

АЛГОРИТМЫ ПРОСТРАНСТВЕННО-ВРЕМЕННОЙ КОРРЕКЦИИ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ В ЗАДАЧАХ СТЕРЕОВИДЕНИЯ

Приведен метод пространственно-временной коррекции видеопоследовательностей в задачах стереовидения. Рассмотрен алгоритм оценки движения и получения интерполяционных кадров в видеопоследовательностях. Приведен алгоритм совмещения видеопоследовательностей для построения стереовидеопоследовательности. Представлены результаты тестирования методов интерполяции и совмещения видеопоследовательностей.

Ключевые слова: видеопоследовательность, изображение, стереовидение, интерполяция.

На сегодняшний день стереосъемка становится все более популярной. Однако по сравнению со съемками в профессиональных студиях, в которых используется специализированное дорогостоящее оборудование, стереосъемка в домашних условиях затруднительна. Это обусловлено тем, что необходимо обеспечивать синхронную запись на несколько видеокамер одновременно. Бюджетный вариант видеокамер не позволяет достичь синхронной записи при частоте 24...30 кадров в секунду ввиду различных факторов: различного времени доступа к устройству хранения, несинхронного старта и т. д. Это обуславливает необходимость длительной обработки и коррекции двух видеопоследовательностей в видеоредакторах. Разработка алгоритмов пространственно-временной коррекции видеопоследовательностей позволяет решить эти проблемы.

Основой стереосъемки является подготовка двух подобных, но не идентичных видеопоследовательностей, полученных с двух точек съемки относительно объекта. В идеальном случае оси видеокамеры и объектов в этих двух положениях должны быть параллельны. При нормальной стереосъемке базис видеосъемки должен быть приблизительно равен расстоянию между центрами зрачков человеческого глаза (примерно 65 мм) [1]. При совмещении двух видеопоследовательностей, снятых с левой и правой видеокамер, получается стереовидеопоследовательность, которую можно преобразовать под метод просмотра: анаглиф, линейную поляризацию, метод затворных ЖК-панелей.

Рассмотрим общую схему метода пространственно-временной коррекции (рис. 1).

1. Осуществляется преобразование каждой из видеопоследовательностей в YUV-пространство. Преобразование связано с получением компоненты яркости Y. Значения яркости пикселей используются для нахождения векторов движения.

2. Выбирается произвольным образом одна из видеопоследовательностей и осуществляется преобразование частоты кадров в ней на основе интерполяции (генерируется несколько кадров между двумя соседними кадрами этой последовательности).

3. Осуществляется совмещение текущего кадра базовой видеопоследовательности (которая не подвергалась процедуре преобразования частоты) с соответ-

ствующими полученными дополнительными кадрами преобразованной видеопоследовательности (интерполированные кадры).

4. Генерируется новая видеопоследовательность, полученная из текущего совмещенного кадра, полученного интерполяцией, путем добавления его в выходной поток.

5. Повторяется шаг 2 до тех пор, пока все кадры базовой видеопоследовательности не будут совмещены.

Это обеспечивает коррекцию одной видеопоследовательности в пространственной и временной области относительно другой (базовой) видеопоследовательности.

Получение интерполированных кадров. Пусть дана видеопоследовательность $L_{in}(t)$, $t \in \{0, 1, 2, 3, \dots\}$. Необходимо вставить между каждыми двумя кадрами определенное число кадров, заданное параметром n . Тогда видеопоследовательность $L_{out}(z)$ после преобразования частоты кадров будет представлена выражением

$$L_{out}(z) = \begin{cases} L_{in}(t), & z = n \cdot t, \quad t \in \{0, 1, 2, 3, \dots\}, \\ L_{in}(t), & z = n \cdot t + l, \quad l \in \{1, 2, 3, \dots, n-1\}, \end{cases} \quad (1)$$

$$z \in \{0, 1, 2, 3, \dots\}, \quad n \in \{2, 3, 4, \dots\},$$

где n – параметр, определяющий, во сколько раз нужно увеличить частоту кадров; z – общее число кадров.

Получение интерполированных кадров производится по следующей общей схеме.

1. Рассматривается текущий кадр $L(t)$ и следующий кадр $L(t+1)$ видеопоследовательности $L_{in}(t)$, $t \in \{0, 1, 2, 3, \dots\}$.

2. Производится оценка движения согласно выbranному алгоритму.

3. Вычисляется поле векторов движения.

4. Производится интерполяция между текущим кадром $L(t)$ и следующим кадром $L(t+1)$.

5. Строятся интерполированные кадры.

Алгоритм оценки движения. Получение интерполированных кадров базируется на оценке движения в соседнем кадре относительно текущего кадра. Для оценки движения был выбран метод сопоставления блоков. Данный метод основан на поиске соответствия между блоками и позволяет с высокой вычислительной эффективностью получать поле векторов для блоков кадров.

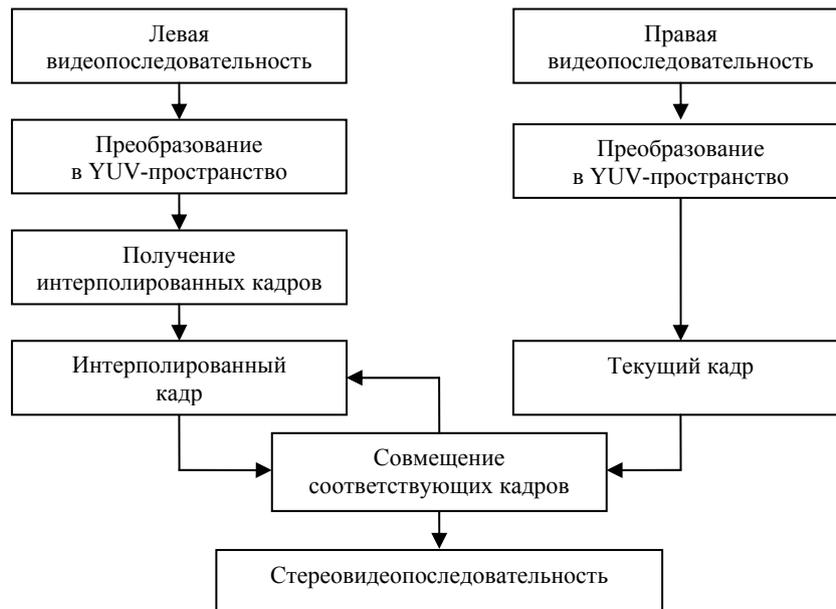


Рис. 1. Общая схема метода пространственно-временной коррекции видеопоследовательностей

Для его реализации под задачу получения стереовидеопоследовательности потребовалась модификация существующих методов. В традиционных методах поиска движения много времени уходит на вычисление смещений между кадрами. Предлагаемый метод основан на модификации метода полного перебора и метода поиска по шаблону.

Оценка движения производится по следующей общей схеме (рис. 2).

1. Рассматривается текущий кадр $L(t)$ и следующий кадр $L(t + 1)$.

2. Кадры $L(t)$ и $L(t + 1)$ разбиваются на равные блоки $B(x, y)$, размер которых кратен двум и задается параметром s_block .

3. На кадре $L(t + 1)$ находится сумма s значений яркости каждого пикселя данного блока $B(x, y)$, определяемая выражением

$$s = \sum_{x=0}^{s_block-1} \sum_{y=0}^{s_block-1} B(Y(x, y)), \quad (2)$$

где $Y(x, y)$ – значение яркости пикселя (x, y) .

4. Определяется, возможно ли движение между кадрами $L(t)$ и $L(t + 1)$ для данного блока $B(x, y)$. Для этого вводится пороговое значение err , которое определяется следующим выражением:

$$err = (255^2 - Y_{lim}^2) \cdot s_block^2, \quad (3)$$

где $Y_{lim} \in [0; 255]$ – параметр, обозначающий предельное значение яркости пикселя (x, y) блока $B(x, y)$ размером $s_block \times s_block$. Чем больше Y_{lim} , тем меньше пороговое значение err . Если $Y_{lim} = 255$, то $err = 0$. Затем сумма s значений яркости пикселей данного блока $B(x, y)$ сравнивается с порогом. Если $s > err$, то движение в блоке ищется (переход к следующему шагу). Если $s < err$, движение в блоке не ищется, возврат к шагу 3 (выбрав следующий блок).

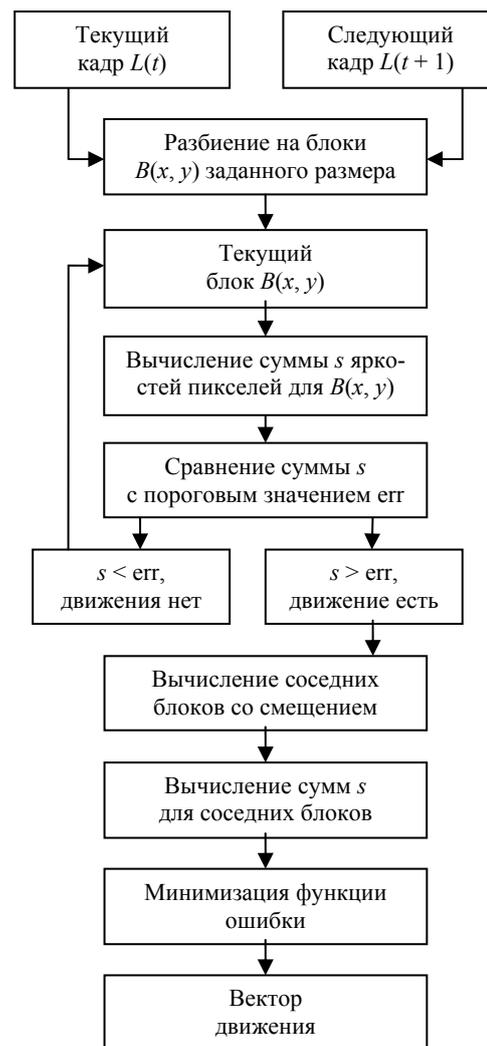


Рис. 2. Общая схема оценки движения

5. Задается параметр, определяющий область поиска движения, называемый смещением *shift*. Данный параметр принимает значения *shift(x)*, где *x* – число пикселей, на величину которых необходимо смещение.

6. На кадре $L(t + 1)$ определяются соседние блоки по диагонали, вертикали и горизонтали относительно рассматриваемого блока $B(x, y)$ с кадра $L(t)$, смещенные от этого блока на величину *shift*.

7. На кадре $L(t + 1)$ аналогично находятся суммы значений пикселей соседних блоков по диагонали, вертикали и горизонтали. Используя выражение (2), получают значения сумм блоков $s_n, n \in \{0, 1, 2, 3, \dots, 8\}$.

8. Для принятия решения о направлении вектора движения необходимо минимизировать функцию ошибки. Суммы значений пикселей блоков $s_n, n \in \{0, 1, 2, 3, \dots, 8\}$ сравниваются поочередно с суммой *s* блока $B(x, y)$ с помощью функции суммы квадратов разностей:

$$\text{move} = \min \left(\sum_{n=0}^8 (s - s_n)^2 \right). \quad (4)$$

Находится соседний блок, содержащий минимальное значение ошибки. Данный блок указывает направление вектора движения для блока $B(x, y)$ на кадре $L(t + 1)$. Направление вектора движения для блока $B(x, y)$ найдено.

9. На кадрах $L(t)$ и $L(t + 1)$ берется следующий блок. Повторяются шаги 4...7 до тех пор, пока все блоки не будут найдены. Таким образом будет получено поле движения векторов.

Данный алгоритм может быть модифицирован. Например, для уменьшения времени выполнения данного алгоритма можно не рассматривать диагональные соседние блоки, а рассматривать только четыре соседних пикселя, но тогда точность нахождения поля векторов движения изменится.

Алгоритм интерполяции кадров. После того как были найдены векторы движения, производится линейная интерполяция кадров поблочко по следующей схеме.

1. Рассматривается текущий кадр $L(t)$ и следующий кадр $L(t + 1)$ видеопоследовательности $L_m(t)$, $t \in \{0, 1, 2, 3, \dots\}$ с найденным вектором движения *move* для блока $B(x, y)$.

2. Пусть *n* – количество интерполированных кадров, которые необходимо вставить между кадрами $L(t)$ и $L(t + 1)$. Рассматривается каждый пиксель C_1 блока $B(x, y)$ на текущем кадре $L(t)$ и каждый пиксель C_2 блока $B(x, y)$ на следующем кадре $L(t + 1)$.

Согласно направлению вектора движения на каждом интерполированном кадре, данный пиксель плавно изменяет свое значение каждой цветовой компоненты (*Y, U, V*) C_n . Для получения плавных интерполированных кадров используется выражение, представляющее собой линейную интерполяцию значений пикселей:

$$C_n = C_1 + (C_2 - C_1) \cdot t, \quad t \in [0; 1], \quad (5)$$

где C_1 – значение цветовой компоненты *Y* пикселя (x, y) блока $B(x, y)$ на текущем кадре $L(t)$; C_2 – значение цветовой компоненты *Y* пикселя (x, y) блока

$B(x, y)$ на следующем кадре $L(t + 1)$; C_n – значение цветовой компоненты *Y* пикселя (x, y) блока $B(x, y)$ на интерполированном кадре *n*; *t* – параметр, задающий линейное смещение пикселей на каждом интерполированном кадре. Параметр *t* вычисляется по формуле

$$t = \frac{m}{n-1}, \quad m \in \{1, 2, 3, \dots, n\}, \quad (6)$$

где *n* – количество интерполированных кадров. Параметр $t \in [0; 1]$, что дает плавное смещение цветовой компоненты для каждого интерполированного кадра.

3. Шаг 2 повторяется для цветовой компонент *U, V*.

4. На кадрах $L(t)$ и $L(t + 1)$ берется следующий блок. Повторяются шаги 2...3 до тех пор, пока все блоки не будут интерполированы. Таким образом, будет получено *n* интерполированных кадров.

Алгоритм совмещения кадров. Заключается в сравнении кадра $L(t)$, интерполированных кадров, полученных между кадрами $L(t)$ и $L(t + 1)$ левой видеопоследовательности, с кадром $R(t)$ правой видеопоследовательности. Находится интерполированный кадр, полученный между кадрами $L(t)$ и $L(t + 1)$ левой видеопоследовательности, наиболее соответствующий кадру $R(t)$ правой видеопоследовательности.

Совмещение кадров производится по следующей общей схеме.

1. Рассматривается левая видеопоследовательность *L* с полученными *n*-интерполированными кадрами $L(z), z \in \{2, 3, \dots, n\}$ между кадрами $L(t)$ и $L(t + 1)$, кадр $L(t)$ и правая видеопоследовательность с кадром $R(t)$;

2. Находится попиксельная межкадровая разность. Для этого кадр $L(t)$ и каждый интерполированный кадр $L(z), z \in \{2, 3, \dots, n\}$ сравнивается с кадром $R(t)$ с помощью функции, которая определяет пиковое отношение сигнала к шуму [2]. PSNR включает в себя среднеквадратическое отклонение и определяется следующим выражением:

$$\text{PSNR}(z) = 20 \log_{10} \frac{\text{MAX}}{\sqrt{\text{MSE}}}, \quad (7)$$

где $\text{PSNR}(z), z \in \{2, 3, \dots, n\}$ – значение схожести между интерполированным кадром $L(z)$ левой видеопоследовательности и кадром $R(t)$ правой видеопоследовательности; $\text{MAX} \in [0; 255]$ – максимальное значение, принимаемое пикселем изображения; MSE – среднеквадратическое отклонение для каждой цветовой компоненты пикселя, вычисляемое по формуле

$$\text{MSE} = \frac{\sum_{x=0}^{h-1} \sum_{y=0}^{w-1} ((Y_2(x, y) - Y_1(x, y))^2)}{3hw} + \frac{\sum_{x=0}^{h-1} \sum_{y=0}^{w-1} ((U_2(x, y) - U_1(x, y))^2 + (V_2(x, y) - V_1(x, y))^2)}{3hw}, \quad (8)$$

где *h* – высота кадра; *w* – ширина кадра; Y_2, U_2, V_2 – значения цветовой компоненты пикселя (x, y) на кадре $R(t)$ правой видеопоследовательности; Y_1, U_1, V_1 – значения цветовой компоненты пикселя (x, y) на интерполированном кадре $L(z)$ левой видеопоследовательности.

3. Аналогично вычисляются значения MSE и $PSNR(L(t))$ для текущего кадра $L(t)$ левой видеопоследовательности L .

4. Среди вычисленных значений PSNR между интерполированным кадром $L(z)$ (или текущим кадром $L(t)$) левой видеопоследовательности и кадром $R(t)$ правой видеопоследовательности находится максимальное значение like по формуле

$$like = \max(PSNR(z), PSNR(L(t))), z \in \{2, 3, \dots, n\}. \quad (9)$$

Интерполированный кадр $L(z)$ либо текущий кадр $L(t)$, межкадровая разность с кадром $R(t)$ правой видеопоследовательности которого является максимальной, и принимается за наилучший кадр.

5. Выбирается следующая пара кадров с левой видеопоследовательности и следующий кадр с правой

видеопоследовательности и повторяются шаги 2–3 до окончательного сравнения всех кадров.

После сравнения кадров полученные стереокадры соединяются. Получается стереовидеопоследовательность, содержащая синхронизированную пару видеопоследовательностей.

Экспериментальные исследования. Для тестирования предложенного метода использовались две видеопоследовательности: *foreman*, где происходит быстрое движение; *container*, где происходит медленное движение. Исследования производились по следующему параметрам: среднее PSNR интерполированных кадров каждой видеопоследовательности, время выполнения метода. Размер смещения для поиска соседних блоков был установлен в один пиксель. Результаты исследований представлены на рис. 3–5.

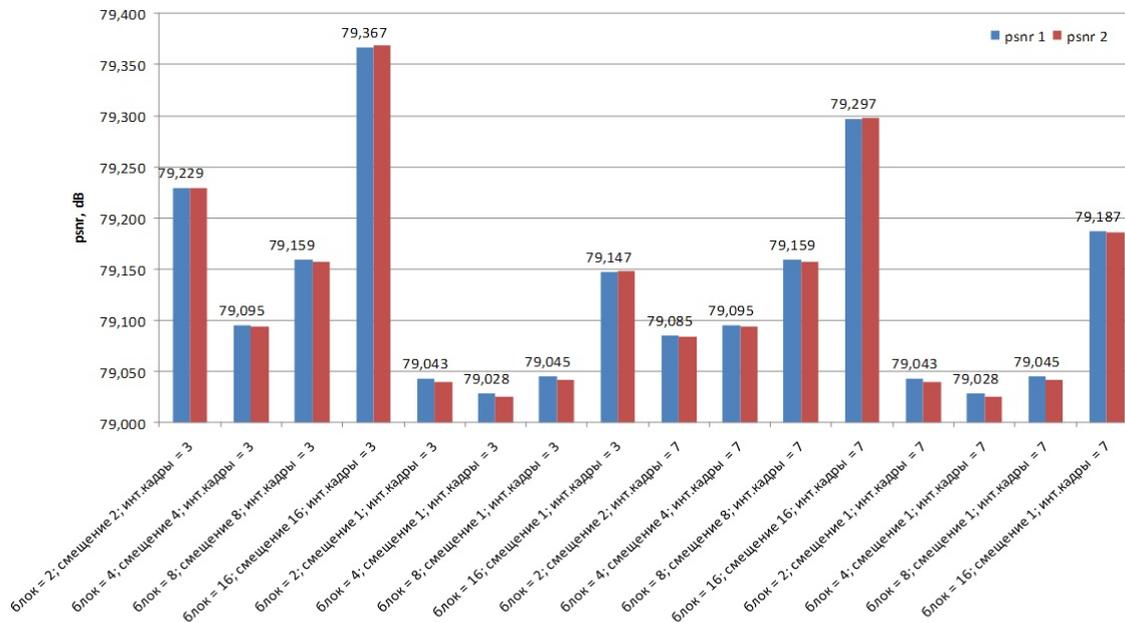


Рис. 3. График PSNR для быстрой видеопоследовательности *foreman*

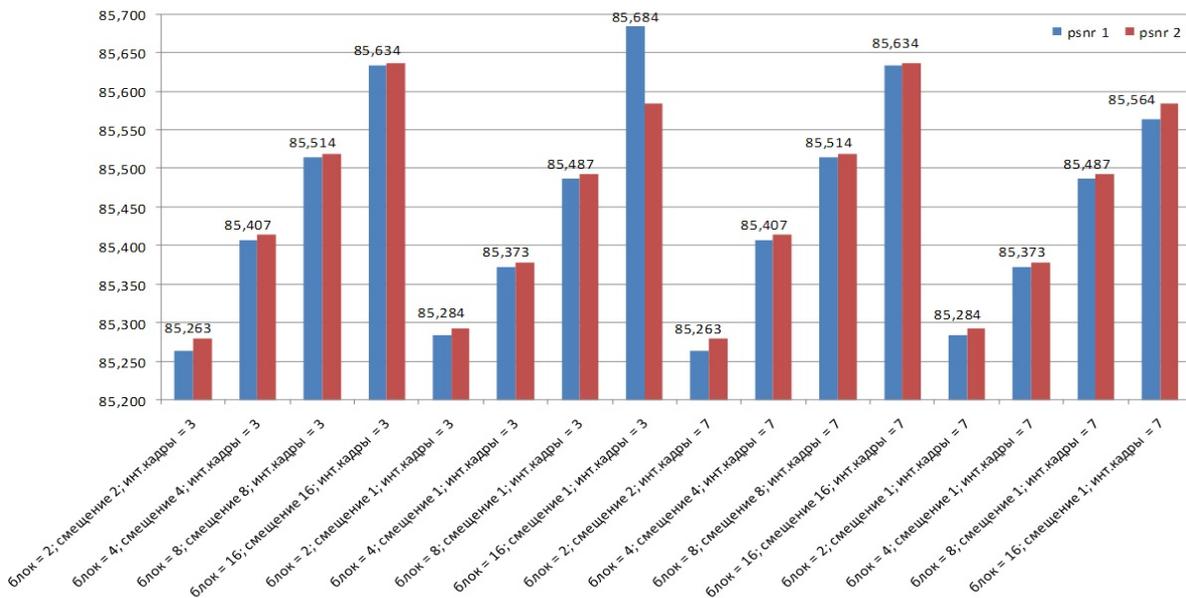


Рис. 4. График PSNR для медленной видеопоследовательности *container*

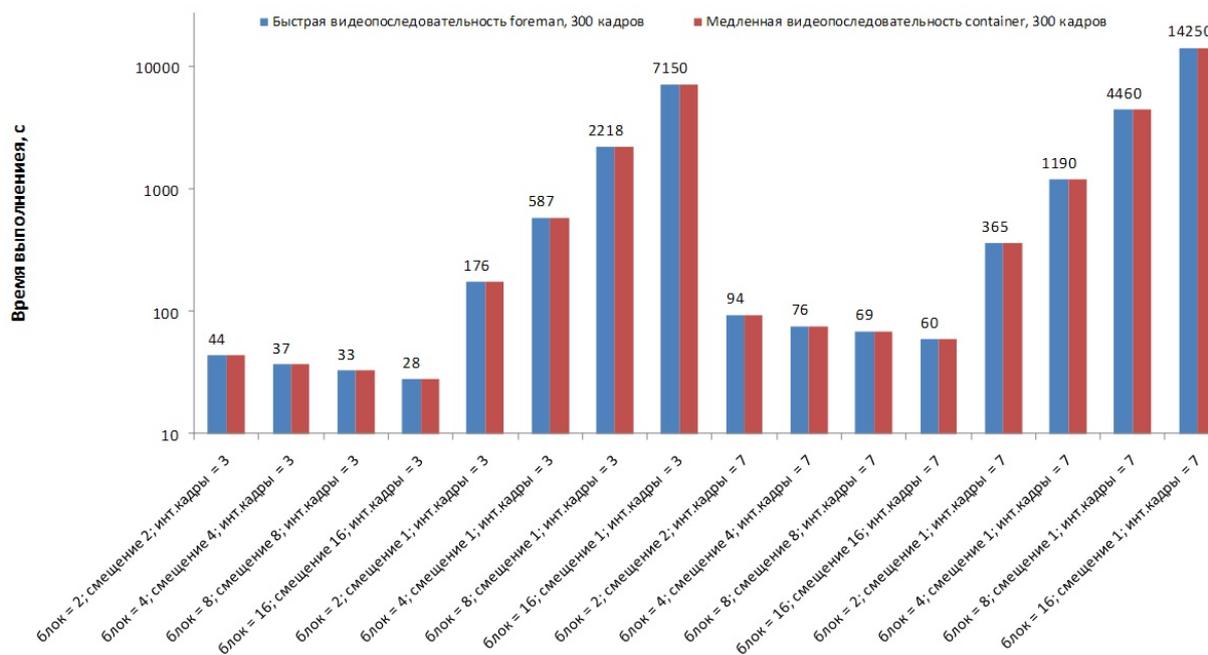


Рис. 5. График времени выполнения алгоритма интерполяции

Параметр PSNR показывает, насколько интерполированные кадры отличаются от текущего, и вычисляется в децибелах по логарифмической шкале. Максимальное значение – 120 децибел, когда кадры полностью одинаковы.

PSNR1 – среднее PSNR интерполированных кадров видеопоследовательности от текущего кадра к следующему. PSNR2 – среднее PSNR интерполированных кадров видеопоследовательности от следующего кадра к текущему. Графики PSNR представлены в логарифмической шкале. Для быстрой видеопоследовательности *foreman* параметр PSNR меньше, чем для медленной видеопоследовательности *container*. Это объясняется тем, что в быстрой видеопоследовательности за счет резкой смены кадров больше шума, чем в медленной видеопоследовательности.

Результаты тестирования быстрой видеопоследовательности показали, что при смещении между блоками в ширину блока на блоке 2×2 получается фактически пиксельный блок, и ошибки там почти нет (практически не находятся вектора движения). При размерах блока 4×4 уже начинает сказываться блочность метода, поэтому и получается самый худший результат. Лучшая точность достигается при большом блоке 16×16 , который уже попадает в области движения и интерполирует корректно. Для медленной видеопоследовательности при увеличении блока интерполированные кадры получаются наиболее точными, так как в такой видеопоследовательности вектора движения имеют большое значение по абсолютной величине [3].

Для обеих видеопоследовательностей время выполнения алгоритма интерполяции одинаково, так как их размеры одинаковы. При смещении между блоками в ширину блока время выполнения алгоритма

уменьшается при увеличении размера блока. При смещении между блоками в один пиксель время выполнения алгоритма увеличивается при увеличении размера блока, так как количество операций возрастает в разы. Уменьшить время работы алгоритма можно, используя адаптивный порог, учитывающий по-пиксельную межкадровую разность, чтобы интерполировать только те блоки, в которых эта разность отличается.

Предложенный метод позволяет строить интерполированные кадры в видеопоследовательностях, совмещать две видеопоследовательности для построения стереовидеопоследовательности. Это обеспечивает пространственно-временную коррекцию. Разработанные алгоритмы можно улучшить, используя различную область поиска движения с помощью моментов Зернике, что является предметом дальнейших исследований.

Библиографические ссылки

1. Вазенмиллер Е. А. Как снимать стереопары [Электронный ресурс] // СтереoAPT. 2001. URL: <http://stereoart.ru/pg.php?page=paperIp0006.html> (дата обращения: 01.11.2010).
2. Ватолин Д. MSU Quality Measurement Tool: Информация о метриках [Электронный ресурс] // Сайт о сжатии. URL: http://www.compression.ru/video/quality_measure/info.html (дата обращения: 01.11.2010).
3. Пьянков Д. И., Горюшкин А. Н. Оценка движения для интерполяции кадров в задачах стереовидения // Материалы 16 Междунар. науч.-техн. конф. «Проблемы передачи и обработки информации в сетях и системах телекоммуникаций». Рязань, 2010. С. 151–152.

**ALGORITHMS OF SPATIO-TEMPORAL CORRECTION OF VIDEO SEQUENCES
IN TASKS OF STEREO VISION**

A method of spatio-temporal correction of video sequences, in tasks of stereo vision, is presented. The algorithm of motion estimation and creation of interpolation frames in video sequences is investigated. The algorithm of image registration for creation of stereo video sequences is presented. Results of interpolation methods testing and video sequences registration are introduced.

Keywords: video sequence, image, stereo vision, interpolation.

© Пьянков Д. И., 2011

УДК 51-74:621

О. В. Репецкий, Буй Мань Кьонг

**РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИКИ И ОЦЕНКИ УСТАЛОСТНОЙ
ПРОЧНОСТИ ЛОПАТОК ТУРБОМАШИН С УЧЕТОМ ГЕОМЕТРИЧЕСКОЙ НЕЛИНЕЙНОСТИ**

Представлены результаты исследований математических моделей для задач динамики лопаток турбомашин с учетом изменения жесткости конструкции лопаток, обусловленной температурой и центробежными силами. Также представлены результаты расчета усталостной прочности лопаток турбомашин на основе разработанных математических моделей. Предложен комплекс рекомендаций по повышению усталостной прочности лопаток турбомашин с учетом вращения и неравномерного нагрева.

Ключевые слова: турбомашин, виброусталость, анализ усталостной прочности, математическое моделирование, численные методы, метод конечных элементов.

Известно, что лопатки турбомашин работают в тяжелых условиях, подвергаясь действию не только переменных аэродинамических нагрузок, но и высокой температуры и большой скорости вращения. Под влиянием этих факторов свойства материала и жесткость конструкции лопаток изменяются. Это приводит к изменениям начальных характеристик работы лопаток. Кроме этого, под действием переменных термомеханических нагрузок лопатки турбомашин часто разрушаются по усталостным причинам. В связи с этим разработка математических моделей для решения задач динамики и оценки долговечности лопаток турбомашин с учетом изменения жесткости конструкции лопаток от температуры и скорости вращения в соответствии с реальными условиями работы, а также обеспечение надежности работы лопаток турбомашин является актуальной проблемой.

В разработанной математической модели влияние температуры и скорости вращения на характеристики работы лопаток турбомашин учтено изменением параметров упругости материала в соответствии с температурой и изменением жесткости конструкции. Для учета изменения жесткости конструкции вследствие температуры и скорости вращения считается, что напряжения от температуры играют роль начальных напряжений конструкции и поэтому в общей матрице жесткости добавлена матрица геометрической жесткости $[K_G]$, обусловленная этими начальными напряжениями. При этом уравнение колебания лопаток

турбомашин с учетом геометрической нелинейности имеет вид

$$[M]\{\ddot{\delta}\} + [C]\{\dot{\delta}\} + ([K] + [K_G])\{\delta\} = \{F(t)\}, \quad (1)$$

где $[M]$, $[C]$ и $[K]$ – матрицы масс, демпфирования и жесткости соответственно; $\{F\}$ – вектор внешней узловой нагрузки; $\{\delta\}$, $\{\dot{\delta}\}$ и $\{\ddot{\delta}\}$ – векторы узловых перемещений, скоростей и ускорений ансамбля конечных элементов.

Матрица геометрической жесткости выражается как [1; 2]

$$[K_G] = \int_V [G^T][S][G]dV, \quad (2)$$

где матрицы $[S]$, $[G]$ определяются следующими уравнениями для трехмерных и двухмерных элементов соответственно:

$$[S] = \begin{bmatrix} \sigma_{xx}^0[I_3] & \sigma_{yx}^0[I_3] & \sigma_{zx}^0[I_3] \\ \sigma_{xy}^0[I_3] & \sigma_{yy}^0[I_3] & \sigma_{zy}^0[I_3] \\ \sigma_{xz}^0[I_3] & \sigma_{yz}^0[I_3] & \sigma_{zz}^0[I_3] \end{bmatrix};$$

$$[G] = \begin{bmatrix} [I_3] \frac{\partial}{\partial x} \\ [I_3] \frac{\partial}{\partial y} \\ [I_3] \frac{\partial}{\partial z} \end{bmatrix} [N]; [I_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$