27]. Реализуется полная схема распределенного кодека, в рамках которой все алгоритмы во всех модулях

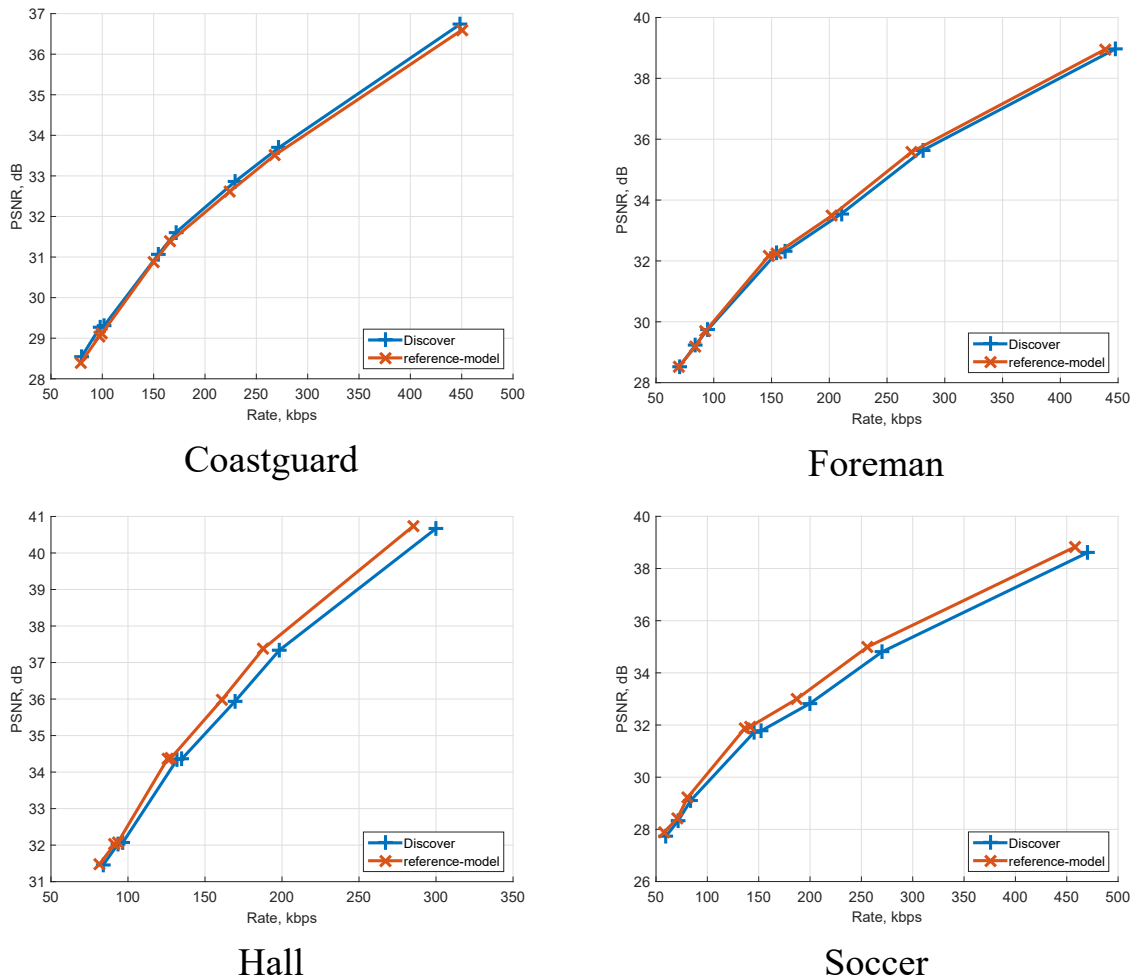


Рисунок 4.1 — Сравнение разработанного программного комплекса и кодека DISCOVER по кривым «скорость-искажение»

считаются зафиксированными. Подобный кодек в дальнейшем считается эталонным. Далее в одном из модулей кодека алгоритм изменяется на тестируемый. Получившийся модифицированный кодек считается тестовым. Эталонный и тестовый кодек прогоняются на одном наборе видеопоследовательностей и полученные результаты сравниваются по кривым «скорость-искажение»: считается, что чем выше идет кривая, тем лучше работает оцениваемый алгоритм в схеме DVC.

Основная проблема, возникающая при использовании подобного подхода, заключается в том, что модули кодека работают совместно в рамках сложной системы, обладающей высокой эмерджентностью. Например, если оценивать эффективность алгоритма генерации дополнительной информации с использованием полной схемы кодирования-декодирования и кривых «скорость-искажение», существенное влияние на итоговый результат оказывают также модуль моделиро-

вания виртуального канала, характеристики помехоустойчивого кодирования, метод управления битовой скоростью, процедура восстановления кадров и т. д. Модуль оценки параметров ошибок межкадрового предсказания в свою очередь принимает на вход аппроксимацию ошибок, рассчитываемую на стороне декодера в модуле генерации дополнительной информации. Наличие подобных зависимостей означает, что сравнительная оценка и демонстрация эффективности разработанных алгоритмов должна выполняться с учетом комплексного влияния всех модулей распределенного видеокодека как друг на друга, так и на итоговый результат. В связи с этим в дальнейших подразделах для каждого разработанного в рамках данной диссертационной работы алгоритма приведено описание и результаты ряда экспериментов, которые необходимо выполнить для того, чтобы продемонстрировать выигрыш от применения данного алгоритма в схеме DVC. В завершение экспериментов приведено сравнение распределенного видеокодека, включающего оба разработанных алгоритма, с кодеком DISCOVER, считающимся базовой реализацией распределенного видеокодека, а также с существующими стандартными алгоритмами сжатия видеоданных.

4.5.2 Сравнительная оценка алгоритмов генерации дополнительной информации

4.5.2.1 Методы оценки алгоритмов генерации дополнительной информации

Как было показано в разделе 2 данной диссертационной работы, в основе алгоритмов генерации дополнительной информации лежит процедура межкадрового предсказания, основанная на временной интерполяции. Чем точнее выполнена интерполяция промежуточных кадров, т. е. чем больше аппроксимирующий кадр на стороне декодера похож на соответствующий кадр на кодере, тем меньше ошибок межкадрового предсказания будет содержаться в дополнительной информации декодера. Таким образом, для проведения сравнительного анализа алгоритмов генерации дополнительной информации достаточно сравнить результаты временной интерполяции, полученные с использованием базового алгоритма

(подраздел 2.3), используемого в программном комплексе, и разработанного алгоритма (подраздел 2.5). Следует отметить, что, учитывая приведенные в подразделе 4.5.1 наблюдения, данное сравнение целесообразно проводить с использованием экспериментов двух типов.

1. Сравнение без учета особенностей распределенного кодека.
2. Сравнение в рамках распределенного кодека.

Эксперимент первого типа позволяет судить об алгоритме межкадрового предсказания в целом. Если на данном этапе не наблюдается выигрыша по сравнению с эталонным алгоритмом, то с высокой вероятностью использование данного алгоритма не даст выигрыша и в схеме DVC. Эксперимент второго типа позволяет получить понимание и приблизительные оценки выигрыша/проигрыша от применения тестируемого алгоритма в рамках DVC. Следует отметить, что при разработке эксперимента второго типа особое внимание должно быть уделено вопросам устранения взаимного влияния между модулями распределенного кодека.

4.5.2.2 Оценка алгоритма межкадрового предсказания без учета особенностей распределенного кодека

Для осуществления оценки без учета особенностей распределенного кодека используются существующие методы сравнения алгоритмов временной интерполяции, например в рамках задачи преобразования кадровой скорости, позволяющие оценить визуальное качество интерполяции независимо от того, как данные алгоритмы используются в схеме DVC.

Наиболее распространенным методом оценки алгоритмов временной интерполяции является следующий. Формируется множество тестовых видеопоследовательностей и каждая последовательность прореживается, например, выкидыванием каждого второго кадра. Далее для восстановления выкинутых кадров к каждой паре оставшихся смежных кадров применяется алгоритм временной интерполяции. Затем оригинальные выкинутые кадры сравниваются с интерполированными.

Графики со сравнением покадровых значений критерия PSNR для всех тестовых последовательностей приведены на рисунке 4.2. Средние значения PSNR, рассчитанные по всем интерполированным кадрам, сведены в таблице 4.2.

Таблица 4.2

Среднее значение критерия PSNR для промежуточных кадров

Последовательность	Базовый алгоритм	Предложенный алгоритм
Coastguard	32.13	34.66
Football	22.84	23.96
Foreman	32.55	34.25
Hall	36.89	37.35
Soccer	25.13	27.66

Из приведенных результатов видно, что разработанный алгоритм, описанный в подразделе 2.5, показывает в целом более высокое качество интерполяции, чем базовый алгоритм. Кроме того, следует отметить тот факт, что величина выигрыша зависит от характеристик видеопоследовательности. Для того, чтобы пояснить данный вывод введем в рассмотрение понятие «сложности движения». Под сложностью движения будем понимать качественную характеристику, зависящую от интенсивности и направления перемещений объектов в видеопоследовательности: чем быстрее и хаотичнее перемещаются объекты, тем более сложное движение. Отсортируем последовательности в соответствии с данным критерием, а также оценим для каждой последовательности выигрыш от применения разработанного алгоритма генерации дополнительной информации с помощью критерия BD-PSNR, позволяющего оценивать разницу между двумя кривыми «скорость-искажение» [60]. Результаты данного анализа приведены в таблице 4.3.

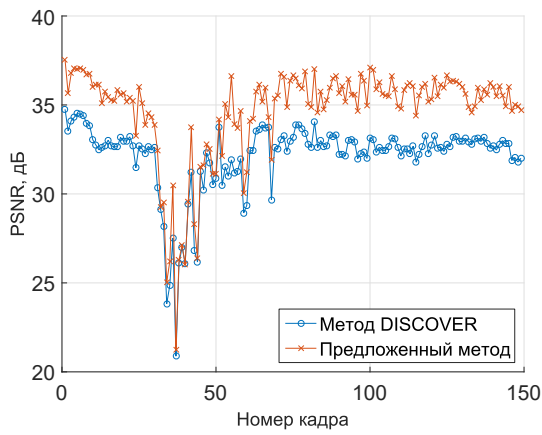
Зависимость выигрыша от сложности движения

Последовательность	Сложность движения	Разница в значении PSNR, дБ
Coastguard	2	2.53
Football	4	1.12
Foreman	3	1.7
Hall	1	0.46
Soccer	4	2.53

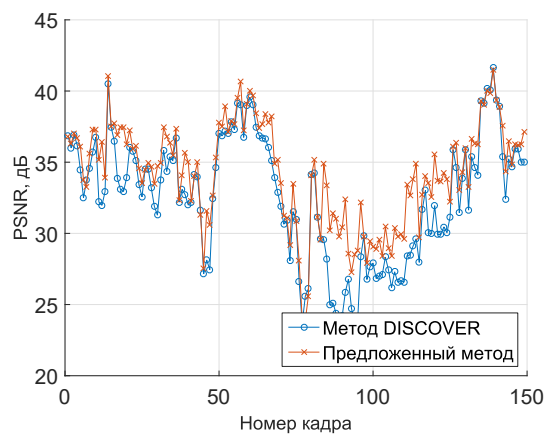
Проанализировав данные в таблице 4.3, можно сделать вывод, что чем более сложное движение в видеопоследовательности, тем больше выигрыш от применения разработанного алгоритма генерации дополнительной информации. Полученный результат является ожидаемым, т. к. на последовательностях с сравнительно простым движением точность временной интерполяции довольно высока у многих алгоритмов, т. е. число ошибок в дополнительной информации относительно невелико. По мере роста сложности движения, растет и число ошибок. Полученные результаты показывают, что число ошибок в разработанном алгоритме возрастает медленнее, по сравнению с алгоритмом, реализованным в кодеке DISCOVER. Это объясняется тем, что данный алгоритм учитывает модель истинного движения при поиске оптимального векторного поля лучше чем подход, реализованный в DISCOVER.

4.5.2.3 Оценка алгоритма межкадрового предсказания с использованием распределенного кодека

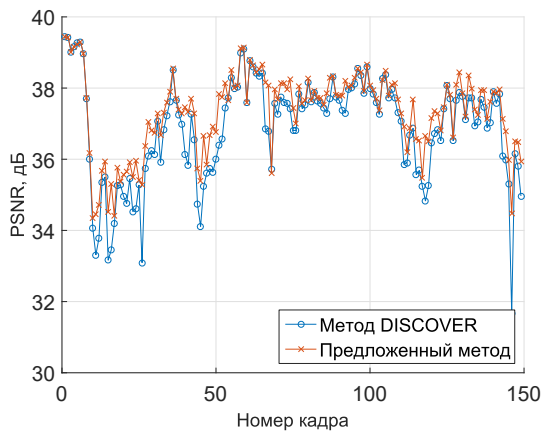
При оценке алгоритма межкадрового предсказания с использованием распределенного кодека необходимо учесть наблюдения, указанные в подразделе 4.5.1, а именно тот факт, что модуль генерации дополнительной информации находится в тесной взаимосвязи с прочими модулями кодека. В настоящей диссертационной работе для решения данной проблемы предлагается использовать следующий подход. Реализуется видеокodeк, работающий по схеме, описанной в работах [64], [5]. Опорные кадры обрабатываются независимо, например в соот-



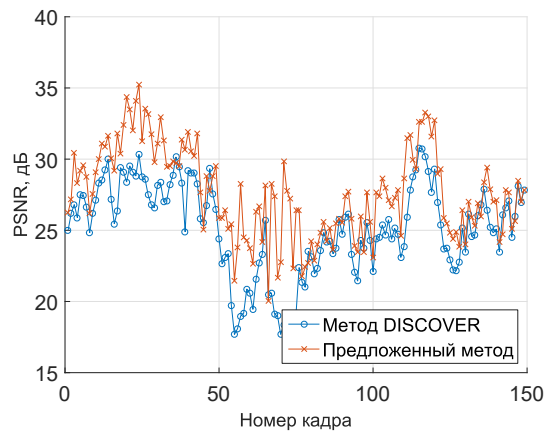
Coastguard



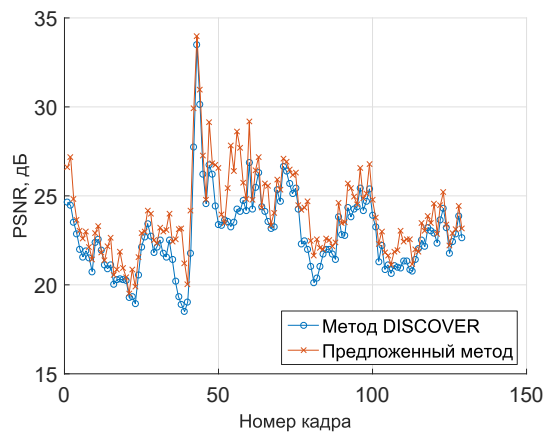
Foreman



Hall



Soccer



Football

Рисунок 4.2 — Графики PSNR для интерполированных кадров

ветствии с алгоритмом, используемым в кодеке DISCOVER. По восстановленным опорным кадрам выполняется межкадровое предсказание промежуточного кадра, результат предсказания выдается в выходной поток в качестве восстановленного промежуточного кадра. Помехоустойчивое кодирование для исправления ошибок в дополнительной информации при этом не выполняется, т. е. промежуточные кадры обрабатываются только на стороне декодера и дополнительные биты от кодера не требуются. Таким образом, в подобной схеме битовые затраты на передачу промежуточных кадров составляют 0 бит, а в качестве восстановленных кадров выдаются аппроксимирующие кадры, полученные в процессе генерации дополнительной информации. Схема кодека без обратной связи приведена на рисунке 4.3. Суммарные битовые затраты на сжатие всей последовательности определяются только затратами на хранение информации об опорных кадрах. Изменяя степень сжатия опорных кадров, строятся кривые «скорость-искажение», по которым осуществляется сравнительная оценка различных методов алгоритмов временной интерполяции.

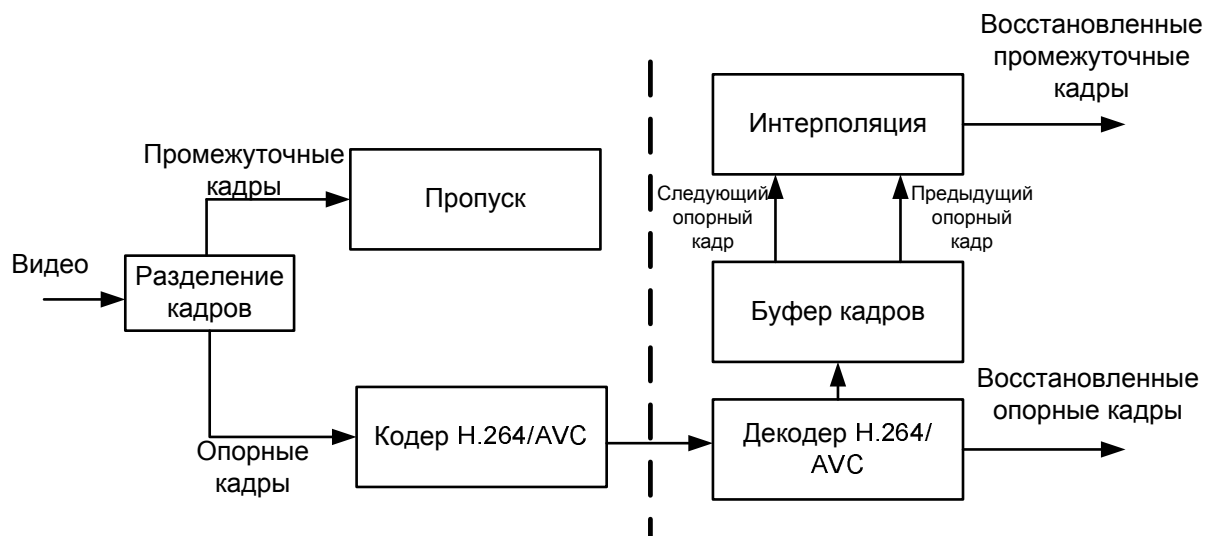


Рисунок 4.3 — Структурная схема видеокodeка для проведения сравнительной оценки алгоритмов генерации дополнительной информации

Кривые «скорость-искажение», построенные с использованием описанного метода, приведены на рисунке 4.4. Видно, что разработанный алгоритм межкадрового предсказания выигрывает у базового алгоритма на всех тестовых последовательностях. При этом, как правило, чем более сложное движение в видео, тем более существенен выигрыш.

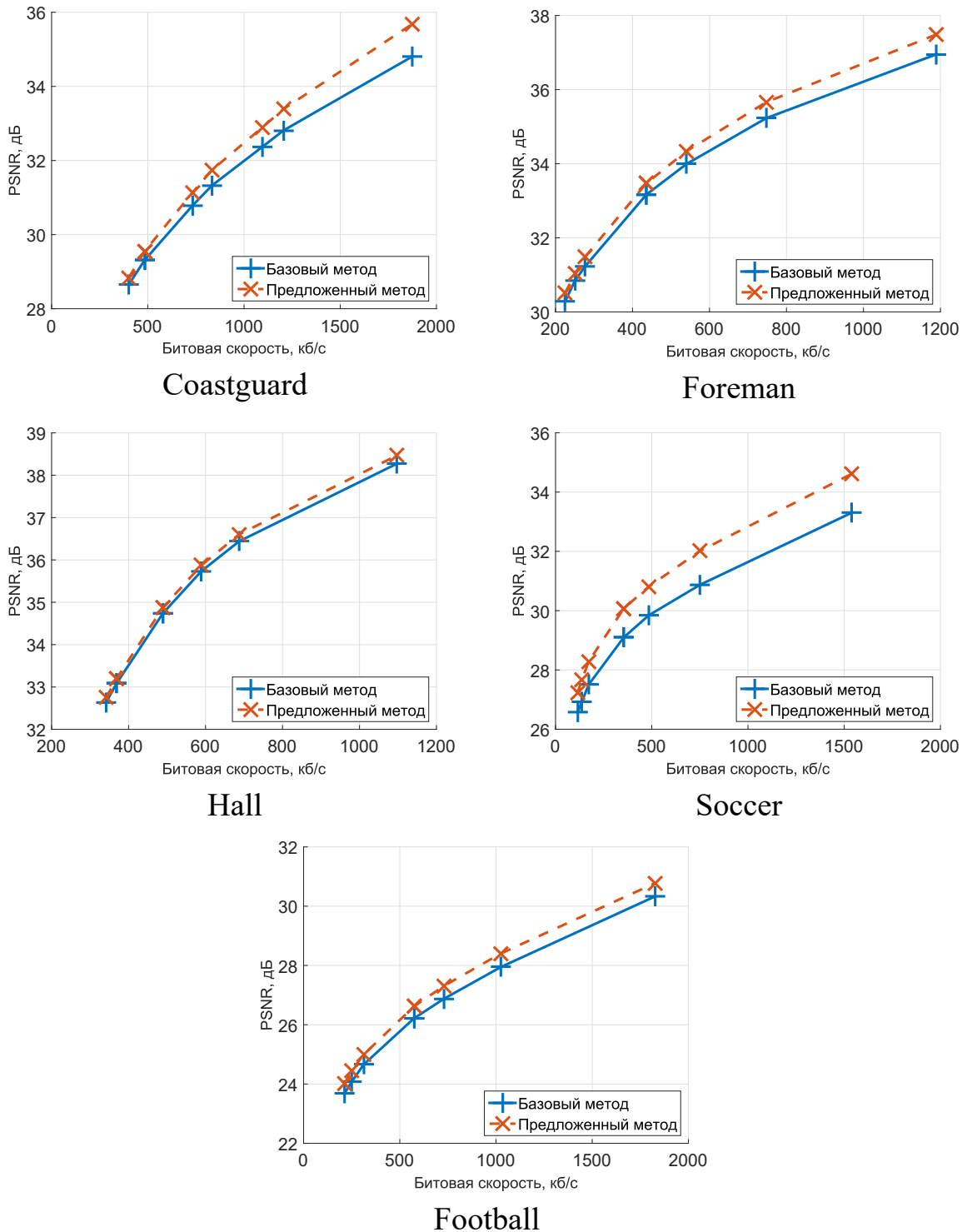


Рисунок 4.4 — Результаты сравнения алгоритмов генерации дополнительной информации в рамках распределенного кодека

Следует отметить, что т. к. дополнительные биты со стороны кодера для восстановления промежуточных кадров не передаются, кривые «скорость-искажение» для упрощенной модели без обратной связи можно построить только для низкого и среднего качества сжатия. Исправление ошибок позволит продлить этот график в область высокого качества за счет повышения битовой скорости.

4.5.3 Сравнение алгоритмов оценки параметров ошибок межкадрового предсказания

При сравнении алгоритмов оценки параметров ошибок межкадрового предсказания необходимо учесть два факта:

- способ аппроксимации ошибок межкадрового предсказания на стороне декодера в модуле генерации дополнительной информации;
- влияние рассчитанных надежностей символов на последующие процедуры восстановления кадра.

Рассмотрим данные факты подробнее. Исследование второго факта, связанного с определением влияния рассчитанных надежностей символов на последующее восстановление кадра, включающее помехоустойчивое декодирование и оптимальное восстановление спектральных коэффициентов, в настоящее время является открытой задачей. На данный момент нет работ, связывающих в какой-либо форме точность оценки параметров ошибок с числом бит, необходимых для их исправления, и, как следствие, качеством восстановленного кадра. Однако, можно сделать разумное предположение, что чем точнее выполнена оценка параметров, тем меньше бит необходимо декодеру для исправления ошибок, и тем выше будет качество восстановленного кадра. Таким образом, будем далее предполагать, что влиянием модуля, выполняющего помехоустойчивое кодирование, при сравнении алгоритмов моделирования виртуального канала можно пренебречь.

Рассмотрим вопросы, связанные с влиянием способа аппроксимации ошибок межкадрового предсказания. В базовой работе [52] по моделированию виртуального канала, а также в ряде других работ, например [65], в качестве аппроксимации ошибок межкадрового предсказания в спектральной области предлага-

ется использовать значения преобразованных в спектральную область разностей сопоставления блоков, которые рассчитываются в процедуре оценки движения (подраздел 3.2.2). Следует отметить, что недостатком такого подхода является тот факт, что малое значение ошибки сопоставления блоков не гарантирует отсутствие искажений на аппроксимирующем кадре, и наоборот. Более того, как было показано в подразделе 2.4, векторное поле, отражающее истинное движение объектов, является в некотором смысле компромиссом между гладкостью векторов движения и ошибкой сопоставления блоков. Также было отмечено, что малая энергия разностного кадра, которую можно обеспечить с помощью полного перебора в некотором радиусе всех возможных векторов-кандидатов, как правило не соответствует истинному движению. Таким образом, для проведения сравнительного анализа алгоритмов моделирования виртуального канала представляется целесообразным проводить сравнение с использованием экспериментов двух типов.

1. Сравнение кривых «скорость-искажение», полученных для настоящих ошибок межкадрового предсказания.
2. Сравнение кривых «скорость-искажение», полученных для ошибок межкадрового предсказания, оцененных с использованием подхода, описанного в работе [52].

Эксперимент первого типа позволяет сделать вывод об эффективности алгоритмов моделирования виртуального канала, когда аппроксимация ошибок произведена безошибочно, т. е. $\hat{\mathbf{n}}^{(b)} \triangleq \mathbf{n}^{(b)}$. Следует отметить, что подобная ситуация является нереализуемой в реальной системе, т. к. для расчета $\mathbf{n}^{(b)}$ декодер должен заранее обладать информацией о восстановленном промежуточном кадре. Данный сценарий, хоть и нереализуемый на практике, позволяет сравнивать различные алгоритмы моделирования виртуального канала в идеальных условиях, предоставляя информацию о верхней границе для кривых «скорость-искажение», когда все прочие модули в декодере являются зафиксированными.

Эксперимент второго типа позволяет сравнивать эффективность различных алгоритмов моделирования виртуального канала в рамках реальной системы. Отметим, что данное сравнение делается в предположении, что выполнено допущение 3.2, приведенное в подразделе 3.2.1, т. е. рассчитанная декодером аппроксимация ошибок в некотором смысле «похожа» на реальные ошибки.

Перед тем, как переходить к результатам экспериментов, напомним, что в разработанном алгоритме при решении задачи разделения смеси на втором шаге выполняется поиск параметров законов распределений для заданных скрытых состояний и весов компонент. Будем считать, что функция плотности вероятности ошибок межкадрового предсказания аппроксимируется смесью распределений Лапласа. Данное допущение позволяет использовать при расчете мягкого входа помехоустойчивого декодера и при оптимальном восстановлении кадра те же формулы, что и при использовании базового алгоритма.

Напомним, что модифицированный алгоритм моделирования виртуального канала позволяет учесть свойство пространственной группировки ошибок, характерное в первую очередь для последовательностей со сложным движением. В связи с этим для проведения сравнительного анализа данного алгоритма будем использовать только последовательности Soccer и Football. Кривые «скорость-искажение», построенные для этих последовательностей в рамках экспериментов первого и второго типа приведены на рисунке 4.5 и 4.6 соответственно. Из приведенных графиков видно, что модифицированный алгоритм выигрывает у базового, однако, следует отметить три факта.

1. Основной выигрыш наблюдается в области высокого и среднего качества, на низком качестве кривые «скорость-искажение» почти совпадают. Это связано в первую очередь с тем, что на низкой битовой скорости существенное влияние на ошибку межкадрового предсказания вносит шум квантования, причем чем более грубое квантование используется, тем сильнее этот вклад. Данное влияние никак не учитывается в базовом и модифицированном алгоритме.
2. Значение выигрыша, полученное для экспериментов первого типа, существенно превышает значение выигрыша, полученное для экспериментов второго типа. Это означает, что на практике допущение о схожести аппроксимации ошибок на реальные значения ошибок выполняется не в полной мере.
3. Выигрыш от применения модифицированного алгоритма в целом незначителен. Данный факт связан с тем, что не совсем корректно аппроксимировать ошибки межкадрового предсказания в областях с низким качеством интерполяции распределением Лапласа с нулевым математическим ожиданием. Расширение алгоритма на другие распределения яв-

ляется предметом дальнейших исследований. В целом можно сделать вывод, что результаты, полученные для модифицированного алгоритма, подтверждают обоснованность концепции моделирования виртуального канала с использованием независимых регионов.

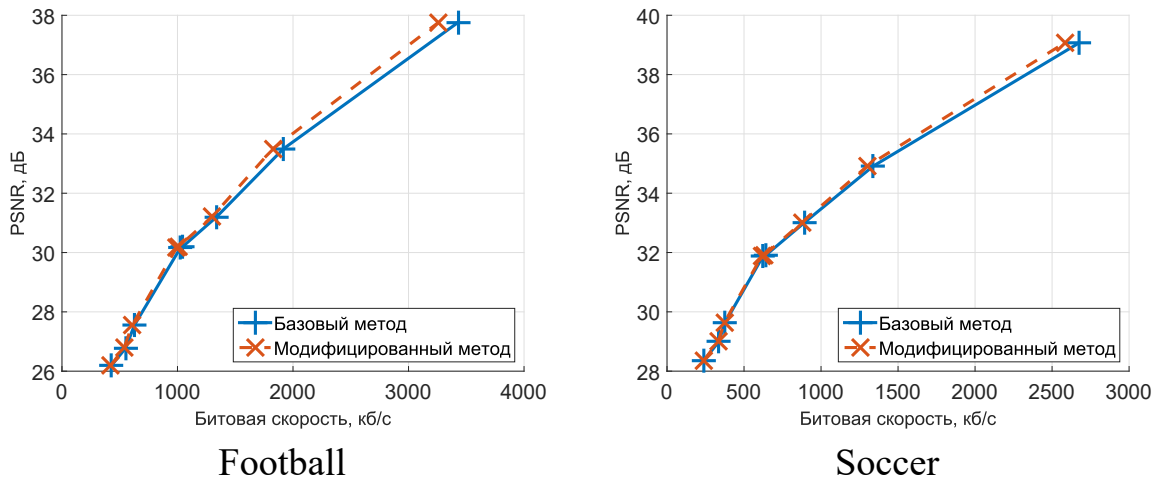


Рисунок 4.5 — Кривые «скорость-искажение», полученные для сравнения алгоритмов моделирования виртуального канала в рамках экспериментов первого типа

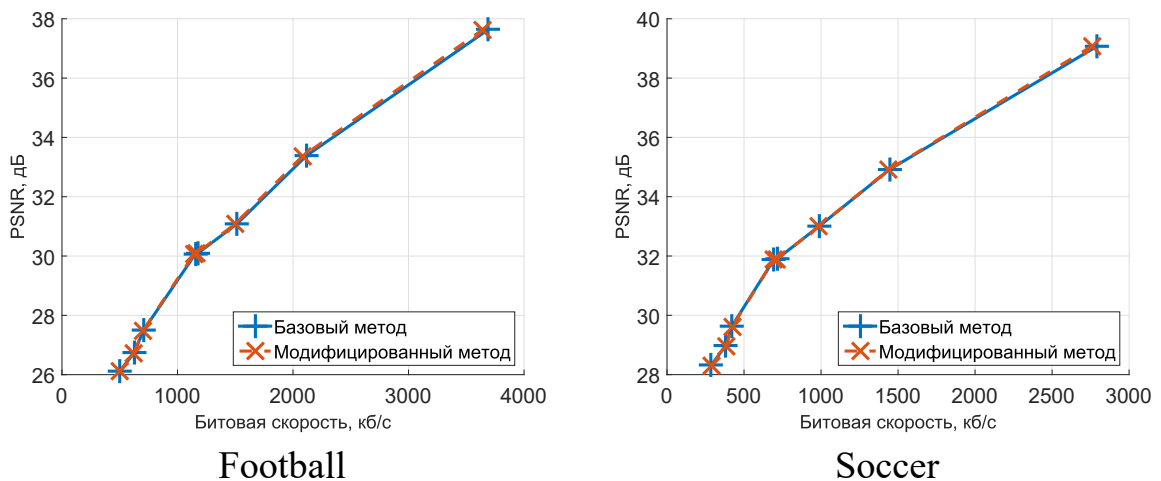


Рисунок 4.6 — Кривые «скорость-искажение», полученные для сравнения алгоритмов моделирования виртуального канала в рамках экспериментов второго типа

Таблица 4.4

Значение критерия BD-RATE для экспериментов первого типа

Последовательность	BD-Rate, %
Football	3.8
Soccer	2.2

Таблица 4.5

Значение критерия BD-RATE для экспериментов первого типа

Последовательность	BD-Rate, %
Football	1
Soccer	0.6

4.6 Сравнительный анализ разработанного алгоритма распределенного видеокодирования

В данном подразделе приведены результаты сравнения разработанной программной модели распределенного видеокodeка с существующими алгоритмами сжатия:

- кодек стандарта H.264, работающий в режиме Intra;
- кодек стандарта H.264, работающий в режиме Inter;
- кодек стандарта H.264, работающий в режиме Inter No Motion;
- кодек DISCOVER.

Во всех тестах в качестве кодека использовалась эталонная реализация JM версии 9.5. В режиме H.264 Intra все кадры обрабатывались как ключевые. В остальных режимах использовался режим IBI, т.е. каждый второй кадр обрабатывался как ключевой. Отметим, что в режиме H.264 Inter межкадровое предсказание и устранение временной избыточности осуществляется кодером, т. е. данную кривую можно рассматривать в качестве примера для совместного кодирования-декодирования зависимых источников.

Результаты сравнения по кривым «скорость-искажение» приведены на рисунке 4.7. На всех последовательностях из тестового множества разработанная программная модель выигрывает у кодека DISCOVER в среднем. При этом оба кодека проигрывают на всех последовательностях кодеку H.264 Inter. На после-

довательностях с высокой сложностью движения (Football и Soccer) и Discover, и разработанный алгоритм проигрывают как H.264 Inter, так и H.264 Intra. Однако следует отметить, что на высоком качестве для последовательности Soccer разработанный алгоритм, в отличие от DISCOVER, показывает качество, близкое к H.264 Intra. Результаты сравнения разработанного программного комплекса и кодека DISCOVER по критерию BD-RATE приведены в таблице 4.6. Можно сделать вывод, что применение разработанных алгоритмов позволяет снизить битовые затраты на 4 – 16%, причем чем сложнее движение в видеопоследовательности, тем больше выигрыш.

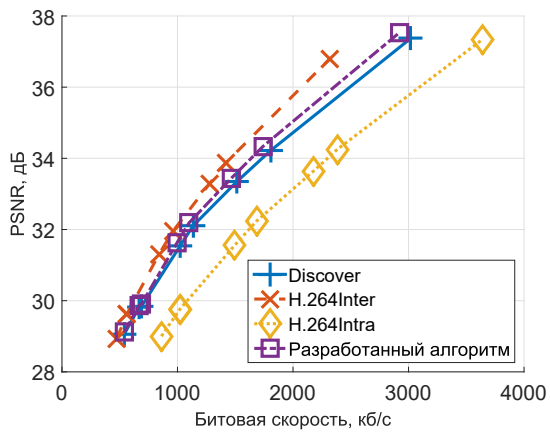
Таблица 4.6

Результаты сравнения реализованного программного комплекса и кодека DISCOVER по критерию BD-RATE

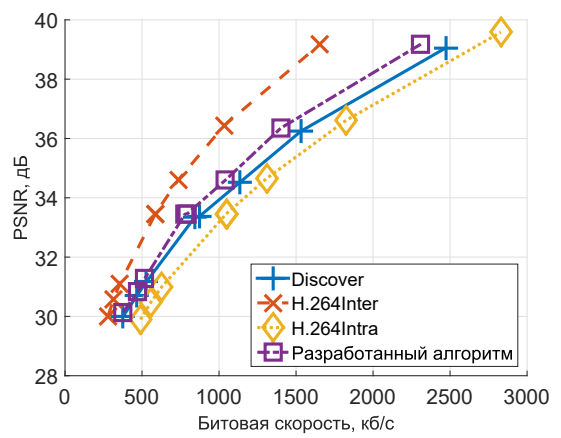
Последовательность	BD-Rate, %
Coastguard	4.6
Football	12.4
Foreman	8.7
Hall	4.3
Soccer	15.8

4.7 Выводы по разделу

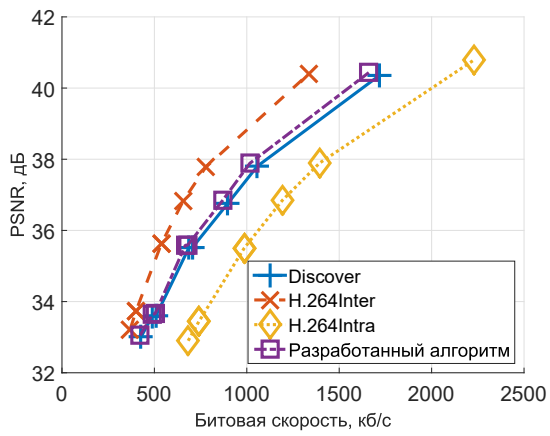
В данном разделе приведены результаты сравнительного анализа разработанных в настоящей диссертационной работе алгоритмов обработки видеoinформации в рамках системы сжатия, основанной на принципах кодирования зависимых источников. Было показано, что в настоящее время не существует открытой реализации распределенного кодека, с помощью которой можно было бы проводить сравнительный анализ различных алгоритмов генерации дополнительной информации декодера и моделирования виртуального канала. В связи с этим, на основе базовой модели распределенного кодирования DISCOVER, была разработана новая программная модель кодека, сравнимая по степени сжатия с кодеком, реализованным в рамках проекта DISCOVER. Были приведены основные особенности данной программной модели, а также решены задачи, связанные с внедрен-



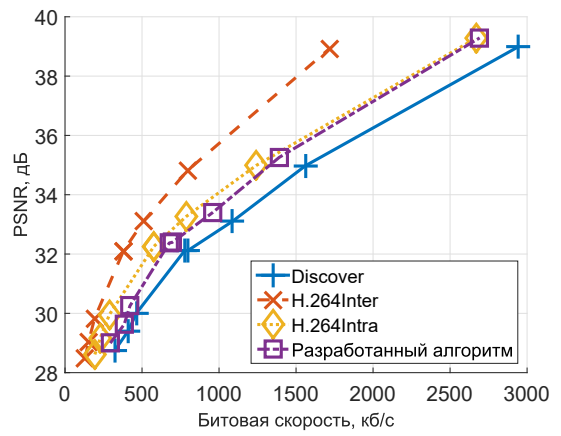
Coastguard



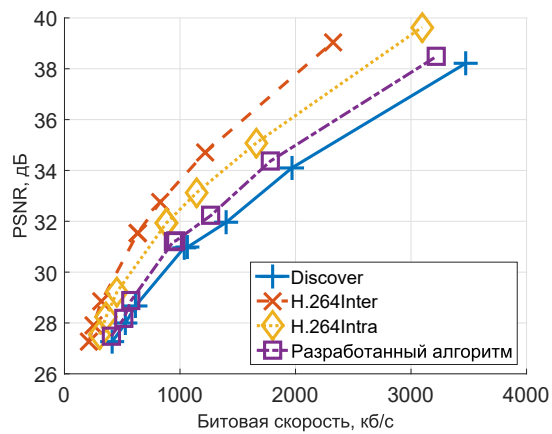
Foreman



Hall



Soccer



Football

Рисунок 4.7 — Сравнение с существующими алгоритмами сжатия

нием в данную модель алгоритмов, разработанных в разделах 2 и 3. Далее была описана методика выполнения сравнительного анализа алгоритмов, в рамках которой были указаны основные особенности, которые необходимо учесть при сравнении. Также, был предложен новый метод сравнительной оценки алгоритмов генерации дополнительной информации декодера, использующий на упрощенную модель кодека без обратной связи. Показано, что разработанные алгоритмы позволяют повысить степень сжатия по сравнению с кодеком DISCOVER на 4 – 16%.

Основные результаты раздела можно сформулировать следующим образом:

- разработана программная модель видеокодека, основанная на базовой модели распределенного кодирования Discover;
- решены задачи, связанные с внедрением в программную модель алгоритмов, разработанных в разделах 2 и 3;
- предложен метод сравнительной оценки алгоритмов генерации дополнительной информации декодера;
- продемонстрирован выигрыш от применения алгоритмов, разработанных в настоящей диссертационной работе.

ЗАКЛЮЧЕНИЕ

В диссертационной работе рассмотрены методы повышения эффективности сжатия видеоданных с использованием алгоритмов, основанных на принципах кодирования зависимых источников. Основное внимание уделено вопросам, связанным с генерацией дополнительной информации декодера и моделированием виртуального канала.

Основные результаты работы заключаются в следующем.

1. Разработан алгоритм межкадрового предсказания для кодеков видеоинформации, основанных на принципах кодирования зависимых источников с дополнительной информацией на декодере, использующий метод временной интерполяции, позволяющий за счет использования процедуры оценки истинного движения объектов уменьшить по сравнению с существующими алгоритмами число ошибок интерполяции и тем самым повысить степень сжатия на 10 – 30% для последовательностей с быстрым движением и на 3 – 10% на последовательностях с медленным движением.
2. Предложен метод выполнения сравнительной оценки алгоритмов межкадрового предсказания в системах кодирования зависимых источников видеоинформации с дополнительной информацией на декодере, учитывающий тот факт, что модули системы распределенного кодирования оказывают сильное влияние друг на друга. В рамках предложенного метода разработана упрощенная модель видеокодека без обратной связи, позволяющая выполнять сравнительный анализ алгоритмов межкадрового предсказания, устраняя влияние остальных модулей системы.
3. Предложена модификация базового алгоритма оценки параметров ошибок межкадрового предсказания из кодека DISCOVER, основанная на процедуре разбиения неоднородного множества ошибок на подмножества с однородными характеристиками с последующим применением базового алгоритма в каждом подмножестве независимо. Использование модифицированного алгоритма позволяет повысить на последовательностях с быстрым движением степень сжатия на 0.6 – 1% по сравнению с базовым алгоритмом.

4. Проведено исследование существующих моделей виртуального канала в системах распределенного кодирования видеоданных. Показано, что общепринятый набор допущений не в полной мере описывает особенности ошибок межкадрового предсказания. Предложен новый расширенный набор допущений и на его основе предложена новая модель виртуального канала, основанная на модели ошибок, представляющей процесс возникновения искажений с помощью скрытого Марковского случайного поля (СМСП).

СПИСОК ЛИТЕРАТУРЫ

1. *Ричардсон, Я.* Видеокодирование. H.264 и MPEG-4 - стандарты нового поколения / Я. Ричардсон. — Москва: Техносфера, 2005. — С. 368.
2. *Brites, C.* Distributed Video Coding: Bringing New Applications to Life / C. Brites, F. Pereira // Conf. on Telecommunications - ConfTele. — 2005. — April.
3. Fast Motion Estimation for Real Time H.264 Video Encoder / Ouyang Kun, Ouyang Qing, Zhou Zhengda, Li Zhitang // International Conference on MultiMedia and Information Technology. — 2008. — Dec. — Pp. 310–313.
4. *Веселов, А.И.* Обработка видеоинформации в системах сжатия, основанных на принципах кодирования зависимых источников / А.И. Веселов, М.Р. Гильмутдинов. — ГУАП, 2014. — С. 71.
5. *Веселов, А.И.* Метод генерации сторонней информации для систем распределенного кодирования видеоисточников / А.И. Веселов, М.Р. Гильмутдинов, Б.С. Филиппов // *Известия вузов. Приборостроение*. — 2013. — Т. 56, № 8. — С. 62–68.
6. *Slepian, D.* Noiseless coding of correlated information sources / D. Slepian, J.K. Wolf // *IEEE Transactions on Information Theory*. — 1973. — Vol. 19, no. 4. — Pp. 471–480.
7. *Колесник, В.Д.* Курс теории информации / В.Д. Колесник, Г.Ш. Полтырев. — М.: Наука, 1982. — С. 416.
8. *Wyner, A.D.* The rate-distortion function for source coding with side information at the decoder / A.D. Wyner, J. Ziv // *IEEE Transactions on Information Theory*. — 1976. — Vol. 22, no. 1. — Pp. 1–10.
9. *Кудряшов, Б.Д.* Теория информации: [учеб. пособие по направлению подгот. 230200 "Информ. системы"] / Б.Д. Кудряшов. Учебник для вузов. — Питер, 2009. — С. 314.

10. *Pradhan, S.S.* Duality between source coding and channel coding and its extension to the side information case / S.S. Pradhan, J. Chou, K. Ramchandran // *IEEE Transactions on Information Theory*. — 2003. — Vol. 49, no. 5. — Pp. 1181–1203.
11. *Pradhan, S.S.* Distributed source coding using syndromes (DISCUS): design and construction / S.S. Pradhan, K. Ramchandran // *IEEE Transactions on Information Theory*. — 2003. — Mar. — Vol. 49, no. 3. — Pp. 626–643.
12. *Aaron, A.* Wyner-Ziv coding of motion video / A. Aaron, Rui Zhang, B. Girod // *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*. — Vol. 1. — 2002. — Nov. — Pp. 240–244.
13. *Koopman, Philip.* Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks. — 2004.
14. *Aaron, Anne.* Compression with side information using turbo codes / Anne Aaron, Bernd Girod // *Data Compression Conference Proceedings / IEEE*. — 2002. — Pp. 252–261.
15. Transform-domain Wyner-Ziv codec for video / Anne Aaron, Shantanu D Rane, Eric Setton, Bernd Girod // *Electronic Imaging 2004 / International Society for Optics and Photonics*. — 2004. — Pp. 520–528.
16. *Puri, Rohit.* PRISM: an uplink-friendly multimedia coding paradigm / Rohit Puri, Kannan Ramchandran // *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP'03) / IEEE*. — Vol. 4. — 2003. — Pp. 856–859.
17. *Puri, Rohit.* PRISM: A video coding paradigm with motion estimation at the decoder / Rohit Puri, Abhik Majumdar, Kannan Ramchandran // *IEEE Transactions on Image Processing*. — 2007. — Vol. 16, no. 10. — Pp. 2436–2448.
18. The DISCOVER codec: architecture, techniques and evaluation / Xavi Artigas, João Ascenso, Marco Dalai et al. // *Picture Coding Symposium / Lisbon, Portugal*. — Vol. 17. — 2007. — Pp. 1103–1120.
19. *Dragotti, Pier Luigi.* Distributed source coding : theory, algorithms, and applications / Pier Luigi Dragotti, Michael Gastpar. — Amsterdam, Boston: Academic Press, 2009.

20. Overview of the H.264/AVC video coding standard / T. Wiegand, G.J. Sullivan, G. Bjontegaard, A. Luthra // *IEEE Transactions on Circuits and Systems for Video Technology*. — 2003. — Vol. 13, no. 7. — Pp. 560–576.
21. *Khayam, Syed Ali*. The Discrete Cosine Transform (DCT): Theory and Application. Department of electrical and computing engineering. — 2003.
22. *Kubasov, D*. A Hybrid Encoder/Decoder Rate Control for Wyner-Ziv Video Coding with a Feedback Channel / D. Kubasov, K. Lajnef, C. Guillemot // *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*. — 2007. — Pp. 251–254.
23. *Varodayan, D*. Rate-Adaptive Distributed Source Coding using Low-Density Parity-Check Codes / D. Varodayan, A. Aaron, B. Girod // *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*. — 2005. — Pp. 1203–1207.
24. *Pereira, F*. — Application Scenarios and Functionalities for DVC, 2007.
25. *Veselov, A*. Iterative hierarchical true motion estimation for temporal frame interpolation / A. Veselov, M. Gilmudtinov // *IEEE 16th International Workshop on Multimedia Signal Processing*. — 2014. — Sept. — Pp. 1–6.
26. *Веселов, А.И.* Алгоритм временной интерполяции кадров, основанный на процедуре итеративной оценки движения / А.И. Веселов, М.Р. Гильмутдинов // *Информационно-управляющие системы*. — 2014. — Т. 71, № 4. — С. 25–33.
27. *Huang, Xin*. Improved side information generation for Distributed Video Coding / Xin Huang, S. Forchhammer // *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*. — 2008. — Pp. 223–228.
28. *Tome, Antonio*. Low Delay Distributed Video Coding: Ph.D. thesis / University of Lisbon. — 2009.
29. *Chen, Yen-Kuang*. True Motion Estimation - Theory, Application, and Implementation. — 1998.
30. True-motion estimation with 3-D recursive search block matching / G. de Haan, P.W.A.C. Biezen, H. Huijgen, O.A. Ojo // *IEEE Transactions on Circuits and Sys-*

- tems for Video Technology*. — 1993. — Oct. — Vol. 3, no. 5. — Pp. 368–379, 388.
31. *Bartels, Chris*. Occlusion classifiers for picture rate conversion. — 2009.
 32. *Choi, Byung-Tae*. New frame rate up-conversion using bi-directional motion estimation / Byung-Tae Choi, Sung-Hee Lee, Sung-Jea Ko // *IEEE Transactions on Consumer Electronics*. — 2000. — Aug. — Vol. 46, no. 3. — Pp. 603–609.
 33. *Ascenso, J*. Content Adaptive Wyner-ZIV Video Coding Driven by Motion Activity / J. Ascenso, C. Brites, F. Pereira // 2006 IEEE International Conference on Image Processing. — 2006. — Pp. 605–608.
 34. *Ascenso, Joao*. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding / Joao Ascenso, Catarina Brites, Fernando Pereira // 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services / Citeseer. — 2005. — Pp. 1–6.
 35. Motion estimation optimization / S.A. Rajala, I.M. Abdelqadar, G.L. Bilbro, W.E. Snyder // IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-92). — Vol. 3. — 1992. — Mar. — Pp. 253–256 vol.3.
 36. *Boykov, Y*. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision / Y. Boykov, V. Kolmogorov // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 2004. — Sept. — Vol. 26, no. 9. — Pp. 1124–1137.
 37. Graph-Cut versus Belief-Propagation Stereo on Real-World Images / Sandino Morales, Joachim Penc, Tobi Vaudrey, Reinhard Klette // Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications / Ed. by Eduardo Bayro-Corrochano, Jan-Olof Eklundh. — Springer Berlin Heidelberg, 2009. — Vol. 5856 of *Lecture Notes in Computer Science*. — Pp. 732–740.
 38. *Sohn, Young Wook*. Block-based Motion Vector Smoothing for Periodic Pattern Region / Young Wook Sohn, Moon Gi Kang // Proceedings of the 4th International Conference on Image Analysis and Recognition. — ICIAR'07. — Berlin, Heidelberg: Springer-Verlag, 2007. — Pp. 491–500.

39. *Liu, Lurng-Kuo*. A block-based gradient descent search algorithm for block motion estimation in video coding / Lurng-Kuo Liu, E. Feig // *IEEE Transactions on Circuits and Systems for Video Technology*. — 1996. — Aug. — Vol. 6, no. 4. — Pp. 419–422.
40. Optimization of Hierarchical 3DRS Motion Estimators for Picture Rate Conversion / A. Heinrich, C. Bartels, R.J. van der Vleuten et al. // *Selected Topics in Signal Processing, IEEE Journal of*. — 2011. — April. — Vol. 5, no. 2. — Pp. 262–274.
41. *Mémin, E*. A Multigrid Approach for Hierarchical Motion Estimation / E. Mémin, P. Pérez // *Proceedings of the Sixth International Conference on Computer Vision*. — ICCV '98. — Washington, DC, USA: IEEE Computer Society, 1998. — Pp. 933–.
42. *Huang, Ai-Mei*. A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation / Ai-Mei Huang, T.Q. Nguyen // *IEEE Transactions on Image Processing*. — 2008. — May. — Vol. 17, no. 5. — Pp. 694–708.
43. *Huang, Ai-Mei*. Motion vector processing using the color information / Ai-Mei Huang, Truong Nguyen // *16th IEEE International Conference on Image Processing (ICIP)*. — 2009. — Nov. — Pp. 1605–1608.
44. *Huang, Ai Mei*. Correlation-Based Motion Vector Processing With Adaptive Interpolation Scheme for Motion-Compensated Frame Interpolation / Ai Mei Huang, Truong Nguyen // *IEEE Transactions on Image Processing*. — 2009. — April. — Vol. 18, no. 4. — Pp. 740–752.
45. *Huang, Ai-Mei*. Motion vector processing using the color information / Ai-Mei Huang, Truong Nguyen // *2009 16th IEEE International Conference on Image Processing (ICIP)*. — 2009. — Nov. — Pp. 1605–1608.
46. *Orchard, M.T*. Overlapped block motion compensation: an estimation-theoretic approach / M.T. Orchard, G.J. Sullivan // *IEEE Transactions on Image Processing*. — 1994. — Sep. — Vol. 3, no. 5. — Pp. 693–699.
47. *Veselov, A*. A Generalized Correlation Noise Model for Pixel Domain Wyner-Ziv Video Coding / A. Veselov, M. Gilmutdinov // *6th International Congress on Ultra*

- Modern Telecommunications and Control Systems and Workshops. — 2014. — Oct.
48. Non-stationary Correlation Noise Modeling for Transform Domain Wyner-Ziv Video Coding / A. Veselov, B. Filippov, V. Yastrebov, M. Gilmutdinov // *Smart Innovation, Systems and Technologies*. — Springer, 2015. — Vol. 40. — Pp. 179–189.
 49. *Веселов, А.И.* Анализ моделей шума в виртуальном канале в системах распределенного кодирования видеоданных / А.И. Веселов // *Научная сессия ГУАП*. — 2011. — С. 48–51.
 50. *Ishwar, P.* Towards a theory for video coding using distributed compression principles / P. Ishwar, V.M. Prabhakaran, K. Ramchandran // *International Conference on Image Processing*. — Vol. 2. — 2003. — Sept. — Pp. II–687–90 vol.3.
 51. *Borchert, S.* Distributed Video Coding (DVC): Motion Estimation and DCT Quantization in Low Complexity Video Compression / S. Borchert. — TU Delft, Mediamatica, 2010.
 52. *Brites, C.* Correlation Noise Modeling for Efficient Pixel and Transform Domain Wyner-Ziv Video Coding / C. Brites, F. Pereira // *IEEE Transactions on Circuits and Systems for Video Technology*. — 2008. — Sept. — Vol. 18, no. 9. — Pp. 1177–1190.
 53. *Bishop, Christopher M.* Pattern Recognition and Machine Learning (Information Science and Statistics) / Christopher M. Bishop. — Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
 54. *Wu, C. F. J.* On the convergence properties of the EM algorithm / C. F. J. Wu // *The Annals of Statistics*. — 1983. — Vol. 11, no. 1. — Pp. 95–103.
 55. *Li, Stan Z.* Markov Random Field Modeling in Image Analysis / Stan Z. Li. — 3rd edition. — Springer Publishing Company, Incorporated, 2009.
 56. *Koller, Daphne.* Probabilistic Graphical Models: Principles and Techniques / Daphne Koller, Nir Friedman. — MIT Press, 2010. — P. 1270.

57. *Chien, Wei-Jung*. BitpLANe SelecTive distributed video coding / Wei-Jung Chien, L.J. Karam // 42nd Asilomar Conference on Signals, Systems and Computers. — 2008. — Pp. 2238–2242.
58. Power Consumption Analysis for Distributed Video Sensors in Machine-to-Machine Networks / Shao-Yi Chien, Teng-Yuan Cheng, Shun-Hsing Ou et al. // *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*. — 2013. — March. — Vol. 3, no. 1. — Pp. 55–64.
59. Wyner-Ziv Coding of Video with Unsupervised Motion Vector Learning / David Varodayan, David Chen, Markus Flierl, Bernd Girod // *Image Commun.* — 2008. — Vol. 23, no. 5. — Pp. 369–378.
60. *Bjontegaard, G*. Calculation of average PSNR differences between RD-curves (VCEG-M33) / G. Bjontegaard // VCEG Meeting (ITU-T SG16 Q.6). — 2001.
61. Studying the Feedback Channel in Transform Domain Wyner-Ziv Video Coding / J. Pedro, C. Brites, J. Ascenso, F. Pereira // Conf. on Telecommunications - ConfTele. — 2007. — May.
62. H.264/AVC JM Reference Software. — 2008.
63. *Kubasov, Denis*. Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side / Denis Kubasov, Jayanth Nayak, Christine Guillemot // Information”, Int. Workshop on Multimedia Signal Processing. — 2007.
64. *Gilmutdinov, Marat*. Analysis of side information generation impact on distributed video coding performance / Marat Gilmutdinov, Anton Veselov, Boris Filippov // Problems of Redundancy in Information and Control Systems (RED), 2012 XIII International Symposium on / IEEE. — 2012. — Pp. 26–30.
65. *Huang, Xin*. Improved virtual channel noise model for transform domain Wyner-Ziv video coding / Xin Huang, S. Forchhammer // IEEE International Conference on Acoustics, Speech and Signal Processing. — 2009. — April. — Pp. 921–924.
66. *Петюшко, А.А.* О марковских случайных полях и их связи с цепями Маркова / А.А. Петюшко // *Интеллектуальные системы*. — 2010. — Т. 14, № 1-4. — С. 225–236.

СПИСОК РИСУНКОВ

1.1	Структура системы обработки информации с использованием независимого кодирования зависимых источников	13
1.2	Область допустимых скоростей при кодировании зависимых источников	14
1.3	Структура системы передачи информации при кодировании зависимых источников с дополнительной информацией на декодере	15
1.4	Альтернативная интерпретация кодирования с дополнительной информацией на декодере	16
1.5	Система передачи информации с независимой обработкой	18
1.6	Система передачи информации с совместной обработкой	19
1.7	Система передачи с обработкой информации по схеме Вайнера-Зива	20
1.8	Возможные направления смещения точки между соседними кадрами	23
1.9	Структурная схема обобщенной системы передачи информации от зависимых источников с общим декодером	24
1.10	Классификация методов распределенного кодирования видеоданных	29
1.11	Типовая схема кодека, основанного на концепции Стэнфорд	30
1.12	Типовая схема кодека, основанного на концепции PRISM	33
1.13	Структурная схема модели кодека DISCOVER	36
2.1	Обобщенная схема типового алгоритма временной интерполяции .	43
2.2	Схема алгоритма межкадрового предсказания в модели DISCOVER	46
2.3	Схема разработанного алгоритма оценки движения	54
3.1	Пример ошибок межкадрового предсказания для кадра из последовательности SOCCER (значение 128 соответствует нулевой ошибке)	68
3.2	Пример оцененной на стороне декодера ошибки межкадрового предсказания в пространстве пикселей	71
3.3	Пример аппроксимации ошибок межкадрового предсказания в спектральной области (коэффициент DC).	72

3.4	Функции плотности вероятности для различных значений параметра масштаба.	74
3.5	Пример вероятностной графической модели для скрытого Марковского случайного поля.	85
4.1	Сравнение разработанного программного комплекса и кодека DISCOVER по кривым «скорость-искажение»	106
4.2	Графики PSNR для интерполированных кадров	111
4.3	Структурная схема видеокодека для проведения сравнительной оценки алгоритмов генерации дополнительной информации	112
4.4	Результаты сравнения алгоритмов генерации дополнительной информации в рамках распределенного кодека	113
4.5	Кривые «скорость-искажение», полученные для сравнения алгоритмов моделирования виртуального канала в рамках экспериментов первого типа	117
4.6	Кривые «скорость-искажение», полученные для сравнения алгоритмов моделирования виртуального канала в рамках экспериментов второго типа	117
4.7	Сравнение с существующими алгоритмами сжатия	120

СПИСОК ТАБЛИЦ

1.1	Таблица стандартной расстановки	21
4.1	Параметры квантования (QP) ключевых кадров для тестовых последовательностей	98
4.2	Среднее значение критерия PSNR для промежуточных кадров . . .	109
4.3	Зависимость выигрыша от сложности движения	110
4.4	Значение критерия BD-RATE для экспериментов первого типа . . .	118
4.5	Значение критерия BD-RATE для экспериментов первого типа . . .	118
4.6	Результаты сравнения реализованного программного комплекса и кодека DISCOVER по критерию BD-RATE	119

ПРИЛОЖЕНИЕ А

СЛУЧАЙНЫЕ МАРКОВСКИЕ ПОЛЯ

Приведем описание основных понятий Марковских случайных полей (МСП) [56], [55], [66]. Данный вероятностный аппарат широко используется в системах компьютерного зрения. МСП позволяют описывать вероятностные ансамбли с большим количеством зависимых случайных величин. При этом для МСП существует большое число алгоритмов поиска оптимальных конфигураций, обладающих доказанными свойствами сходимости. С точки зрения обработки изображений, МСП предоставляют удобный математический аппарат для моделирования сложных взаимодействий внутри изображения:

- изображения разбиваются на множество узлов, которые соответствуют отдельным пикселям или группам пикселей;
- с каждым узлом ассоциируется скрытая переменная, которая позволяет в некотором смысле обосновать значение пикселя;
- совместная вероятностная модель строится над узлами и скрытыми переменными;
- статистические зависимости между скрытыми переменными выражаются за счет их группировки, выражающейся в наличии дуг между парами вершин графа вероятностной модели МСП.

Понятие МСП удобно формулировать в терминах задачи раскрашивания (или помечивания) элементов некоторого множества. Пусть задано множество из m элементов. Далее будем называть эти элементы узлами. Обозначим множество индексов узлов как

$$\mathcal{S} = \{1, 2, \dots, m\}.$$

Узел, как правило, представляет собой точку или регион в Евклидовом пространстве. При обработке изображений в качестве узлов принято использовать пространственные координаты групп пикселей. Множество узлов может обладать свойством регулярности, например узлы, расположенные на двумерной решетке можно рассматривать как пространственно регулярное множество. Прямоугольную решетку для пикселей двумерного изображения, состоящего из n строчек и

n столбцов можно ввести как

$$\mathcal{S} = \{(i, j) | 1 \leq i, j \leq n\}.$$

Множество узлов, не обладающее регулярной структурой в пространстве, будем называть нерегулярным. Далее будем рассматривать только регулярные множества узлов.

Меткой узла будем называть случайное событие, которое может происходить в узле. Обозначим множество всех возможных меток для узла с индексом i как \mathcal{L}_i .

Задача помечивания узлов заключается в назначении каждому узлу с индексом $i \in \mathcal{S}$ некоторой метки из \mathcal{L}_i . Обозначим множество меток, назначенных всем узлам, как

$$f = \{f_1, f_2, \dots, f_m\}.$$

где $f_i \in \mathcal{L}_i$, $f \in \prod_{i=1}^m \mathcal{L}_i$ (здесь под знаком произведения понимается декартово произведение множеств). В терминах случайных полей множество f принято называть конфигурацией случайного поля.

Пусть над множеством \mathcal{S} задано отношение соседства (система соседства). Множество соседей определяется как

$$\mathcal{N} = \{\mathcal{N}_i | \forall i \in \mathcal{S}\}.$$

где \mathcal{N}_i – множество узлов, находящихся в отношении соседства с узлом i (шаблон соседства). Отношение соседства обладает следующими свойствами:

1. Анти-рефлексивность: $i \notin \mathcal{N}_i$;
2. Симметричность: $i \in \mathcal{N}_{i'} \leftrightarrow i' \in \mathcal{N}_i$

Для множества индексов \mathcal{S} заданного над регулярным множеством узлов, множество соседей узла i можно определить как множество узлов, находящихся на расстоянии не больше r от i :

$$\mathcal{N}_i = \{i' \in \mathcal{S} | d(i, i') \leq r, i' \neq i\}.$$

где через $d(i, i')$ обозначено Евклидово расстояние между узлами с индексами i и i' .

В том случае, если над элементами множества \mathcal{S} задано отношение порядка, множество соседей можно определить в явном виде. Например, если $\mathcal{S} = \{1, 2, \dots, m\}$ содержит индексы пикселей одномерного изображения, то множество соседей для некоторого внутреннего узла i при $r = 1$ можно определить как $\mathcal{N}_i = \{i - 1, i + 1\}$. Для пикселей находящихся на границе изображения, множества соседей будут включать по одному элементу: $\mathcal{N}_1 = \{2\}$ и $\mathcal{N}_m = \{m - 1\}$.

Пара $(\mathcal{S}, \mathcal{N}) \triangleq G$ задает граф: \mathcal{S} содержит множество вершин графа, \mathcal{N} определяет дуги.

Введем понятие случайного Марковского поля. Обозначим через $F = (F_1, F_2, \dots, F_m)$ многомерную случайную величину, причем F_i принимает значение $f_i \in \mathcal{L}$. Величина F называется случайным полем. Следует отметить, что приведенные далее рассуждения справедливы и в том случае, когда поле задается как множество случайных величин, а не как одна многомерная случайная величина. Обозначим через $F_i = f_i$ событие, заключающееся в том, что случайная величина F_i принимает значение f_i , а через $(F_1 = f_1, \dots, F_m = f_m)$ обозначим совместное событие. Совместное событие будем обозначать как $F = f$, где $f = (f_1, \dots, f_m)$ – конфигурация F , соответствующая реализации поля.

Для дискретного множества \mathcal{L} вероятность того, что F_i примет значение f_i обозначим как $p(f_i)$, а вероятность совместного события – $p(f)$.

Определение А.1. Случайное поле F называется Марковским случайным полем над \mathcal{S} по отношению к \mathcal{N} если и только если выполняются два условия:

- положительность: $p(f) > 0, \forall f \in \mathcal{L}^m$;
- марковость: $p(f_i | f_{\mathcal{S} \setminus \{i\}}) = p(f_i | f_{\mathcal{N}_i})$;

где $\mathcal{S} \setminus \{i\}$ – разность двух множеств (множество узлов из \mathcal{S} , кроме узла i), $f_{\mathcal{S} \setminus \{i\}}$ – множество меток в узлах из $\mathcal{S} \setminus \{i\}$, и

$$f_{\mathcal{N}_i} = \{f'_i | i' \in \mathcal{N}_i\}. \quad (A.1)$$

Первое условие заключается в том, что не существует конфигурации поля с нулевой вероятностью. Второе свойство определяет локальные характеристики поля: только метки, соответствующие узлам, находящимся в отношении соседства, оказывают влияние (зависят) друг на друга.

Марковские случайные поля можно рассматривать как обобщение двухсторонних цепей Маркова изменив индексы времени на координаты пространственного положения.

Приведем один из ключевых результатов, известный как теорема Хаммерсли-Клиффорда, позволяющий рассчитывать вероятности различных конфигураций МСП.

Введем понятие клики. Кликой c для графа G называется полносвязный подграф графа G . Клика определяется как подмножество вершин множества \mathcal{S} . Она может состоять единственного узла $\{i\}$, пары соседних узлов $\{i, i'\}$, тройки соседних узлов $\{i, i', i''\}$ и т. д. Обозначим множество клик, состоящих из одного узла как

$$\mathcal{C}_1 = \{i | i \in \mathcal{S}\},$$

из двух узлов как

$$\mathcal{C}_2 = \{\{i, i'\} | i' \in N_i, i \in \mathcal{S}\}$$

и т. д. Тогда множество всех клик графа G можно определить как

$$\mathcal{C} = \bigcup_i \mathcal{C}_i.$$

Будем называть потенциальной функцией $V_c(c)$, $c \in \mathcal{C}$, любую функцию, отображающую значения случайных величин из клики c (конфигурацию клики) на вещественную ось.

Определение А.2. *Дискретное распределение называется распределением Гиббса, если*

$$P(f) = \frac{1}{Z} e^{-\sum_{c \in \mathcal{C}} V_c(f)},$$

где Z – нормировочный коэффициент, обеспечивающий, что $P(f)$ – распределение вероятностей, f – значения, которые приняли случайные величины, поставленные в соответствие с вершинами из клики c .

Теорема А.1. *Поле F является Марковским случайным полем тогда и только тогда, когда $P(f)$ является распределением Гиббса.*

Таким образом, чтобы задать распределение вероятностей над МСП, достаточно определить потенциальные функции на всех кликах графа.

ПРИЛОЖЕНИЕ Б**ПАРАМЕТРЫ КОДЕКА КЛЮЧЕВЫХ КАДРОВ**

Параметры кодека ключевых кадров (JM 9.5), общие для всех последовательностей:

InputHeaderLength = 0

FrameRate = 30.0

TraceFile = "trace_enc.txt"

OutputFile = "test.264"

ProfileIDC = 77

LevelIDC = 40

IntraPeriod = 1

EnableOpenGOP = 0

IDRIntraEnable = 0

QPPSlice = 28

FrameSkip = 0

ChromaQPOffset = 0

UseHadamard = 0

DisableSubpelME = 0

SearchRange = 16

NumberReferenceFrames = 5

PList0References = 0

Log2MaxFrameNum = 0

GenerateMultiplePPS = 0

ResendPPS = 0

MbLineIntraUpdate = 0

RandomIntraMBRefresh = 0

InterSearch16x16 = 1

InterSearch16x8 = 1

InterSearch8x16 = 1

InterSearch8x8 = 1

InterSearch8x4 = 1

InterSearch4x8 = 1

InterSearch4x4 = 1
IntraDisableInterOnly = 0
Intra4x4ParDisable = 0
Intra4x4DiagDisable = 0
Intra4x4DirDisable = 0
Intra16x16ParDisable = 0
Intra16x16PlaneDisable = 0
ChromaIntraDisable = 0
UseFME = 0
NumberBFrames = 0
QPBSlice = 30
BRefPicQPOffset = 0
DirectModeType = 1
DirectInferenceFlag = 1
BList0References = 0
BList1References = 1
BReferencePictures = 0
PyramidCoding = 0
ExplicitPyramidFormat = "b2r28b0e30b1e30b3e30b4e30"
PyramidRefReorder = 1
PocMemoryManagement = 1
BiPredMotionEstimation = 0
BiPredMERefinements = 3
BiPredMESearchRange = 16
BiPredMESubPel = 1
SPPicturePeriodicity = 0
QPSPSlice = 28
QPSP2Slice = 27
SymbolMode = 1
OutFileMode = 0
PartitionMode = 0
ContextInitMethod = 1
FixedModelNumber = 0
PicInterlace = 0

MbInterlace = 0
IntraBottom = 0
WeightedPrediction = 0
WeightedBiprediction = 0
UseWeightedReferenceME = 0
RDPictureDecision = 0
RDPictureIntra = 0
RDPSliceWeightOnly = 1
RDBSliceWeightOnly = 0
LoopFilterParametersFlag = 0
LoopFilterDisable = 0
LoopFilterAlphaC0Offset = 0
LoopFilterBetaOffset = 0
SliceMode = 0
SliceArgument = 50
num_slice_groups_minus1 = 0
slice_group_map_type = 0
slice_group_change_rate_minus1 = 85
SliceGroupConfigFileName = "sg0conf.cfg"
UseRedundantSlice = 0
RestrictSearchRange = 2
RDOptimization = 1
DisableThresholding = 0
DisableBSkipRDO = 0
SkipIntraInInterSlices = 0
UseExplicitLambdaParams = 0
LambdaWeightIslice = 0.65
LambdaWeightPslice = 0.68
LambdaWeightBslice = 2.00
LambdaWeightRefBslice = 1.50
LambdaWeightSPslice = 1.50
LambdaWeightSIslice = 0.65
LossRateA = 10
LossRateB = 0

LossRateC = 0
NumberOfDecoders = 30
RestrictRefFrames = 0
UseConstrainedIntraPred = 0
LastFrameNumber = 0
ChangeQPI = 16
ChangeQPP = 16
ChangeQPB = 18
ChangeQPBSRefOffset = 2
ChangeQPStart = 0
NumberOfLeakyBuckets = 8
LeakyBucketRateFile = "leakybucketrate.cfg"
LeakyBucketParamFile = "leakybucketparam.cfg"
NumberFramesInEnhancementLayerSubSequence = 0
NumberOfFrameInSecondIGOP = 0
SparePictureOption = 0
SparePictureDetectionThr = 6
SparePicturePercentageThr = 92
PicOrderCntType = 0
RateControlEnable = 0
Bitrate = 45020
InitialQP = 24
EarlySkipEnable = 0
SelectiveIntraEnable = 0
YUVFormat = 1
RGBInput = 0
BitDepthLuma = 8
BitDepthChroma = 8
CbQPOffset = 0
CrQPOffset = 0
Transform8x8Mode = 0
ResidueTransformFlag = 0
ReportFrameStats = 0
DisplayEncParams = 0

Verbose = 1
QmatrixFile = "q_matrix.cfg"
ScalingMatrixPresentFlag = 0
ScalingListPresentFlag0 = 0
ScalingListPresentFlag1 = 0
ScalingListPresentFlag2 = 0
ScalingListPresentFlag3 = 0
ScalingListPresentFlag4 = 0
ScalingListPresentFlag5 = 0
ScalingListPresentFlag6 = 0
ScalingListPresentFlag7 = 0
OffsetMatrixPresentFlag = 0
QOffsetMatrixFile = "q_offset.cfg"
AdaptiveRounding = 0
AdaptRndPeriod = 1
AdaptRndChroma = 0
AdaptRndWFactorIRef = 4
AdaptRndWFactorPRef = 4
AdaptRndWFactorBRef = 4
AdaptRndWFactorINRef = 4
AdaptRndWFactorPNRef = 4
AdaptRndWFactorBNRef = 4
QPPrimeYZeroTransformBypassFlag = 0