

УДК 004.932.2

АЛГОРИТМ ОПТИМИЗАЦИИ ВЕКТОРНОГО ПОЛЯ В ЗАДАЧЕ ВРЕМЕННОЙ ИНТЕРПОЛЯЦИИ КАДРОВ

А. И. Веселов^а, ассистентМ. Р. Гильмутдинов^а, канд. техн. наук, доцент^аСанкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, РФ

Цель: разработка алгоритма оценки истинного движения для систем интерполяции кадров во времени (временной интерполяции кадров). **Результаты:** предложен алгоритм оценки движения по двум смежным кадрам видеопоследовательности. Алгоритм учитывает модель истинного движения объектов в видеопоследовательностях, основанную на допущении о гладкости векторного поля. При этом в предложенном алгоритме дополнительно обрабатываются ситуации, когда это допущение не верно, в частности, на регионах с границами объектов. В основе алгоритма лежит итеративная процедура решения оптимизационной задачи, введенной в рамках модели. Процедура базируется на постепенном уточнении движения от больших объектов к малым, расчете надежности векторов движения и сглаживании векторного поля с учетом надежных векторов. Результаты работы нового алгоритма представлены для задачи преобразования кадровой скорости на стандартных тестовых множествах. Показано, что простой алгоритм интерполяции, использующий только предложенный метод оценки движения, сравним по эффективности с алгоритмами преобразования кадровой скорости, использующими более сложные эвристики. **Практическая значимость:** предложенный алгоритм позволяет повысить визуальное качество интерполяции промежуточных кадров (данное приложение является одной из ключевых компонент современных телевизионных приемников), что является важной и актуальной задачей в современных системах обработки видеоданных.

Ключевые слова — истинная оценка движения, временная интерполяция, оптимизационная задача, векторное поле, модель истинного движения, преобразование кадровой скорости.

Введение

Процедура временной интерполяции кадров является одним из ключевых блоков многих современных систем обработки визуальных данных, в частности, систем преобразования кадровой скорости в телевизионных приемниках [1], распределенного кодирования видеоданных [2], масштабируемого кодирования видеисточника [3] и т. д. Одним из основных методов, применяемых при временной интерполяции, является оценка истинного движения объектов в видеопоследовательности [4]. С точки зрения теории модель истинного движения хорошо формализована, и существуют алгоритмы, явно учитывающие эту модель при поиске движения. Однако на практике многие допущения, принятые в модели истинного движения, не выполняются, и для временной интерполяции разрабатываются эвристические алгоритмы, часто имеющие высокую сложность (сегментация, выделение текстур и т. п.) [5].

В настоящей статье рассмотрена модель истинного движения объектов в видеопоследовательностях. Предложен алгоритм оценки движения для задачи временной интерполяции кадров. Алгоритм хорошо согласуется с моделью истинного движения и может рассматриваться как частный способ решения оптимизационной задачи, определенной в рамках этой модели. Принцип метода базируется на подходах, основанных на предсказании векторов движения, что способствует точной оценке движения на краях объек-

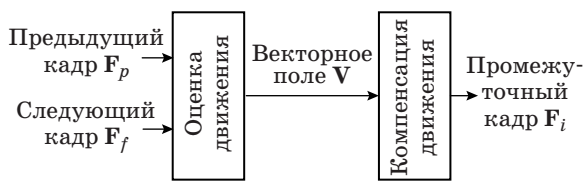
тов при сохранении гладкости векторного поля в целом. Эффективность предложенного алгоритма исследуется с использованием программного комплекса временной интерполяции кадров. В статье приводятся результаты сравнения данного комплекса с известными реализациями алгоритмов временной интерполяции.

Теоретические основы

Одной из основных задач временной интерполяции кадров является задача точного отслеживания перемещений объектов на оригинальных кадрах видеопоследовательности, называемая задачей поиска истинного движения (true motion estimation). Информация об истинном движении далее используется для корректной интерполяции объектов на промежуточных кадрах. В связи с этим большинство современных алгоритмов временной интерполяции кадров основано на процедурах оценки и компенсации движения (Motion Compensated Frame Interpolation). Обобщенная схема алгоритма интерполяции (рис. 1) включает два основных блока:

— блок оценки движения (Motion Estimation) — применяется для оценки изменений между соседними кадрами, которая оперирует с терминами «вектор движения» и «векторное поле» (множество всех векторов движения);

— блок компенсации движения — применяется для интерполяции нового кадра с использованием информации от предыдущего блока.



■ **Рис. 1.** Обобщенная схема алгоритма интерполяции кадров

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

- 1) поиск истинного движения;
- 2) обработка регионов на интерполированном кадре, не имеющих вектора движения или имеющих более одного вектора движения [6].

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, принимают ряд допущений о векторном поле, отражающих специфику перемещений объектов на кадрах. Одним из наиболее распространенных является допущение о *гладкости* векторного поля, заключающееся в том, что векторы движения, соответствующие пространственно близким регионам, должны быть коррелированы. Как правило, для расчета гладкого векторного поля используются схемы, основанные на предсказании движения. В этих схемах вектор для нового региона оценивается 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. Поиск истинного движения для таких регионов проводится с помощью предсказания с учетом векторов, полученных по обе стороны границы объекта. Эти предсказания затем используются для инициализации векторов, соответствующих региону, содержащему границу. Подобный подход применяется в одном из первых алгоритмов поиска истинного движения — трехмерном рекурсивном поиске (3D Recursive Search — 3DRS) [7].

Вторая задача временной интерполяции связана с назначением векторов движения областям на интерполированном кадре. Существует два основных подхода к решению этой задачи: однонаправленная [7] и двунаправленная (билатеральная) [8] оценка/компенсация движения.

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. В нем основным допущением является допущение о *равномерности* и *прямолинейности* движения объектов, что позволяет разбивать на блоки не один из оригинальных кадров, а интерполируемый кадр. Для каждой фиксированной позиции блока на интерполируемом кадре выполняется поиск похожих блоков на оригинальных кадрах, при котором векторы смещения коллинеарны, разнонаправлены и откладываются относительно блока в интерполируемом кадре. Этот подход не решает проблемы, возникающие из-за появления/исчезновения/наложения объектов на кадрах, но, в отличие от однонаправленного поиска, представляет завершённую процедуру обработки регионов возле этих объектов.

Разработанный алгоритм оценки движения использует пару соседних кадров оригинальной видеопоследовательности F_p и F_f для оценки векторного поля V , описывающего смещение объектов между этими кадрами. Кадры F_p и F_f будем называть *базовыми* кадрами. Векторное поле V используется для интерполяции 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 описанная далее методика может быть легко расширена на изображения с произвольным числом компонент.

Модель векторного поля для задачи оценки истинного движения может быть введена как результат оптимизации следующей функции при фиксированных кадрах F_p и F_f [9]:

$$V^* = \arg \min_{V \in \Omega} E(F_f, F_p, V), \quad (1)$$

где Ω — пространство всех возможных векторных полей \mathbf{V} и

$$E(\mathbf{F}_f, \mathbf{F}_p, \mathbf{V}) = E_1(\mathbf{F}_f, \mathbf{F}_p, \mathbf{V}) + \alpha E_2(\mathbf{V}). \quad (2)$$

Слагаемое E_1 соответствует энергии разностного кадра между кадрами \mathbf{F}_p и \mathbf{F}_f с использованием векторного поля \mathbf{V} ; слагаемое E_2 отражает гладкость 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. 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.

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

$$E(\mathbf{F}_f, \mathbf{F}_p, \mathbf{V}^*) \leq E(\mathbf{F}_f, \mathbf{F}_p, \mathbf{V}). \quad (3)$$

В том случае, если (3) выполняется для всех $\mathbf{V} \in \Omega$, говорят, что векторное поле \mathbf{V} доставляет *глобальный* минимум E .

Следует отметить, что существуют алгоритмы, которые позволяют искать минимумы функции (2) в явном виде [10]. Так как функция (2) зависит от большого числа параметров и не является унимодальной, найденные минимумы являются, как правило, локальными. Однако следует отметить, что с помощью формулы (2) 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 обычно не является гладким. Поэтому

найденное для данной модели решение не гарантирует оптимального визуального качества интерполяции. В связи с этим более хорошие результаты с точки зрения визуального качества достигаются с использованием подоптимальных алгоритмов, рассматривающих оптимизацию (1) 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), т. е. выдают результат, согласованный с моделью истинного движения. Разработанный алгоритм оценки движения принадлежит именно к такому классу алгоритмов.

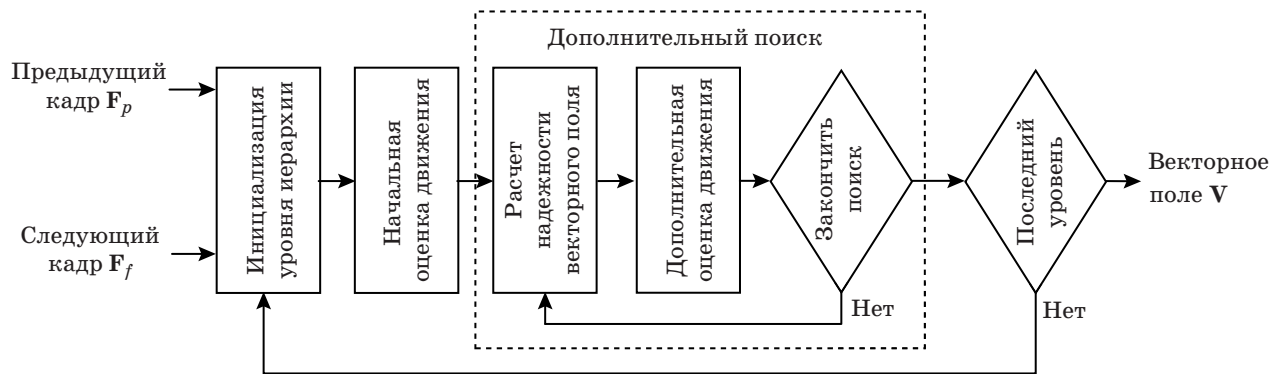
Процедура поиска оптимального вектора движения, лежащая в основе всех методов оценки движения, также может быть реализована с использованием либо полного перебора всех возможных векторов (оптимальный поиск), либо некоторого подоптимального алгоритма. Полный перебор оптимален в том смысле, что он гарантирует минимум ошибки E_1 в (2). В ряде работ [11] указывается, что полный перебор плохо согласуется с поиском истинного движения, особенно 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 подоптимальных методов, например, градиентного спуска.

В основе разработанного алгоритма оценки движения (рис. 2) лежит многоуровневая иерархическая билатеральная процедура блочной оценки движения с дополнительным итеративным поиском. На каждом уровне иерархии для фиксированного размера блока выполняются три 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);
- 3) итеративный дополнительный поиск, повышающий гладкость векторного поля [уменьшение E_2 в (2)].

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 $\tilde{\mathbf{V}}^*$, доставляющее локальный минимум (2). Полученные векторы движения используются в качестве начального смещения на следующей стадии.

Иерархическая оценка реализована с использованием многосеточного подхода — размер блока уменьшается с номером стадии [12]. Идея такого подхода заключается в том, что сначала



■ Рис. 2. Схема предложенного алгоритма оценки движения

выполняется поиск движения для больших областей на базовых кадрах, затем полученные векторы уточняются для все меньших и меньших регионов. Такой метод позволяет осуществлять поиск локального смещения объектов с учетом глобального движения в кадре.

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 процесс, основанный на *предсказании* с учетом надежности векторов. Вектор считается надежным, если соответствующая ему билатеральная ошибка сопоставления мала (маленькое значение слагаемого E_1) и его значение коррелировано со значениями соседей (маленькое значение слагаемого E_2), 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 векторы. Таким образом, итеративность дополнительного поиска позволяет распространить влияние надежных векторов в векторном поле.

Суммарное значение E в (2) уменьшается по 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 (2). 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 добавляет, как и в работе [7], *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.

Инициализация уровня иерархии

Каждый уровень иерархии инициализируется перед выполнением оценки движения. Для упрощения обозначений будем полагать, что выполняются соотношения

$$\frac{W}{2^{h_{\max}}} \in \mathbb{N}; \quad \frac{H}{2^{h_{\max}}} \in \mathbb{N},$$

где W и H — ширина и высота кадров видеоследовательности соответственно; h_{\max} — индекс максимального уровня иерархии.

При инициализации уровня иерархии выполняются следующие операции:

- подготовка сетки блоков на промежуточном кадре;
- 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 h определяется как $2^h \times 2^h$.

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

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

где $Y = \{1, 2, \dots, H\}$; $X = \{1, 2, \dots, W\}$. Значение пикселя в координате $\mathbf{p} = (y, x)^T$ на кадре \mathbf{F} задается как триплет $\mathbf{F}(\mathbf{p}) = (\mathbf{Y}(\mathbf{p}), \mathbf{Cb}(\mathbf{p}), \mathbf{Cr}(\mathbf{p}))$.

Определим сетку блоков на уровне h как

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

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

$$\mathbf{B}_{i,j}^{(h)} = \{ (y, x)^T \in P \mid y \in Y_i^{(h)}, x \in X_j^{(h)} \},$$

где $Y_i^{(h)} = \{(i-1)2^h + 1, \dots, i2^h\}$; $X_j^{(h)} = \{(j-1)2^h + 1, \dots, j2^h\}$.

Следует отметить, что описанная процедура сопоставления координат пикселей блокам уменьшает размер блока (число пикселей, входящих в блок) с ростом 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, рассчитываются по формуле

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

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

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

Приведенные соотношения завершают описание процедуры подготовки сетки блоков на про-

межуточном 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 умножением пикселей с соответствующей границы кадра. Обозначим множество координат пикселей в расширенном базовом кадре на уровне h через $P^{(h)}$.

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 пены, который можно формально описать как последовательность следующих шагов (обновлений):

$$\mathbf{v}^{(k)}(\mathbf{g}_{i,j}^{(h)}) = \arg \min_{\mathbf{c} \in C_{i,j}^{(k)}} R(\mathbf{c}, \mathbf{g}_{i,j}^{(h)}),$$

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

$$C_{i,j}^{(k)} = \{ \mathbf{c} \mid \mathbf{c} = \mathbf{v}^{(k-1)}(\mathbf{g}_{i,j}^{(h)}) + \mathbf{u} \},$$

где $\mathbf{u} \in U$ — векторы обновления,

$$U = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \pm 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix} \right\}.$$

Начальный вектор для градиентного спуска определяется как

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

где $\mathbf{v}(\mathbf{g}_{i,j}^{(h-1)})$ — вектор с предыдущего уровня иерархии (для первого уровня используется нулевое смещение).

Функция $R(\cdot)$ описывает ошибку сопоставления блока. В данной работе в качестве этой функции предлагается использовать следующий

функционал, основанный на сумме абсолютных разностей (Sum of Absolute Differences):

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

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

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

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

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

Значение коэффициента $\mathbf{w} = (1, 2, 2)$ [13].

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

$$GSS = (k = k_{\max}) \text{ или } \left(\mathbf{v}^{(k)}(\mathbf{g}_{i,j}^{(h)}) = \mathbf{v}^{(k-1)}(\mathbf{g}_{i,j}^{(h)}) \right).$$

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

Результатом процедуры начальной оценки движения являются векторное поле $\mathbf{V}^{(h)} = \{\mathbf{v}_{i,j}^{(h)}\}$ и множество ошибок сопоставления блоков $\mathbf{V}^{(h)} = \{e_{i,j}^{(h)}\}$, $i \in \mathbf{I}^{(h)}$, $j \in \mathbf{J}^{(h)}$.

Итеративный дополнительный поиск

Дополнительный поиск реализован как последовательная итеративная процедура, на каждой итерации которой осуществляются следующие действия:

- 1) расчет надежностей векторов движения;
- 2) дополнительная оценка движения;
- 3) анализ критериев завершения дополнительного поиска.

На каждой итерации n вычисляется уточненная версия $\mathbf{V}_n^{(h)}$ векторного поля с предыдущей итерации $\mathbf{V}_n^{(h-1)}$ (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 надежности основан на подходе, описанном в работе [14], 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 $\mathbf{v}_{i,j}^{(h)}$ осуществляется по следующему правилу:

$$r_{i,j}^{(h)} = \begin{cases} 0, & \text{если } i \in \{1, H^{(h)}\} \text{ или } j \in \{1, W^{(h)}\} \\ 1, & \text{если } e_{i,j}^{(h)} > T_1 \\ 2, & \text{если } s_{i,j}^{(h)} > sa_{i,j}^{(h)} \text{ и } s_{i,j}^{(h)} > T_2 \\ 3, & \text{иначе} \end{cases},$$

где $s_{i,j}^{(h)}$ и $sa_{i,j}^{(h)}$ являются характеристиками локальной гладкости векторного поля и рассчитываются в соответствии с работой [14].

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

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

где $\mathbf{N}(\mathbf{g}_{i,j}^{(h)})$ — множество блоков, соседних с блоком $\mathbf{g}_{i,j}^{(h)}$:

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

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

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

Затем для блока $\mathbf{g}_{i,j}^{(h)}$ выполняется процедура градиентного спуска с использованием векторов из $CS_{i,j}^{(h)}$ в качестве начальных смещений. Вектор с минимальной ошибкой сопоставления выбира-

ется в качестве оптимального для $\mathbf{g}_{i,j}^{(h)}$. 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 $\mathbf{g}_{i,j}^{(h)}$.

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:

$$AS = (n = n_{\max}^{(h)}) \text{ или } \left(\sum_{i=1}^{H^{(h)}} \sum_{j=1}^{W^{(h)}} I(d_{i,j} > T_{s1}^{(h)}) < T_s^{(h)} \right),$$

где $d_{i,j} \in \left| \mathbf{V}_n^{(h)} - \mathbf{V}_{n-1}^{(h)} \right|$ и $I(\cdot)$ — индикаторная функция, определенная как

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

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

Экспериментальные результаты

Для оценки эффективности разработанного алгоритма был проведен ряд экспериментов по интерполяции кадров в реальных видеопоследовательностях для задачи увеличения кадровой скорости в 2 раза. В качестве тестового множества был взят стандартный набор последовательностей [15]. Для выполнения экспериментов все последовательности были предварительно проре-

жены в 2 раза за счет удаления кадров с четными номерами. Затем удаленные кадры были восстановлены с использованием алгоритмов преобразования кадровой скорости. В работе были рассмотрены следующие алгоритмы:

- метод, разработанный в Московском государственном университете (MSU) [16];

- метод, используемый в проекте Smooth Video Project (SVP) [17];

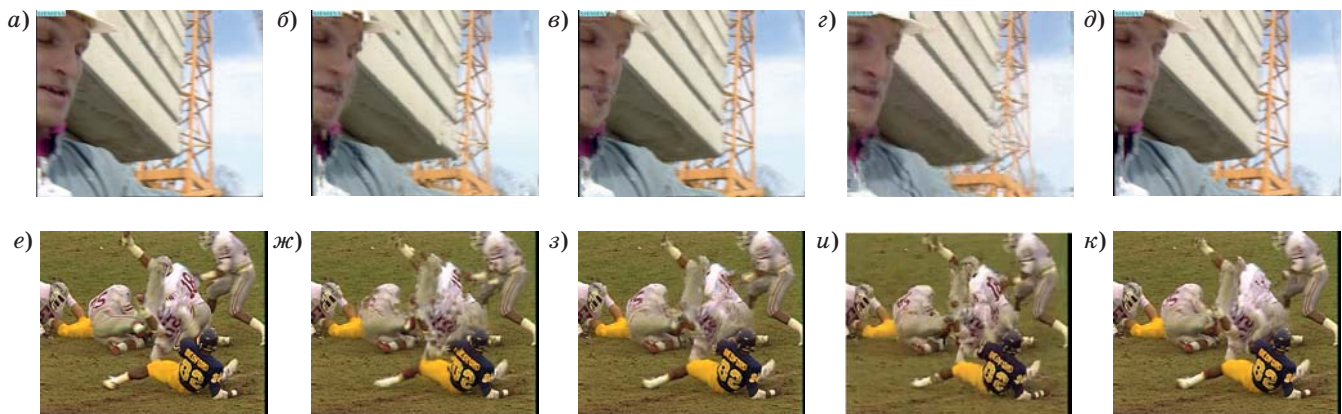
- алгоритм временной интерполяции кадров (Nguyen), описанный в работе [13];

- метод, основанный на предложенной процедуре оценки движения (PROP). В качестве метода компенсации движения в алгоритме используется компенсация движения с перекрытиями блоков (Overlapped Block Motion Compensation [18]).

Сравнение алгоритмов проводилось по критерию среднего отношения сигнал-шум (Peak Signal-to-Noise Ratio) для компоненты Y (Y-PSNR).

Для всех сравниваемых алгоритмов были выставлены параметры, соответствующие максимальному качеству интерполяции. Визуальное сравнение приведено на рис. 3. Объективное сравнение алгоритмов по критерию среднего Y-PSNR приведено в таблице.

Расчет Y-PSNR выполнялся дважды для каждой последовательности: для кадра целиком и для области кадра с некоторым сдвигом от границ. Такой метод расчета связан с тем, что предложенный алгоритм, в отличие от методов SVP и MSU, не использует никакой дополнительной обработки на краях кадров. В связи с этим ошибка предложенного алгоритма в этих регионах существенно превышает ошибку остальных алгоритмов. Использование отступа позволяет провести оценку в тех регионах, где алгоритмы находятся в одинаковых условиях. Видно, что предложенный алгоритм, основанный на простых процедурах оценки и компенсации движения,



■ **Рис. 3.** Визуальное сравнение для 186-го кадра последовательности Foreman: *a* — оригинал; *б* — MSU, PSNR = 26,24; *в* — SVP, PSNR = 27,63; *г* — Nguyen, PSNR = 24,85; *д* — PROP, PSNR = 27,23 — и 220-го кадра последовательности Football: *е* — оригинал; *ж* — MSU, PSNR=21,79; *з* — SVP, PSNR=22,25; *и* — Nguyen, PSNR = 22,21; *к* — PROP, PSNR = 22,54

■ Объективное сравнение по критерию среднего PSNR

Последовательность	PSNR, отступ 0			PSNR, отступ 32		
	MSU	SVP	PROP	MSU	SVP	PROP
Bus	27,99	27,32	26,90	27,68	26,30	26,66
Football	22,94	24,03	24,23	22,46	23,51	23,97
Foreman	34,12	34,35	34,34	34,82	34,91	35,28
Mobile	31,09	31,23	30,28	31,21	31,39	30,33
Soccer	26,40	28,20	27,73	26,40	28,22	28,15
Stefan	26,18	27,98	27,40	26,68	28,10	28,28
Tennis	31,34	32,25	32,25	30,48	31,22	31,35

показывает результаты, сравнимые с результатами сложных алгоритмов интерполяции кадров, а при обработке некраевых блоков даже превосходит эти алгоритмы (кроме последовательностей со сложными текстурными регионами — Bus, Mobile).

Заключение

В статье описан итеративный алгоритм иерархической оценки движения для систем временной интерполяции кадров в видеопоследовательностях. Представлены результаты сравнительного анализа ряда известных алгоритмов интерполяции и простой схемы, использующей предложен-

ный подход. Показано, что разработанный метод оценки движения обеспечивает высокое качество интерполяции промежуточных кадров, сравнимое с результатами более сложных подходов. Предложенный метод поиска движения может быть использован в ряде прикладных задач, например, в устройствах преобразования кадровой скорости, системах распределенного кодирования видеоисточников, алгоритмах сжатия и передачи видеоданных.

Работа выполнена при финансовой поддержке Минобрнауки России в рамках базовой части государственного задания вузу на 2014 год (проект № 2452).

Литература

1. Gedik O., Tretken E., Alatan A. Signals and Communication Technology: High Quality Visual Experience. — Berlin, Springer Berlin Heidelberg, 2010. P. 219–244.
2. Girod B., Aaron A., Rane S., Rebollo-Moderer D. Distributed Video Coding // Proc. of the IEEE. 2005. N 1. P. 71–83. doi:10.1109/JPROC.2004.839619
3. Schwarz H., Marpe D., Wiegand T. Overview of the Scalable Video Coding Extensions of the H.264/AVC Standard // IEEE Transactions on Circuits and Systems for Video Technology. 2007. N 9. P. 1103–1120. doi:10.1109/TCSVT.2007.905532
4. Dikbas S., Altunbasak Y. Novel True-Motion Estimation Algorithm and its Application to Motion-Compensated Temporal Frame Interpolation // IEEE Transactions on Image Processing. 2013. N 8. P. 2931–2945. doi:10.1109/TIP.2012.2222893
5. Jeong S. G., Lee C., Kim C. S. Motion-Compensated Frame Interpolation Based on Multihypothesis Motion Estimation and Texture Optimization // IEEE Transactions on Image Processing. 2013. N 11. P. 4497–4509. doi:10.1109/TIP.2013.2274731
6. Bartels C., De Haan G. Occlusion Classifiers for Picture Rate Conversion // SPIE Digital Library. 2009. N 7257. <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=813521> (дата обращения: 17.04.2014). doi:10.1117/12.805959
7. De Haan G., Biezen P., Huijgen H., Ojo O. True-motion Estimation with 3-D Recursive Search Block Matching // IEEE Transactions on Circuits and Systems for Video Technology. 1993. N 5. P. 368–379. doi:10.1109/76.246088
8. Choi B. T., Lee S. H., Ko S. J. New Frame Rate Up-Conversion Using Bi-Directional Motion Estimation // IEEE Transactions on Consumer Electronics. 2000. N 3. P. 603–609. doi:10.1109/30.883418
9. Rajala S., Abdelqadar I., Bilbro G., Snyder W. Motion Estimation Optimization // Proc. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP 1992), San Francisco, Mar. 23–26, 1992. P. 253–256. doi:10.1109/ICASSP.1992.226203
10. Boykov Y., Kolmogorov V. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. N 9. P. 1124–1137. doi:10.1109/TPAMI.2004.60
11. Braspenning R., De Haan G. True-Motion Estimation Using Feature Correspondences // Proc. of Visual Communications and Image Processing (VCIP 2004), San Jose, Jan. 2004. P. 396–407. doi:10.1117/12.525625
12. Heinrich A., Bartels C., Van Der Vleuten R., Cordes C., De Haan G. Optimization of Hierarchical 3DRS Motion Estimators for Picture Rate Conversion // IEEE J. of Selected Topics in Signal Processing. 2011. N 2. P. 262–274. doi:10.1109/JSTSP.2010.2063014

13. Huang A. M., Nguyen T. Motion Vector Processing Using the Color Information // Proc. of the 16th IEEE Intern. Conf. on Image Processing (ICIP 2009). Nov. 2009. P. 1605–1608. doi:10.1109/ICIP.2009.5413657
14. Huang A. M., Nguyen T. Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion-Compensated Frame Interpolation // IEEE Transaction on Image Processing. 2009. N 4. P. 740–752. doi:10.1109/TIP.2008.2010206
15. Набор тестовых последовательностей Xiph.org. <http://media.xiph.org/video/derf> (дата обращения: 17.04.2014).
16. Фильтр преобразования кадровой скорости MSU. http://www.compression.ru/video/frame_rate_conversion/index_msu.html (дата обращения: 17.04.2014).
17. Фильтр преобразования кадровой скорости SVP. <http://avisynth.org.ru/mvtools/mvtools2.html> (дата обращения: 17.04.2014).
18. Orchard M., Sullivan G. Overlapped Block Motion Compensation: an Estimation-Theoretic Approach // IEEE Transactions on Image Processing. 1994. N 5. P. 693–699. doi:10.1109/83.334974

UDC 004.932.2

Algorithm of Motion Field Optimization for Temporal Frame InterpolationVeselov A. I.^a, Assistant Professor, anton.veselov@gmail.comGilmudtinov M. R.^a, PhD, Tech., Associate Professor, mgilmudtinov@gmail.com^aSaint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaya St., 190000, Saint-Petersburg, Russian Federation

Purpose: Development of a true motion estimation algorithm for temporal frame interpolation. **Results:** The paper proposes a new true motion estimation algorithm which considers both the model of true motion of objects in video sequences and the cases when this model is incorrect, e.g. at the boundaries of the objects. The algorithm is based on iterative search of the local minima for the optimization problem introduced in the framework of the model. The search procedure assumes coarse-to-fine refinement of motion from large objects to small ones, motion reliability estimation and reliability-based smoothing of the motion field. The paper evaluates the algorithm for the problem of frame rate up-conversion on a set of standard test sequences. It is shown that a simple interpolation algorithm based on the proposed motion estimation method demonstrates performance comparable to the existing frame rate up-conversion schemes based on more sophisticated heuristics. **Practical relevance:** The proposed algorithm can enhance the visual quality of temporal frame interpolation (a key component in modern TV receivers) which is an important task for video processing systems.

Keywords — True Motion Estimation, Optimization Problem, Temporal Frame Interpolation, Motion Field, True Motion Model.

References

- Gedik O., Tretken E., Alatan A. *Signals and Communication Technology: High Quality Visual Experience*. Berlin, Springer Berlin Heidelberg, 2010, pp. 219–244.
- Girod B., Aaron A., Rane S., Rebollo-Modero D. Distributed Video Coding. *Proc. of the IEEE*, 2005, vol. 93, no. 1, pp. 71–83. doi:10.1109/JPROC.2004.839619
- Schwarz H., Marpe D., Wiegand T. Overview of the Scalable Video Coding Extensions of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, vol. 17, no. 9, pp. 1103–1120. doi:10.1109/TCSVT.2007.905532
- Dikbas S., Altunbasak Y. Novel True-Motion Estimation Algorithm and its Application to Motion-Compensated Temporal Frame Interpolation. *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 8, pp. 2931–2945. doi:10.1109/TIP.2012.2222893
- Jeong S. G., Lee C., Kim C. S. Motion-Compensated Frame Interpolation Based on Multihypothesis Motion Estimation and Texture Optimization. *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 11, pp. 4497–4509. doi:10.1109/TIP.2013.2274731
- Bartels C., de Haan G. Occlusion Classifiers for Picture Rate Conversion. *SPIE Digital Library*, 2009, vol. 7257. Available at: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=813521> (accessed 17 April 2014). doi:10.1117/12.805959
- De Haan G., Biezen P., Huijgen H., Ojo O. True-motion Estimation with 3-D Recursive Search Block Matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 1993, vol. 3, no. 5, pp. 368–379. doi:10.1109/76.246088
- Choi B. T., Lee S. H., Ko S. J. New Frame Rate Up-Conversion Using Bi-Directional Motion Estimation. *IEEE Transactions on Consumer Electronics*, 2000, vol. 46, no. 3, pp. 603–609. doi:10.1109/30.883418
- Rajala S., Abdelqadar I., Bilbro G., Snyder W. Motion Estimation Optimization. *Proc. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP 1992)*, San Francisco, March 23–26, 1992, pp. 253–256. doi:10.1109/ICASSP.1992.226203
- Boykov Y., Kolmogorov V. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 9, pp. 1124–1137. doi:10.1109/TPAMI.2004.60
- Braspenning R., De Haan G. True-Motion Estimation Using Feature Correspondences. *Proc. of Visual Communications and Image Processing (VCIP 2004)*, San Jose, January 2004, pp. 396–407. doi:10.1117/12.525625
- Heinrich A., Bartels C., Van Der Vleuten R., Cordes C., De Haan G. Optimization of Hierarchical 3DRS Motion Estimators for Picture Rate Conversion. *IEEE Journal of Selected Topics in Signal Processing*, 2011, vol. 5, no. 2, pp. 262–274. doi:10.1109/JSTSP.2010.2063014
- Huang A. M., Nguyen T. Motion Vector Processing Using the Color Information. *Proc. of the 16th IEEE Intern. Conf. on Image Processing (ICIP 2009)*, November 2009, pp. 1605–1608. doi:10.1109/ICIP.2009.5413657
- Huang A. M., Nguyen T. Correlation-Based Motion Vector Processing with Adaptive Interpolation Scheme for Motion-Compensated Frame Interpolation. *IEEE Transaction on Image Processing*, 2009, vol. 18, no. 4, pp. 740–752. doi:10.1109/TIP.2008.2010206
- Nabor testovyykh posledovatel'nostei Xiph.org* [A set of test sequences Xiph.org]. Available at: <http://media.xiph.org/video/derf> (accessed 17 April 2014).
- Fil'tr preobrazovaniia kadrovoi skorosti MSU* [Conversion filter frame rate MSU]. Available at: http://www.compression.ru/video/frame_rate_conversion/index_msu.html (accessed 17 April 2014).
- Fil'tr preobrazovaniia kadrovoi skorosti MSU* [Conversion filter frame rate MSU]. Available at: <http://avisynth.org.ru/mvtools/mvtools2.html> (accessed 17 April 2014).
- Orchard M., Sullivan G. Overlapped Block Motion Compensation: an Estimation-Theoretic Approach. *IEEE Transactions on Image Processing*, 1994, vol. 3, no. 5, pp. 693–699. doi:10.1109/83.334974