

УДК 517.51+514.17

## Разбиение железнодорожного состава на отдельные вагоны

Малков А.Н., Штерн Г.П., Михайлов И.А.

*Ярославский государственный университет им. П.Г. Демидова*

*e-mail: ShternGP@yanos.slavneft.ru, waskot@mail.ru*

*получена 3 февраля 2011*

**Ключевые слова:** автоматическая регистрация железнодорожных составов, разбиение состава на отдельные вагоны, эвристический алгоритм

Рассматривается задача, возникающая при создании автоматизированных систем регистрации железнодорожных составов, а именно: разбиение состава на отдельные вагоны (цистерны). Приводится описание алгоритма, предназначенного для решения данной задачи. Входными данными для него служит фильм, в котором можно видеть движение железнодорожного состава. Для тестирования алгоритма проводится эксперимент с использованием 16 различных фильмов.

### Введение

Одна из задач, возникающих при создании автоматизированных систем регистрации железнодорожных составов (см. работы [3, 4]), заключается в разбиении состава на отдельные вагоны. В настоящей работе предлагается алгоритм для решения этой задачи. На вход алгоритму передаётся фильм, полученный с помощью видеокамеры в тот период времени, когда состав двигался в её области видимости; кроме того, передаются параметры, определяющие перспективное искажение изображения вагонов на кадрах фильма. На выходе алгоритм для каждого кадра, на котором виден край некоторого вагона, должен сообщать абсциссу вертикальной линии, отделяющей данный вагон от соседнего вагона (или от фона, если вагон является первым или последним). Задача разбиения состава решается нами путём обнаружения просветов между вагонами на кадрах фильма.

Предлагаемый алгоритм был протестирован с использованием 16 фильмов, один из которых был снят ночью, остальные — днём. Среднее количество кадров в фильмах — 1960. В каждом фильме изображено движение одного состава, в среднем состоящего из 26,5 вагонов. В ходе тестирования алгоритм не пропустил ни одного края и не сообщил о лишнем крае (т.е. о таком, которого не было в действительности), однако расположение вертикальной линии, отделяющей вагоны, на части

кадров было определено неточно. Один из возможных путей улучшения алгоритма — более точная оценка мгновенной скорости состава.

Приведём теперь план работы. Данная работа состоит из 11 разделов.

В первом разделе приведены входные и выходные данные для описываемого в работе алгоритма.

Во втором разделе описаны характеристики входных данных (фильмов), на которых предлагаемый алгоритм был протестирован.

Третий раздел отведён для обозначений и определений, используемых в статье.

В четвёртом приведена структура предлагаемого алгоритма разбиения состава.

В пятом описаны объекты (переменные), используемые в алгоритме и сохраняющие свои значения между вызовами различных процедур.

В шестом представлены общие простые вспомогательные процедуры, используемые затем в более сложных.

В седьмом разделе описаны процедуры и функции, выполняемые на первом основном этапе алгоритма.

Подобным образом восьмой раздел отведён для второго этапа алгоритма, девятый — для третьего, десятый — для четвёртого.

В одиннадцатом разделе приведены результаты работы алгоритма разбиения железнодорожного состава.

## 1 Входные и выходные данные

### *Входные данные*

- Фильм, представляющий собой последовательность полутоновых кадров. Каждая цистерна представлена на нескольких кадрах (примерно от 10 до 100). Кадр имеет размер  $288 \times 352$  пикселя.
- Четыре точки на кадре, задающие горизонтальные прямые в его (кадра) *нормальном прообразе*, то есть изображении, которое было бы получено, если бы горизонтальный поворот видеокамеры относительно состава был равен нулю.

*Выходные данные:* для тех кадров, на которых запечатлён край некоторого вагона, — абсцисса вертикальной линии, отделяющей данный вагон от соседнего вагона (вертикальная линия должна попадать в просвет, если он виден на кадре) или от фона, если вагон является первым или последним; для иных кадров выходные данные отсутствуют.

## 2 Характеристики фильмов

Отметим некоторые особенности фильмов, которые подлежат обработке. Съёмка может производиться в светлое (днём) и тёмное (ночью) время суток. Железнодорожный состав включает в себя локомотив, бустеры (пустые или нагруженные тележки) и цистерны. Окраска цистерн может быть светлой или тёмной, допускаются значительные загрязнения (например, нефтяные потёки). Скорость движения

состава не является постоянной, длина и высота цистерн, запечатлённых в одном и том же фильме, не фиксирована. Расстояние между видеокамерой и составом, горизонтальный поворот между ними и другие параметры, характеризующие их взаимное расположение, могут быть различными для разных фильмов. Верхняя часть цистерн может быть не видна на кадрах. Фон (то, что можно видеть в пролетах между вагонами) в общем случае не является однородным и также может быть различным.

### 3 Обозначения и определения

#### 3.1 Общие обозначения и определения

Приведём вначале некоторые общие обозначения, а также правило построения составных обозначений. Под символом  $f$  будем понимать некоторый обрабатываемый кадр фильма. Символ  $W$  будет обозначать горизонтальный размер кадра, а  $H$  — его вертикальный размер. Для обозначения атрибута какого-либо объекта будем записывать название этого атрибута, а в скобках указывать сам объект. Например, номер кадра  $f$  в содержащем его фильме будем обозначать как  $ind(f)$ .

Пусть имеется некоторое упорядоченное множество  $X = \{x_{i_1}, x_{i_2}, \dots, x_{i_n}\}$ . Чтобы обозначить  $k$ -й элемент этого множества, будем использовать запись  $X(k)$ . Первый и последний элементы множества  $X$  будем обозначать также  $x_1(X)$  и  $x_e(X)$  соответственно. Количество элементов произвольного множества  $S$  будем обозначать как  $|S|$ .

Под *вагоном* будем понимать любую единицу железнодорожного состава: локомотив, бустер или цистерну. Под *элементом вагона* будем понимать область на поверхности одного или двух соседних вагонов (например, светлая вертикальная полоса или стык между вагонами).

#### 3.2 Пометки, серии, основной поток, основное течение

Дадим теперь обозначения и определения структур данных, имеющих непосредственное отношение к решаемой задаче.

*Пометка* — прямоугольная область на кадре, соответствующая некоторому элементу вагона. Формально пометка  $m$  — это тройка  $(ind(f), x_l, x_r)$ , где  $ind(f)$  — номер кадра, на котором обнаружена данная пометка,  $x_l, x_r$  — абсциссы левой и правой границ пометки (в системе координат кадра). Примеры обнаруженных пометок можно видеть на рис. 1 (хотя области пометок ограничены здесь сверху и снизу, к пометке следует относить всю вертикальную полосу, заключённую между вертикальными границами  $x_l$  и  $x_r$ ).

Абсциссу центра пометки  $m$  будем обозначать как  $x(m) = \lfloor \frac{1}{2}(x_l + x_r) \rfloor$ . *Шириной* пометки  $m$  будем называть величину  $l(m) = x_r - x_l + 1$ . Под *смещением* пометки  $m_i$  относительно пометки  $m_j$  будем понимать разницу абсцисс их центров:  $d(m_i, m_j) = x(m_i) - x(m_j)$ .

Пометки группируются в *серии*. Принадлежность пометки к некоторой серии определяется в ходе выполнения процедуры *ADD-MARK*, описанной в разд. 9.5.2.



Рис. 1. Пример обнаруженных пометок

Пометки, составляющие серию, упорядочены по возрастанию номера кадра, на котором была обнаружена та или иная пометка. Серия обладает следующими атрибутами.

- *Вес*  $w(s)$ . Порядок вычисления веса описан далее в разд. 10.3.
- *Множество родительских серий*  $P(s)$ . Порядок определения родителей описан в разд. 9.5.2.

*Основным течением*  $T$  будем называть последовательность серий  $s_{i_1}, s_{i_2}, \dots, s_{i_n}$ , такую что каждая серия в ней, кроме последней, является родительской для следующей серии.

Всякое основное течение  $T$  обладает некоторым весом  $w(T)$ ; в разд. 10.1 формально описано, каким образом вычисляется данный атрибут.

*Основным потоком*  $F$  будем называть ориентированный граф  $G(V, E)$ , построенный следующим образом: множество вершин  $V$  — это множество серий, сформированных в ходе обработки данного фильма; ребро  $(s_i, s_j)$  принадлежит  $E$ , если и только если  $s_i$  принадлежит  $P(s_j)$ . Заметим теперь, что основное течение — это некоторый путь в потоке.

### 3.3 $S$ -пометки, краевой поток, краевое течение

Для определения начальной и конечной скоростей движения состава используется ещё одна структура данных, которую будем называть *s-пометкой*. Основной атрибут *s-пометки* — её абсцисса — определяет некоторый элемент вагона. Формально *s-пометка* — это упорядоченный набор  $(ind(f), x, w)$ , где  $ind(f)$  — номер кадра, на котором она обнаружена,  $x$  — её абсцисса,  $w$  — её вес (порядок вычисления атрибутов *s-пометки* описан в разд. 8.2). Как и у серии, у *s-пометки* существует родительская *s-пометка*  $p(m)$  (только одна). Порядок определения родительских отношений между обнаруженными *s-пометками* указан в разд. 8.3.

Множество *s-пометок*, обнаруженных на одном кадре, будем называть *срезом*.

Последовательность  $s$ -пометок  $m_{i_1}, m_{i_2}, \dots, m_{i_n}$ , такую что  $m_{i_k} = p(m_{i_{k+1}})$ ,  $k = \overline{1, n-1}$ , будем называть *краевым течением*  $T_e$ . Последнюю  $s$ -пометку краевого течения  $T_e$  будем называть *ведущей* и обозначать  $m_d(T_e)$ . Краевое течение, в котором некоторая  $s$ -пометка  $m$  является ведущей, будем обозначать как  $T_e(m)$  и называть *ведомым* указанной  $s$ -пометкой.

Определим дополнительный атрибут  $s$ -пометки. *Поколением*  $s$ -пометки  $g(m_{i_k})$  в представленном выше краевом течении будет называться число, на единицу меньшее её номера в последовательности:  $g(m_{i_k}) = k - 1$ . Таким образом, поколение первой  $s$ -пометки, не имеющей родителя, равно 0.

*Предком*  $s$ -пометки  $m$  будем называть  $s$ -пометку  $p^n(m)$ ,  $0 < n \leq g(m)$ , найденную следующим образом:

$$p^n(m) = \begin{cases} p(m), & \text{если } n = 1; \\ p(p^{n-1}(m)), & \text{если } n > 1. \end{cases}$$

Под обозначением  $p^*(m)$  будем понимать предка  $s$ -пометки  $m$ , поколение которого равно 0.

Расстояние между двумя  $s$ -пометками  $m_i$  и  $m_j$  будем обозначать как  $d(m_i, m_j) = x(m_i) - x(m_j)$ .

*Краевым потоком*  $F_e$  будем называть дерево  $G(V, E)$ , построенное следующим образом: множество вершин  $V$  — это множество  $s$ -пометок, обнаруженных в ходе определения начальной или конечной скоростей движения, корневой  $s$ -пометкой является та, которая была обнаружена раньше остальных; ребро  $(m_i, m_j)$  принадлежит  $E$ , если и только если  $m_i = p(m_j)$ . Заметим теперь, что краевое течение — это некоторый путь в краевом потоке.

Краевое течение, предположительно соответствующее траектории края вагона, будем называть *избранным* и обозначать  $T_{e_d}(F_e)$  (формально порядок определения данного краевого течения описан в разд. 8.3).

## 4 Структура алгоритма разбиения состава

Алгоритм разбиения железнодорожного состава состоит из четырёх основных этапов.

1. Подготовительный этап — определение активной области кадра (см. разд. 7). На данном этапе считывается некоторое количество кадров, распределённых по всему фильму.
2. Этап начала движения. Данный этап включает в себя обработку кадров в течение следующих двух периодов:
  - железнодорожный состав ещё не появился в области видимости;
  - движение края первого вагона от одной границы кадра (момент появления) до другой (момент исчезновения из кадра).

Задача, решаемая в ходе обработки, — оценка начальной скорости движения состава. Действия, выполняемые на данном этапе, описаны в разд. 8. Переход к следующему этапу осуществляется при выполнении двух условий (более чётко условия перехода описаны в разд. 8.3):

- накоплено достаточно данных, чтобы оценить начальную скорость движения;
  - дальнейший сбор данных для её уточнения является затруднительным или невозможен.
3. Основной этап движения. Данный этап включает в себя обработку кадров с момента вычисления начальной скорости движения и до того момента, когда будет обнаружено отсутствие железнодорожного состава в области наблюдения. На данном этапе происходит накопление информации о возможных просветах в основном потоке. Действия, выполняемые на данном этапе, описаны в разд. 9.
  4. Заключительный этап. На данном этапе новые кадры фильма не считываются, происходит обработка накопленных к настоящему моменту данных. Решаемая задача — поиск подходящего основного течения в основном потоке, определяющего разбиение железнодорожного состава на вагоны, и последующее определение координаты разделяющей вертикали для каждого кадра, на котором можно видеть край вагона. Действия, выполняемые на данном этапе, описаны в разд. 10

Заметим, что переход с одного этапа на другой осуществляется в алгоритме на основании результатов обработки данных самого фильма и может в большей или меньшей степени не совпадать с действительностью. Например, если за край первого вагона был ошибочно принят перепад яркости на границе белой полосы, расположенной на тёмном фоне вагона, то выход за пределы области наблюдения будет обнаружен позже, поэтому и переход к третьему этапу произойдёт позже, чем это должно было бы случиться при отсутствии ошибок в обработке.

## 5 Используемые переменные

В предлагаемом алгоритме используются два типа переменных. Переменные первого типа сохраняют свои значения между вызовами одной или нескольких процедур, в которых они используются. Каждая переменная второго типа теряет значение в тот момент, когда работа использующей её процедуры завершается. Прежде чем перечислить основные переменные первого типа, заметим, что остальные переменные (как первого, так и второго типов) будут представлены при описании соответствующих процедур.

1.  $v_c$  — текущая предположительная скорость железнодорожного состава (средняя в пределах области видимости). Единица измерения — пиксель на межкадровый период времени.

2.  $v_{avg}$  — вектор с растущей длиной, содержащий для каждого значения  $i \geq 2$  величину, равную текущей скорости железнодорожного состава  $v_c$  в период времени между  $(i - 1)$ -м и  $i$ -м кадрами. Величина  $v_{avg}(1)$  полагается равной  $v_{avg}(2)$ .
3.  $F_{e\_0}$  — краевой поток для определения начальной скорости состава (в том числе избранное краевое течение в нём  $T_{e\_d}(F_{e\_0})$ ).
4.  $F_{e\_e}$  — краевой поток для определения конечной скорости состава (в том числе избранное краевое течение в нём  $T_{e\_d}(F_{e\_e})$ ).
5.  $h(x) = ax + b$  — функция разницы ординат (см. разд. 8.4).
6.  $c_{ef}$  — количество считанных подряд кадров к текущему моменту времени, не содержащих  $s$ -пометок. Используется на этапе основного движения для определения присутствия железнодорожного состава в кадре.
7.  $f_g$  — величина, на единицу большая количества кадров, пропускаемых между соседними считываемыми; иными словами, программой считывается и обрабатывается только каждый  $f_g$ -й кадр.
8.  $F$  — основной поток.
9.  $S_a$  — множество всех серий, обнаруженных во время основного этапа движения (фактически совпадает с первой компонентой  $V$  основного потока  $F = (V, E)$ ).
10.  $S_l$  — множество серий, обнаруженных на последних кадрах. Порядок формирования данного множества описан в разделах 9.5.1.3 и 9.5.2.
11.  $b_b$  — изображение размера  $W_a \times H$  ( $W_a$  — ширина активной области, см. разд. 7), служащее для накопления в нём частей изображений пометок с целью формирования фонового кадра (*накопительный фоновый кадр*).
12.  $c_{b\_b}$  — вектор размера  $W_a$ ; величина  $c_{b\_b}(x)$  для  $1 \leq x \leq W_a$  показывает, сколько раз к накопительному фоновому кадру была добавлена вертикальная полоса шириной в один пиксель с абсциссой  $x$ , вырезанная из изображения очередной пометки.
13.  $Y$  — очередь, содержащая срезы для последних 18 кадров, обработанных к данному моменту времени (формально порядок заполнения этой очереди описан в разд. 9.1).
14.  $L_{avg}$  — предположительная средняя длина вагона в ж/д составе, движение которого изображено в данном фильме; используется на заключительном этапе.

## 6 Вспомогательные процедуры

В данном разделе будут описаны некоторые сравнительно простые процедуры, используемые нами в более сложных процедурах, представленных далее.

## 6.1 Повышение контрастности изображений

Следующая процедура используется нами для повышения контрастности кадров. Процедура *TRIM-BY-HISTO*.

Входные данные: кадр фильма  $f_c$ .

Выходные данные: преобразованный кадр фильма  $f'_c$ .

1. Из изображения  $f_c^a = b(x, y)$  кадра сформировать изображение  $\bar{b}(x, y)$  с усечённой гистограммой яркости подобно тому, как это было сделано в работе [4] (см. алгоритм выравнивания яркости на с. 106, шаги 1–3).
2. Возвратить изображение  $\bar{b}(x, y)$  в качестве результата.

Опытным путём была найдена подходящая величина отсечения гистограммы (слева и справа), равная 3%.

## 6.2 Общая процедура подготовки кадров

Опишем общую процедуру подготовки кадров к дальнейшей обработке.

Процедура *PREPARE-PIC*.

Входные данные: кадр фильма  $f_c$ .

Выходные данные: подготовленный кадр фильма  $f'_c$ .

1. Выделить активную область кадра  $f_c^a = \tilde{b}(x, y)$  с учётом параметров  $W_l, W_a$  (см. разд. 7).
2.  $\bar{b}(x, y) := \text{TRIM-BY-HISTO}(\tilde{b}(x, y))$ .
3. Возвратить изображение  $\bar{b}(x, y)$  в качестве результата.

## 7 Выделение активной области кадра

В данном разделе описан вспомогательный алгоритм, выполняемый на подготовительном этапе основного алгоритма. Задача, решаемая вспомогательным алгоритмом, состоит в выделении активной области кадра (что это такое, будет описано несколько позже). Цель его выполнения — повышение эффективности алгоритма вычисления начальной и конечной скоростей железнодорожного состава и некоторое ускорение обработки кадра в целом. Входными данными для алгоритма служит фильм, выходными — координаты левой и правой границ активной области. Прежде чем приступить к формальному описанию алгоритма, дадим неформальное определение активной области кадра и представим предположение, лежащее в основе алгоритма её выделения.

Некоторые фильмы из экспериментальной выборки, используемой нами, отличаются следующим свойством: около правой или левой границы каждого кадра можно видеть светлую стену (см. рис. 1). *Активная область кадра* может быть получена



путём отсечения от кадра прямоугольной области высоты  $H$ , примыкающей к левой или правой границе и содержащей изображение стены. Таким образом, активная область задаётся абсциссами своих левой и правой границ. Активная область одинакова для всех кадров заданного фильма.

Предположение, лежащее в основе предлагаемого решения задачи выделения активной области, состоит в том, что часть кадра с изображением стены как совокупность яркостей пикселей мало изменяется на протяжении всего фильма, в то же время часть кадра внутри активной области изменяется значительно, так как здесь можно видеть движение железнодорожного состава. Разумеется, на кадре могут присутствовать области, расположенные выше или ниже состава, которые также изменяются незначительно, но можно говорить о том, что внутри активной области изменения яркостей пикселей происходят в каждом пиксельном столбце.

Опишем теперь алгоритм определения активной области. В нём можно выделить два этапа.

1. Для каждого  $k$ -го кадра фильма выполнить процедуру *PREPROCESS-FRAME* ( $f_k$ ).
2. Возвратить координаты активной области, вычисленные с помощью процедуры *PREPROCESS-FINAL*.

На первом из них для каждого  $k$ -го кадра фильма вычисляется суммарная попиксельная разница между данным кадром и предыдущим считанным; результат вычисления сохраняется (накапливается). На втором этапе на основе высказанного выше предположения и с учётом полученных результатов обработки определяются левая и правая границы активной области.

Перейдём к формальному описанию процедур первого и второго этапов.

## 7.1 Накопление суммарной разницы

Опишем процедуру, выполняемую на первом этапе.

Процедура *PREPROCESS-FRAME* ( $f_c$ ).

Входные данные: текущий кадр  $f_c = b(x, y)$ ; переменные  $c_{fp}$ ,  $b_o$ ,  $b_{o\_a}$ ,  $d_s$ .

В ходе выполнения данной процедуры могут измениться значения следующих переменных:  $c_{fp}$ ,  $b_o$ ,  $b_{o\_a}$ ,  $d_s$ .

1.  $c_{fp} := c_{fp} + 1$ .
2.  $\bar{b} := \text{TRIM-BY-HISTO}(b)$ .
3.  $b_a(x) := \frac{1}{H} \cdot \sum_{1 \leq y \leq H} \bar{b}(x, y)$ ,  $1 \leq x \leq W$ .
4. Если текущий кадр является первым обрабатываемым, то перейти к шагу 7.
5.  $d(x, y) := \left| \frac{b_o(x, y)}{b_{o\_a}(x)} - \frac{\bar{b}(x, y)}{b_a(x)} \right|$ ,  $1 \leq x \leq W, 1 \leq y \leq H$ .

6.  $d_s(x, y) := d_s(x, y) + d(x, y), 1 \leq x \leq W, 1 \leq y \leq H.$
7.  $b_{o\_a}(x) := b_a(x), 1 \leq x \leq W.$
8.  $b_o := \bar{b}.$
9. Завершить процедуру (перейти к следующему кадру).

Предполагается, что объекты  $b_o$  (предыдущий обработанный кадр),  $b_{o\_a}$  (средние яркости в каждом столбце предыдущего обработанного кадра),  $d_s(x, y)$  (суммарная попиксельная разница между соседними кадрами) и  $c_{fp}$  (счётчик обработанных кадров) сохраняют свои значения между вызовами данной процедуры.

## 7.2 Определение границ активной области

Опишем процедуру, выполняемую на втором этапе.

Процедура *PREPROCESS-FINAL*.

Входные данные: суммарная разница между кадрами  $d_s$ , счётчик количества обработанных кадров  $c_{fp}$ .

Выходные данные:  $x_l$  и  $x_r$  — левая и правая границы активной области.

1.  $d_{avg}(x, y) := \frac{d_s(x, y)}{c_{fp}}, 1 \leq x \leq W, 1 \leq y \leq H.$
2. Пусть  $X = \{1 \leq x \leq W \mid \max_{1 \leq y \leq H} d_{avg}(x, y) > d_{th}\}.$
3. Если  $X \neq \emptyset$ , то
  - (a)  $x_l := \min_{x \in X} x;$
  - (b)  $x_r := \max_{x \in X} x.$
4. Если  $X = \emptyset$ , то
  - (a)  $x_l := 1;$
  - (b)  $x_r := W.$
5. Возвратить полученные значения  $x_l$  и  $x_r$ .

Приведём значения констант, используемых во время определения активной области:  $d_{th} = 0,5; k = 25.$

Введём теперь дополнительные обозначения. Под шириной активной области будем понимать величину  $W_a = x_r - x_l + 1$ , левую границу активной области будем обозначать как  $W_l = x_l.$

## 8 Обработка кадров на начальном этапе движения

В данном разделе будут описаны процедуры и функции, предназначенные для обработки кадров на начальном этапе движения. Локальная задача, решаемая на данном этапе, — определение начальной скорости железнодорожного состава. Для решения данной подзадачи предлагается отслеживать движение края первого вагона, а затем, зная пройденный путь и затраченное время, оценить скорость. Для того, чтобы минимизировать погрешности оценки, связанные с неточным определением горизонтальной позиции края вагона, предлагается отслеживать его движение в течение максимально возможного периода времени.

Используемая нами процедура обнаружения края вагона в кадре (описанная в разд. 8.2) обладает следующим свойством: вместе с  $s$ -пометками, более или менее точно определяющими край вагона, она в большинстве случаев возвращает множество  $s$ -пометок, далеко отстоящих от края. Поэтому возникает дополнительная подзадача определения  $s$ -пометки, предположительно соответствующей краю, среди всех выделенных на очередном кадре с учётом результатов обработки предыдущих кадров; решение этой подзадачи описано в разд. 8.3. Как оценить скорость движения в условиях перспективного искажения кадров, описано в разделах 8.4 и 8.5. Основная процедура, вызываемая на начальном этапе движения для обработки каждого считанного кадра, описана в разд. 8.1.

### 8.1 Основная процедура обработки кадров на этапе начального движения

В этом разделе будет приведена основная процедура обработки кадров на этапе начального движения. Используемые нами процедуры *DETECT-MOVE*, *ADD-SLICE*, *ESTIMATE-SPEED* будут описаны в последующих подразделах данного раздела; процедура *ADD-MARK* будет описана в разд. 9.5.2.

Процедура *MOVE-START*( $f_c$ ).

Входные данные: текущий кадр  $f_c$ ; переменные  $F_{e\_0}$ ,  $h(x)$ .

В ходе выполнения данной процедуры могут измениться значения следующих переменных:  $F_{e\_0}$ ,  $v_c$ ,  $v_{avg}$ ,  $c_{ef}$ ,  $f_g$ ,  $S_a$ ,  $S_l$ ,  $F$ .

1.  $f'_c := \text{PREPARE-PIC}(f_c)$ .
2.  $M_s := \text{DETECT-MOVE}(f'_c)$ .
3. Если  $\text{ADD-SLICE}(F_{e\_0}, M_s) = 0$ , то выполнить следующие шаги.
  - (a)  $v_c := \text{ESTIMATE-SPEED}(T_{e\_d}(F_{e\_0}))$ .
  - (b) Для каждой  $s$ -пометки  $m \in T_{e\_d}(F_{e\_0})$ , начиная с первой и заканчивая ведущей, выполнить  $\text{ADD-MARK}(f_c, \text{ind}(f(m)), x(m), x(m))$ .

- (c) Для всех  $1 \leq i_f \leq \text{ind}(f(m_d(T_{e_d}(F_{e_0})))$  выполнить  $v_{avg}(i_f) := v_c$ .
- (d)  $c_{ef} := 0$ .
- (e)  $f_g := \widehat{f}_g$ .
- (f) Перейти к этапу основного движения.

Используемое значение константы  $\widehat{f}_g$  равно 3.

## 8.2 Обнаружение края вагона

В данном разделе будет описана процедура обнаружения края первого или последнего вагонов на кадре. Предположение, лежащее в основе данной процедуры, состоит в следующем. Пусть на некотором кадре можно видеть край вагона. Рассмотрим две области на этом кадре: узкую вертикальную полосу рядом с краем, содержащую изображение части вагона, и такую же узкую полосу с другой стороны края, содержащую фоновое изображение. Область с фоном как совокупность яркостей пикселей осталась почти такой же, как и на предыдущем кадре (относительно среднего уровня яркости), в то же время область, содержащая часть изображения вагона, за тот же межкадровый промежуток изменилась значительно в результате движения состава (предполагается, что изображения частей вагона значительно отличаются от фона). Таким образом, если в кадре найти все вертикальные линии, разделяющие области с указанными свойствами, то, по-видимому, одна из них будет соответствовать краю.

Перейдём к формальному описанию процедуры.

Процедура *DETECT-MOVE*( $f_c$ ).

Входные данные: подготовленный кадр фильма  $f_c = b(x, y)$ .

Выходные данные: множество  $s$ -пометок  $M_s$ .

1. Каждую строку изображения  $b(x, y)$  подвергнуть усредняющей фильтрации (т.е. обработать среднеарифметическим фильтром, см. [2, с. 346–347]) с апертурой  $a_{avg}$  и получить изображение  $\bar{b}(x, y)$  как результат.
2.  $M_s := \emptyset$ .
3. 
$$b_{avg} := \frac{1}{W_a \cdot V_{RM}} \sum_{y=y_{min}}^{y_{max}-1} \sum_{x=1}^{W_a} \bar{b}(x, y),$$
 где  $y_{max} = \frac{1}{2} \cdot (H + V_{RM}) + 1$ ,  $y_{min} = \frac{1}{2} \cdot (H - V_{RM}) + 1$ .
4. 
$$\bar{b}_n(x, y - y_{min} + 1) := \frac{\bar{b}(x, y)}{b_{avg}}, 1 \leq x \leq W_a, y_{min} \leq y \leq y_{max} - 1.$$
5. Если обрабатываемый кадр является первым в последовательности, то перейти к шагу 8.

$$6. l_p := \frac{1}{V_{RM}} \cdot \sum_{y=1}^{V_{RM}} |\bar{b}_n(1, y) - \bar{b}_p(1, y)|.$$

7. Для каждого  $x \in \overline{d_m + 1, W_a}$  выполнить следующие шаги.

$$(a) l := \frac{1}{V_{RM}} \cdot \sum_{y=1}^{V_{RM}} |\bar{b}_n(x, y) - \bar{b}_p(x, y)|.$$

(b) Если  $|l - l_p| > l_{d_{min}}$ , то

$$\bullet M_s := M_s \cup \{(ind(f_c), x, |l - l_p|)\};$$

$$\bullet l_p := \frac{1}{V_{RM}} \cdot \sum_{y=1}^{V_{RM}} |\bar{b}_n(x - d_m + 1, y) - \bar{b}_p(x - d_m + 1, y)|.$$

8.  $\bar{b}_p(x, y) := \bar{b}_n(x, y)$  для  $1 \leq x \leq W_a, 1 \leq y \leq V_{RM}$ .

9. Множество  $M_s$  возвращается в качестве результата.

Предполагается, что матрица  $\bar{b}_p(x, y)$  сохраняет значения всех своих элементов между вызовами данной процедуры. Следующие параметры, используемые в приведённом алгоритме, являются константами, их значения определялись экспериментально:  $V_{RM} = 240$ ;  $d_m = 5$ ;  $l_{d_{min}} = 0,1$ ;  $a_{avg} = 5$ .

Приведённая процедура обладает следующим свойством: если в области видимости видеокамеры движение каких-либо объектов отсутствует или мало (ограничивается колыханием листвы деревьев или травы), то в качестве результата возвращается пустое множество; в противном случае возвращаемый срез не пуст (хотя встречаются такие последовательности кадров, для которых данное свойство не выполняется). Это свойство используется для определения присутствия и отсутствия железнодорожного состава в области видимости.

### 8.3 Отслеживание траектории края вагона

В данном разделе будет описано, как осуществляется отслеживание траектории движения края вагона в кадре, в том числе каким образом выбирается  $s$ -пометка, предположительно соответствующая краю, из всего множества  $s$ -пометок, возвращённого процедурой *DETECT-MOVE*. В предлагаемом решении указанных подзадач используются два предположения: процедура *DETECT-MOVE* возвращает срез, содержащий  $s$ -пометку, соответствующую краю вагона, почти для каждого кадра, на котором этот край можно видеть; скорость движения края в пределах кадра изменяется медленно.

Перечислим вначале некоторые локальные переменные, используемые в данной процедуре.

- $G_l$  — последнее поколение (множество  $s$ -пометок, добавленных к краевому потоку  $F_e$  в ходе предыдущего вызова данной процедуры; исходное значение —  $\emptyset$ );
- $m_d = m_d(T_{e\_d}(F_e))$  — ведущая  $s$ -пометка избранного краевого течения (исходное значение — *nil*).

Приведённые выше переменные, как и сам краевой поток  $F_e$ , сохраняют своё значение между вызовами процедуры *ADD-SLICE*.

Заметим также, что под изначально пустыми множествами  $V$  и  $E$  будем понимать множества вершин и рёбер краевого потока  $F_e = G(V, E)$  соответственно; под  $h(F_e)$  будем понимать высоту краевого потока как дерева:  $h(F_e) = \max_{m \in V} g(m) + 1$  (если  $V = \emptyset$ , то  $h(F_e) = 0$ ). Приведём теперь вспомогательные функции.

1. Суммарное смещение краевого течения  $d(T_e(m)) = d(m, p^*(m))$ .

2. Текущее отклонение  $dcs(m, m_p) = d(m, m_p) - \frac{d(m_p, p^{g_a}(m_p))}{g_a}$ ,  
где  $g_a = \min(g(m_p), \Delta g)$ ,  $\Delta g$  — константа.

3. Суммарное отклонение краевого течения

$$h(T_e(m)) = \begin{cases} 0, & \text{если } g(m) < 2; \\ h(T_e(p(m))) + |dcs(m, p(m))|, & \text{если } g(m) \geq 2. \end{cases}$$

4. Среднее отклонение  $gas(m, m_p) = \frac{h(T_e(m_p)) + |dcs(m, m_p)|}{g(m_p)}$ .

5. Среднее смещение последней части краевого течения  $glad(T_e(m)) = \frac{d(m, p^{g_a}(m))}{g_a}$ ,  
где  $g_a = \min(g(m), \Delta_l g)$ ,  $\Delta_l g$  — константа.

Перейдём к описанию самой процедуры.

Процедура *ADD-SLICE*( $F_e, M_s$ ).

Входные данные: используемый краевой поток  $F_e$ , в том числе избранное краевое течение в нём  $T_{e\_d}(F_e) = T_{e\_d}(m_d)$ , очередной срез  $M_s$ ; переменные  $G_l$  и  $m_d$ .

Выходные данные: 0 в том и только в том случае, если можно непосредственно приступить к оценке скорости, иначе 1.

В ходе выполнения данной процедуры могут измениться значения следующих переменных:  $F_e, T_{e\_d}(F_e), G_l, m_d$ .

1. Если  $M_s = \emptyset$  и  $h(F_e) = 0$ , то завершить процедуру с результатом 1.

2.  $G_n := \emptyset$ .

3.  $m_n := 1$ .

4. Если  $h(F_e) = 0$ , то

- $m_d := \arg \max_{m \in M_s} w(m)$ ;
- $G_n := \{m_d\}$ ;
- $V := \{m_d\}$ ;

5. Если  $h(F_e) = 1$ , то выполнить следующие шаги.

- (a) Для каждой  $m \in M_s$  проверить: если  $f_g v_{min} \leq |d(m, m_d)| \leq f_g v_{max}$ , то
- $p(m) := m_d$ ;
  - $G_n := G_n \cup \{m\}$ ;
  - $V := V \cup \{m\}$ ;
  - $E := E \cup \{(m_d, m)\}$ .
- (b) Если  $|G_n| > 1$ , то  $m_d := nil$ .
- (c) Если  $|G_n| = 1$ , то  $m_d := G_n(1)$ .
6. Если  $h(F_e) > 1$ , то
- (a) Для каждой  $m \in M_s$  выполнить следующие шаги.
- i. Построить множество  $G_l^* \subseteq G_l$ , такое что для всех  $m_i \in G_l^*$  выполняются следующие свойства:
    - $f_g v_{min} \leq |d(m, m_i)| \leq f_g v_{max}$ ;
    - $d(m, m_i) \cdot d(m_i, p(m_i)) > 0$ .
  - ii. Если  $G_l^* \neq \emptyset$ , то
    - $m_p := \arg \min_{m_i \in G_l^*} gas(m, m_i)$ ;
    - если  $gas(m, m_p) < (f_g)^2 h_{p\_max}$ , то
      - $p(m) := m_p$ ;
      - $G_n := G_n \cup \{m\}$ ;
      - $V := V \cup \{m\}$ ;
      - $E := E \cup \{(m_p, m)\}$ .
- (b) Если  $G_n \neq \emptyset$ , то
- $m_d := \arg \max_{m_i \in G_n} \left( |d(T_e(m_i))| - \frac{k_d/h}{f_g} \cdot h(T_e(m_i)) \right)$ ;
  - если  $1 \leq x(m_d) + glad(T_e(m_d)) \leq W_a$ , то  $n_m := 1$ , иначе  $n_m := 0$ .
7. Если  $G_n = \emptyset$ , то выполнить следующие шаги.
- (a) Если  $1 \leq h(F_e) < L_{min}$  или  $d(m_d, p(m_d)) \cdot (x(m_d) - \frac{1}{2} \cdot W_a) < 0$ , то
- $m_d := nil$ ;
  - $F_e := (\emptyset, \emptyset)$ ;
  - завершить процедуру с результатом 1.
- (b) Если условия пункта (a) не выполняются, завершить процедуру с результатом 0.
8. Если  $G_n \neq \emptyset$ , то
- (a)  $G_l := G_n$ ;
- (b) если  $n_m = 0$ , то завершить процедуру с результатом 0, иначе завершить процедуру с результатом 1.

Приведём значения используемых констант:  $\Delta g = 2$ ;  $\Delta l g = 4$ ;  $v_{min} = 1$ ;  $v_{max} = 20$ ;  $h_{p\_max} = 7,0$ ;  $k_d/h = 5,0$ ;  $L_{min} = 6$ .

Прокомментируем работу приведённой процедуры. На шаге 1 проверяются два условия: пустоты среза, переданного на вход, и отсутствия в текущем краевом потоке каких-либо  $s$ -пометок. Если эти условия верны (предположительно это соответствует тому случаю, когда движущийся состав ещё не попал в кадр), то процедура завершается с результатом 1, что даёт возможность скорее перейти к обработке следующего кадра. Шаги 2 и 3 — инициализация используемых переменных, назначение которых разъяснится позже. На шаге 4 рассматривается случай, когда текущий краевой поток ещё пуст (его высота равна 0), но на вход было передано непустое множество  $s$ -пометок. В этом случае среди них выбирается та, вес которой является максимальным. Данная  $s$ -пометка помещается в множество  $G_n$  добавленных к краевому потоку в ходе выполнения данной процедуры  $s$ -пометок. Краевое течение, состоящее только из данной  $s$ -пометки, будет считаться избранным, а сама она — ведущей в данном краевом течении. В разобранным случае, когда в краевой поток добавляются первая  $s$ -пометка, все другие  $s$ -пометки среза отвергаются.

Рассмотрим случай  $h(F_e) = 1$  (следующий шаг процедуры). В этом случае дерево краевого потока уже содержит один корневой элемент. Локальная задача состоит в том, чтобы выбрать из среза те  $s$ -пометки, которые могут соответствовать тому же элементу вагона, что и корневая  $s$ -пометка, обнаруженная на предыдущем кадре. Данная возможность проверяется путём проверки смещения между ними, если оно находится в заданных пределах, то очередная  $s$ -пометка добавляется в краевой поток как дочерняя для корневой. Если была добавлена только одна  $s$ -пометка, то она будет считаться ведущей; если более одной, то избранное краевое течение будет считаться неопределённым (все возможные краевые течения из двух  $s$ -пометок равноправны), поэтому переменной  $m_d$  присваивается значение *nil*.

Рассмотрим теперь случай  $h(F_e) > 1$ . Дерево текущего краевого потока уже содержит не менее двух уровней. Локальная задача состоит в том, чтобы для каждой  $s$ -пометки среза определить, для каких существующих краевых течений данная  $s$ -пометка могла бы стать ведущей и какое из них следует предпочесть. Для решения данной задачи (для каждой  $s$ -пометки среза отдельно) вначале строится дополнительное множество  $s$ -пометок  $G_l^* \subseteq G_l$ . Для того, чтобы некоторая  $s$ -пометка  $m_i \in G_l$  попала в это множество, необходимо выполнение двух условий: во-первых, смещение между текущей  $s$ -пометкой среза и  $s$ -пометкой  $m_i$  должно лежать в заданных пределах, во-вторых, данное смещение и смещение между тестируемой  $s$ -пометкой и её родителем должны быть одного знака (это условие возникает из естественного предположения о том, что железнодорожный состав в пределах одного фильма движется всегда в одном направлении). Затем в множестве  $G_l^*$ , если оно не пусто, выбирается  $s$ -пометка, такая что среднее отклонение краевого течения, ведомого ею и дополненного текущей  $s$ -пометкой среза, было минимальным. Если величина найденного минимума достаточно мала, то текущая  $s$ -пометка среза добавляется в краевой поток как дочерняя для выбранной.

Далее на шаге 6.(b) (это ещё тот же случай  $h(F_e) > 1$ ) выбирается избранное краевое течение с учётом суммарных смещения и отклонения, а затем выполняется прогноз, будет ли следующая  $s$ -пометка избранного краевого течения лежать в



пределах кадра или нет.

После того, как рассмотрены всевозможные значения величины  $h(F_e)$ , необходимо определить возвращаемое значение, т.е. решить, можно ли вычислить скорость на основе уже имеющихся данных (с помощью избранного течения) или стоит проанализировать ещё некоторое количество кадров. На шаге 7 рассматривается случай  $G_n = \emptyset$ , что означает, что в ходе данного вызова процедуры ни одна  $s$ -пометка не была принята. Если высота дерева краевого потока ещё мала (первое условие пункта 7.(а)), то выполняется сброс всех значений: краевой поток считается пустым; процедура возвращает значение 1 с тем, чтобы, начиная со следующего кадра, отслеживание траектории края вагона началось заново. Можно привести два довода в пользу такого решения: во-первых, если избранное краевое течение было прервано, будучи сравнительно коротким, то оно с большой вероятностью может быть ошибочным (не соответствовать движению края вагона, даже быть направленным в противоположную сторону); во-вторых, если оно действительно отображает движение края вагона и всё же было прервано, то можно вновь попытаться обнаружить и отслеживать край вагона, так как этот край, скорее всего, ещё не успел пройти значительного расстояния в кадре.

В случае выполнения второго условия пункта 7.(а) также предпринимается попытка заново отследить движение края. Данное условие истинно в том случае, если в настоящее время опорный элемент вагона, которому соответствует ведущая  $s$ -пометка  $m_d$ , находится в одной половине кадра, а выйти за пределы области видимости должен в другой. Это означает, что у нас есть возможность проследить траекторию края ещё раз, причём её длина может быть больше, чем длина текущей траектории (той, которую задаёт построенное избранное краевое течение). Заметим, что используемая в условии переменная  $m_d$  была определена во время предыдущего вызова процедуры, а не текущего.

Наконец, если оба упомянутых выше условия не верны, то процедура завершается с нулевым результатом, что позволяет непосредственно приступить к оценке скорости.

На шаге 8 рассматривается случай  $G_n \neq \emptyset$ . Вначале сохраняется множество  $G_n$  для использования во время следующего вызова процедуры, затем определяется, будет ли следующая  $s$ -пометка избранного краевого течения лежать (предположительно) в пределах кадра или нет (см. шаг 6.(b)). Если результат отрицательный, возвращается нулевое значение, в противном случае возвращается значение 1.

## 8.4 Оценка скорости с учётом перспективного искажения кадров

В данном разделе будут получены соотношения, позволяющие нам оценить среднюю в пределах области видимости скорость движения элементов вагонов в условиях перспективного искажения кадров.

Так как угол в горизонтальной плоскости между направлением движения состава и направлением видеокамеры может отличаться от прямого, изображение состава в общем случае оказывается искажённым перспективным преобразованием (см. рис. 1). Вследствие данного искажения скорость движения видимых элементов

вагонов (например, их границ или надписей) является различной в пределах кадра.

В связи с этим возникает вопрос: как определить скорость движения, которой затем можно было бы воспользоваться для последующего вычисления длин вагонов. В данной работе в качестве такой величины будет выступать средняя скорость движения в пределах кадра:  $v_{avg} = \pm W_a / \Delta T$  (знак выбирается в зависимости от направления движения), где  $\Delta T$  — период времени, за который некоторый элемент вагона сместится от одной границы кадра к другой (данный период времени, как и любой другой в настоящей работе, выражен в межкадровых промежутках). Однако не всегда удаётся проследить движение элемента вагона на всём протяжении кадра, чаще движение может быть обнаружено лишь на некоторой его части. Например, граница первого вагона может быть запечатлена на первом кадре уже около его середины, поэтому, если пытаться оценить скорость, используя именно этот элемент (границу), то можно брать в расчёт лишь период его движения от середины до одной из границ кадра.

С другой стороны, если оценивать скорость как отношение смещения к соответствующему периоду времени, то полученный результат может оказаться различным в зависимости от того, в какой части кадра обнаружено движение (см. выше). Итак, ближайшая задача состоит в том, чтобы, располагая данными о движении какого-либо элемента в произвольной части кадра, уметь оценивать среднюю скорость движения в пределах всего кадра.

Обратим ещё раз внимание на рис. 1. Можно видеть, как перспективное преобразование переводит прямые горизонтальные линии в сходящиеся прямые (например, один из рельсов и верхняя граница цистерны). Если мысленно представить себе вертикальную линию, движущуюся с постоянной скоростью от одной границы кадра к другой, то можно заметить и то, что расстояние между точками пересечения данной вертикали с выбранными сходящимися прямыми будет изменяться (увеличиваться или уменьшаться) также с постоянной скоростью. Можно сделать предположение о том, что размер элементов вагона линейно зависит от их горизонтальной позиции в кадре, то же самое можно сказать и о скорости движения элементов в пределах кадра.

Итак, будем исходить из того предположения, что скорость как функция разницы ординат точек пересечения вертикали с определёнными сходящимися прямыми является линейной:

$$v(h) = k_v \cdot h(x),$$

где  $h$  — указанная разница ординат,  $x$  — абсцисса вертикали. Аргумент функции скорости был выбран так потому, что его (т.е. разность ординат как функцию абсциссы вертикали) легко выразить через входные данные (см. конец этого раздела). Параметр  $k_v$  является постоянным на протяжении всего кадра (для любого  $x$ ); его отношение к реальной скорости состава — постоянная положительная величина в пределах одного фильма.

Для решения поставленной задачи выразим вначале функцию  $h(x)$  через гра-

ничные значения (выше было показано, что она является линейной).

$$\begin{aligned} h(x) &= h(x_e) \cdot \frac{x - x_0}{x_e - x_0} + h(x_0) \cdot \frac{x_e - x}{x_e - x_0} = \frac{1}{l} \cdot (h(x_e)(x - x_0) + h(x_0)(x_e - x)) = \\ &= \frac{1}{l} \cdot (x(h(x_e) - h(x_0)) + x_e h(x_0) - x_0 h(x_e)) = \frac{1}{l} \cdot (a_l x + b_l), \end{aligned}$$

где  $l = x_e - x_0$ ,  $a_l = h(x_e) - h(x_0)$ ,  $b_l = x_e h(x_0) - x_0 h(x_e)$ .

Теперь попытаемся выразить время, необходимое некоторому элементу вагона, чтобы пройти путь от вертикали с абсциссой  $x_0$  к вертикали с абсциссой  $x_e$ , при условии, что функция  $h(x)$  и параметр  $k_v$  известны.

$$\begin{aligned} \Delta t &= \int_{x_0}^{x_e} \frac{dx}{k_v h(x)} = \frac{l}{k_v} \int_{x_0}^{x_e} \frac{dx}{a_l x + b_l} = \frac{l}{k_v a_l} \int_{x_0}^{x_e} \frac{a_l dx}{a_l x + b_l} = \\ &= \frac{l}{k_v a_l} \cdot \ln(a_l x + b_l) \Big|_{x_0}^{x_e} = \frac{l}{k_v a_l} \cdot \ln \frac{a_l x_e + b_l}{a_l x_0 + b_l}. \end{aligned}$$

После обратной подстановки  $a_l$ ,  $b_l$  и  $l$  получаем

$$\begin{aligned} \Delta t &= \frac{l}{k_v a_l} \cdot \ln \frac{a_l x_e + b_l}{a_l x_0 + b_l} = \frac{l}{k_v a_l} \cdot \ln \frac{h(x_e)(x_e - x_0)}{h(x_0)(x_e - x_0)} = \\ &= \frac{x_e - x_0}{k_v (h(x_e) - h(x_0))} \cdot \ln \frac{h(x_e)}{h(x_0)}. \end{aligned} \quad (1)$$

Заметим, что независимо от знаков величин  $a_l$  и  $l$  величина  $\Delta t$  всегда является положительной благодаря тому, что в любом случае выполняется  $k_v \cdot l > 0$  (случай  $x_0 = x_e$  не рассматривается).

Из полученного соотношения легко выразить параметр  $k_v$  (если  $\Delta t$  известно).

$$k_v = \frac{x_e - x_0}{\Delta t (h(x_e) - h(x_0))} \cdot \ln \frac{h(x_e)}{h(x_0)}. \quad (2)$$

Теперь можно узнать, сколько времени потребуется, чтобы пройти путь от одной границы кадра до другой. После подстановки  $x_0 = 0$ ,  $x_e = W_a$  (случай движения слева направо) в соотношение (1) получим

$$\Delta T = \frac{W_a}{k_v (h(W_a) - h(0))} \cdot \ln \frac{h(W_a)}{h(0)}.$$

Наконец, можно выразить среднюю в пределах кадра скорость.

$$v_{avg} = \frac{W_a}{\Delta T} = \frac{k_v (h(W_a) - h(0))}{\ln h(W_a) - \ln h(0)}. \quad (3)$$

Нетрудно проверить, что в случае  $x_0 = W_a$ ,  $x_e = 0$  (движение справа налево) последнее соотношение получается в точности таким же.

Для полноты описания выразим функцию  $h(x)$  через входные данные. Пусть известны четыре точки  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ ,  $(x_4, y_4)$ , первые две из которых задают прямую  $l_1$ , вторые две — прямую  $l_2$ , прямые  $l_1$  и  $l_2$  являются прообразами горизонтальных прямых.

Выпишем уравнения прямых:

$$y = \frac{y_2 - y_1}{x_2 - x_1}x + \frac{x_2y_1 - x_1y_2}{x_2 - x_1} = a_1x + b_1,$$

$$y = \frac{y_4 - y_3}{x_4 - x_3}x + \frac{x_4y_3 - x_3y_4}{x_4 - x_3} = a_2x + b_2,$$

где  $a_1 = \frac{y_2 - y_1}{x_2 - x_1}$ ,  $b_1 = \frac{x_2y_1 - x_1y_2}{x_2 - x_1}$ ,  $a_2 = \frac{y_4 - y_3}{x_4 - x_3}$ ,  $b_2 = \frac{x_4y_3 - x_3y_4}{x_4 - x_3}$ .

Для определённости будем полагать, что  $(a_2x + b_2) - (a_1x + b_1) > 0$  для  $0 \leq x \leq W_a$ . Выразим теперь функцию  $h(x)$ :

$$h(x) = (a_2 - a_1)x + b_2 - b_1 = ax + b, \quad (4)$$

где  $a = a_2 - a_1$ ,  $b = b_2 - b_1$ .

В заключение заметим, что, если геометрическое искажение, вносимое перспективным преобразованием, достаточно мало (это можно определить по величине  $a$ ), то скорость в разных частях кадра можно считать одинаковой.

## 8.5 Непосредственная оценка скорости

В данном разделе будет описана процедура непосредственной оценки скорости железнодорожного состава.

Процедура *ESTIMATE-SPEED* ( $T_e(m)$ ).

Входные данные: краевое течение  $T_e(m)$ ; уравнение функции  $h(x) = ax + b$ .

Выходные данные: средняя скорость в пределах кадра  $v_{avg}$ .

1. Если  $|T_e(m)| < L_{min}$ , то завершить процедуру с результатом *nil*.
2. Если  $|a| \geq a_{min}$ , то
  - воспользовавшись соотношением (2) и положив  $x_0 = x(p^*(m))$ ,  $x_e = x(m)$ ,  $\Delta t = ind(f(m)) - ind(f(p^*(m)))$ , вычислить параметр  $k_v$ ;
  - воспользовавшись соотношением (3), найти среднюю скорость  $v_{avg}$ .
3. Если  $|a| < a_{min}$ , то вычислить скорость следующим образом:
 
$$v_{avg} := \frac{x(m) - x(p^*(m))}{ind(f(m)) - ind(f(p^*(m)))}$$
4. Найденное значение  $v_{avg}$  вернуть в качестве результата.

Специальное возвращаемое значение *nil* свидетельствует о том, что оценка скорости закончилась неудачно. Значение используемой константы  $a_{min}$  в приведённой процедуре равно 0,02.

## 9 Этап основного движения

Основная задача, решаемая на данном этапе, заключается в обнаружении просветов на каждом обрабатываемом кадре, группировании выделенных пометок в серии и определении родительских отношений между сериями. Процедура, выполняемая на этом этапе для каждого считанного кадра, описана в разд. 9.1.

### 9.1 Процедура обработки кадров на этапе основного движения

В этом разделе будет описана процедура обработки кадров на этапе основного движения. Используемые вспомогательные процедуры *FIND-CHINKS*, *ADD-MARK*, *ESTIMATE-CURRENT-SPEED*, *ESTIMATE-FINAL-SPEED* будут описаны в последующих подразделах разд. 9.

Процедура *MAIN-MOVE* ( $f_c$ ).

Входные данные: очередной неподготовленный кадр  $f_c = b(x, y)$ ; переменные  $c_{ef}$ ,  $f_g$ ,  $v_c$ ,  $v_{avg}$ ,  $F$ ,  $S_l$ ,  $h(x)$ ,  $Y$ .

В ходе выполнения данной процедуры могут измениться значения следующих переменных:  $c_{ef}$ ,  $v_c$ ,  $v_{avg}$ ,  $F$ ,  $S_a$ ,  $S_l$ ,  $Y$ ,  $b_b$ ,  $c_{b\_b}$ .

1.  $f'_c := \text{PREPARE-PIC}(f_c)$ .
2.  $M_s := \text{DETECT-MOVE}(f'_c)$ .
3.  $Y \leftarrow M_s$ .
4. Если  $M_s = \emptyset$ , то выполнить следующие шаги.
  - (a)  $c_{ef} := c_{ef} + 1$ .
  - (b) Если  $c_{ef} = c_{ef\_max}$ , то выполнить следующие шаги.
    - i.  $\text{ADD-MARK}(f_c, \text{ind}(f_c) - c_{ef}f_g, x, x)$ , где  $x = W_a$ , если  $v_c > 0$ , и  $x = 1$  в противном случае.
    - ii.  $v_e := \text{ESTIMATE-FINAL-SPEED}(Y)$ .
    - iii. Если  $v_e \neq \text{nil}$ , то
      - пусть  $i_1$  — номер кадра, срез для которого является в  $Y$  первым;
      - для каждого  $i_1 \leq i \leq |v_{avg}|$  выполнить  $v_{avg}(i) := v_e$ .
    - iv. Перейти к заключительному этапу обработки.
5. Если  $M_s \neq \emptyset$ , то выполнить следующие шаги.
  - (a)  $c_{ef} := 0$ .
  - (b)  $M := \text{FIND-CHINKS}(f'_c)$ .
  - (c) Если  $M \neq \emptyset$ , то для каждой  $m \in M$  выполнить  $\text{ADD-MARK}(f_c, m)$ .

6.  $v_c := ESTIMATE-CURRENT-SPEED(ind(f_c))$ .

7. Для каждого  $1 \leq i \leq f_g$  выполнить  
 $v_{avg}(ind(f_c) + i - f_g) := v_c$ .

Прокомментируем работу процедуры. Шаг 1 — подготовка кадра для дальнейшей обработки. На втором шаге выполняется попытка обнаружения движения в области видимости видеокамеры. На третьем шаге текущий срез добавляется в очередь  $Y$  (в том случае, если очередь полна, её первый элемент будет удалён).

Четвёртый шаг соответствует тому случаю, когда движение на кадре отсутствует. Переменная  $c_{ef}$ , задающая количество считанных подряд кадров с пустыми срезами на текущий момент времени, увеличивается на единицу на шаге 4.(a). Достижение данной переменной определённого предела, который задаётся константой  $c_{ef\_max} = 10$  (условие 4.(b)), предположительно означает, что железнодорожный состав некоторое время назад вышел за пределы области видимости (см. разд. 8.2), поэтому можно переходить к заключительному этапу обработки. Перед тем, как выполнить данный переход, порождается и обрабатывается искусственная пометка (шаг 4.(b).i), предположительно определяющая край последнего вагона на том кадре, на котором этот вагон выходит за пределы области видимости. Этот шаг необходим для того, чтобы существовала серия, соответствующая концу железнодорожного состава. Далее предпринимается попытка оценить скорость и в случае успеха выполняется обновление конечной части вектора скорости (шаги 4.(b).ii, iii).

Если на текущем кадре присутствует движение (условие 5; это предположительно означает, что состав ещё находится в области видимости), то после обновления переменной  $c_{ef}$  выполняется попытка обнаружить просветы на кадре, а каждая выделенная пометка обрабатывается (шаги 5.(b, c)). Наконец, на шестом и седьмом шагах производится оценка текущей скорости и соответствующее обновление вектора скорости.

Опишем теперь назначение очереди  $Y$ . Предположим, что за период времени, когда железнодорожный состав покидает область видимости (точнее, когда просвет между последним и предпоследним вагонами уже не виден, а край последнего вагона ещё не вышел за пределы кадра), ни одной пометки обнаружено не было, а состав изменил свою скорость. Это значит, что новое (более точное) значение скорости в этот период не может быть получено с помощью процедуры *ESTIMATE-CURRENT-SPEED*, так как данная процедура для вычисления скорости использует характеристики пометок некоторой серии (см. разд. 9.2). Если считать, что скорость не изменилась за указанный период, то длина последнего вагона будет вычислена неверно (в большей или меньшей степени), что может привести к серьёзной ошибке в работе всего алгоритма разделения состава.

Очередь  $Y$  содержит срезы определённого количества последних обработанных кадров и служит для того, чтобы была возможность оценить скорость состава в то время, когда он покидает область видимости. Оценка выполняется подобно тому, как вычисляется скорость состава в начале его движения в области видимости, то есть путём отслеживания траектории края вагона.

## 9.2 Оценка текущей скорости железнодорожного состава

В этом разделе будет описана процедура оценки текущей скорости состава на этапе основного движения.

Процедура *ESTIMATE-CURRENT-SPEED* ( $ind(f_c)$ ).

Входные данные: номер текущего кадра  $ind(f_c)$ ; множество последних серий  $S_l$ , функция  $h(x) = ax + b$ .

Выходные данные: предположительная величина текущей скорости состава  $v_c$ .

1. Построить множество  $S_l^* \subseteq S_l$ , такое что для каждой серии  $s \in S_l^*$  выполняются следующие свойства:
  - $|s| \geq c_{m\_min}$ ;
  - $|d(m_1(s), m_e(s))| \geq d_{min}$ ;
  - пометка  $m_e(s)$  была обнаружена на текущем кадре  $f_c$ .
2. Если  $S_l^* = \emptyset$ , то вернуть текущее значение переменной  $v_c$  в качестве результата и завершить процедуру.
3.  $s^* := \arg \max_{s \in S_l^*} |s|$ .
4.  $x_0 := x(m_1(s^*)), x_e := x(m_e(s^*)), \Delta t := ind(f(m_e(s^*))) - ind(f(m_1(s^*)))$ .
5. Если  $|a| \geq a_{min}$ , то, воспользовавшись соотношениями (2) и (3), найти среднюю скорость  $v_{avg}$ .
6. Если  $|a| < a_{min}$ , то вычислить среднюю скорость следующим образом:
 
$$v_{avg} := \frac{x_e - x_0}{\Delta t}.$$
7.  $c_p := \max\{0, c_{p\_max} - |s^*|\}$ .
8.  $v_c := \frac{|s^*|v_{avg} + c_p v_c}{|s^*| + c_p}$ .
9. Найденную величину  $v_c$  вернуть в качестве результата.

Приведём значения используемых констант:  $a_{min} = 0,02$ ;  $c_{m\_min} = 2$ ;  $d_{min} = 20$ . Константа  $c_{p\_max} = 10$  определяет степень сдерживания изменения скорости. Указанное сдерживание состоит в том, чтобы учитывать текущее значение переменной  $v_c$  при расчёте её нового значения (шаги 7 и 8). Такой порядок вычисления применяется в данной процедуре по следующим причинам. Встречаются фильмы, при обработке которых пометки, соответствующие некоторому просвету, могут быть обнаружены не на всех кадрах, на которых данный просвет присутствует, а лишь на небольшой их части. В этом случае вполне возможна ситуация, когда в течение длительного периода времени для оценки скорости будут выбираться серии, содержащие малое количество пометок. С другой стороны, координаты пометок могут быть определены со значительной погрешностью. Поэтому скорость, вычисленная с

помощью приведённой процедуры без сдерживания (при  $c_{p\_max} = 0$ ), может значительно отличаться от действительной в течение указанного периода времени, что, в свою очередь, может привести к серьёзным ошибкам в результатах работы алгоритма разделения состава.

Таким образом, задача состоит в том, чтобы скорректировать оценку скорости при использовании серий с малым количеством пометок. Нами используется то предположение, что действительная скорость состава изменяется достаточно медленно, что позволяет применять для коррекции оценки величину  $v_c$ , найденную во время предыдущего запуска данной процедуры. В то же время при использовании серий с большим количеством пометок текущее значение величины  $v_c$  не учитывается, так как погрешность вычисления, вызванная ошибочным определением координат пометок, в данном случае предполагается небольшой.

### 9.3 Оценка скорости во время выхода железнодорожного состава из области видимости

В этом разделе будет описана процедура оценки скорости в конце этапа основного движения.

Процедура *ESTIMATE-FINAL-SPEED* ( $Y$ ).

Входные данные: множество  $Y$ ; краевой поток  $F_{e\_e}$ .

Выходные данные: предположительная скорость в конце движения  $v_e$ .

В ходе выполнения данной процедуры может измениться значение переменной  $F_{e\_e}$  (в том числе  $T_{e\_d}(F_{e\_e})$ ).

1. Для каждого среза  $M_s \in Y$ , начиная с последнего и заканчивая первым, проверить: если  $ADD-SLICE(F_{e\_e}, M_s) = 0$ , то немедленно перейти к шагу 2.
2.  $v_e := ESTIMATE-SPEED(T_{e\_d}(F_{e\_e}))$ .
3. Вернуть величину  $v_e$  как результат.

### 9.4 Обнаружение просветов в кадре

В данном разделе будет описана процедура поиска просветов в кадре. Предположение, лежащее в основе данной процедуры, состоит в том, что средняя яркость просвета больше, чем яркость элементов вагонов, окружающих данный просвет.

Приведём вначале описания и значения используемых в процедуре констант:

- $V_{RG} = 200$  — ширина горизонтальной полосы кадра, используемой для вычисления средних значений яркости в каждом столбце;
- $a_{med} = 41$  — апертура медианного фильтра;
- $w_{max} = 110$  — максимальная допустимая ширина просвета;
- $\Delta_{min}\bar{b} = 20$  — минимальный допустимый перепад яркости между максимумом в области просвета и значениями на его границе;



- $k = 0,5$  — коэффициент для уточнения границ просвета.

Приведём описание самой процедуры.

Процедура *FIND-CHINKS* ( $f_c$ ).

Входные данные: очередной подготовленный кадр  $f_c = b(x, y)$ .

Выходные данные: множество обнаруженных на данном кадре пометок  $M$ , упорядоченное по их абсциссам.

1.  $M := \emptyset$ .

$$2. b_{avg}(x) := \left[ \frac{1}{V_{RG}} \cdot \sum_{y=y_{min}}^{y_{max}-1} b(x, y) \right] \text{ для } 1 \leq x \leq W_a,$$

$$\text{где } y_{max} = \frac{1}{2} \cdot (H + V_{RG}) + 1, y_{min} = \frac{1}{2} \cdot (H - V_{RG}) + 1.$$

3. Обработать сигнал  $b_{avg}(x)$  медианным фильтром (см. [1, с. 342–343]) с апертурой  $a_{med}$  и получить  $\bar{b}(x)$  как результат.

4. Для всех  $2 \leq x \leq W_a - w_{max}$  в порядке возрастания выполнить следующие шаги.

(a)  $x_l := x, x_r := x + w_{max} - 1$ .

(b)  $x_{max} := \arg \max_{x_l \leq x_i \leq x_r} \bar{b}(x_i)$ .

(c) Если  $|\bar{b}(x_{max}) - \max \{\bar{b}(x_l), \bar{b}(x_r)\}| \geq \Delta_{min} \bar{b}$ , то выполнить следующие шаги.

i.  $x'_l := \min \{x_l < x_i \leq x_{max} : \bar{b}(x_i) - \bar{b}(x_l) > k(\bar{b}(x_{max}) - \bar{b}(x_l))\}$ .

ii.  $x'_r := \max \{x_{max} \leq x_i < x_r : \bar{b}(x_i) - \bar{b}(x_r) > k(\bar{b}(x_{max}) - \bar{b}(x_r))\}$ .

iii. Если  $M \neq \emptyset$ , то

- пусть  $m_l$  — последняя пометка, добавленная в  $M$ ;

- если  $x'_l \leq x_r(m_l)$ , то перейти к анализу следующего элемента  $x$  (шаг 4) без добавления новой пометки в  $M$ .

iv.  $M := M \cup \{(ind(f_c), x'_l, x'_r)\}$ .

5. Возвратить множество  $M$  в качестве результата.

Прокомментируем работу приведённой процедуры. Шаг 2 — формирование одномерного сигнала, содержащего средние значения яркости в каждом столбце. Столбцы используются не полностью, а частично (часть определяется константой  $V_{RG}$ ) для того, чтобы исключить из рассмотрения те области кадра, которые, возможно, являются фоновыми. Медианная фильтрация, выполняемая на третьем шаге, применяется с целью сделать сигнал более гладким, в том числе убрать короткие всплески яркости. Далее (шаг 4) сигнал на всём протяжении (первый и последний пиксельные столбцы исключаются, так как они могут быть частью тёмной каймы шириной в 1 пиксель, обрамляющей кадр по периметру) просматривается с помощью окна шириной  $w_{max}$  пикселей. На каждой итерации в пределах окна вычисляется максимум яркости и его абсцисса. Если разница между найденным максимумом

и наибольшим из значений яркости на границах окна превышает некоторый предел (условие 4.(с)), то после уточнения границ и проверки перекрытия (шаги 4.(с).i–iii) к результирующему множеству  $M$  добавляется новая пометка, обнаруженная в данном окне.

Заметим, что не все пометки, возвращаемые данной процедурой, попадают в просвет или располагаются близко к нему; это происходит потому, что указанное выше предположение оказывается верным не только для просветов, но и для некоторых элементов вагонов, например, для светлых вертикальных полос, расположенных между нефтяными потёками.

## 9.5 Регистрация новой пометки

После того как на очередном кадре были выделены пометки, необходимо для каждой из них определить, к какой серии она относится (то есть выяснить, существует ли серия  $s \in S_a$ , пометки которой были обнаружены на ранее обработанных кадрах и соответствуют тому же элементу вагона, что и данная пометка) и, если пометка относится к новой серии, установить родительские отношения между этой серией и уже известными сериями из множества  $S_a$ . Вспомогательные процедуры и функции, необходимые для решения данной задачи, описаны в разд. 9.5.1, а само решение приведено в разд. 9.5.2.

### 9.5.1 Вспомогательные процедуры и функции

#### 9.5.1.1 Вычисление смещения между элементами вагонов

Следующая функция вычисляет смещение между элементами вагонов, которым соответствуют пометки  $m_i$  и  $m_j$ ; для определённости будем полагать, что  $ind(f(m_i)) \leq ind(f(m_j))$ . Вычисление производится без учёта перспективного искажения.

$$dist(m_i, m_j) = d(m_j, m_i) - \sum_{i_f < k_f \leq j_f} v(k_f),$$

где  $i_f = ind(f(m_i))$ ,  $j_f = ind(f(m_j))$ ,

$$v(k_f) = \begin{cases} v_{avg}(k_f), & \text{если } v_{avg}(k_f) \neq nil; \\ v_c, & \text{иначе.} \end{cases}$$

Приведённая функция обладает следующим свойством, непосредственно вытекающим из определения: если элемент вагона, которому соответствует пометка  $m_j$ , расположен правее элемента вагона, которому соответствует пометка  $m_i$ , то выражение  $dist(m_i, m_j)$  положительно; иначе указанное выражение принимает отрицательное значение.

Определим данную функцию и для серий:  $dist(s_i, s_j) = dist(s_i(1), s_j(1))$ , где  $s_i, s_j$  — серии.

### 9.5.1.2 Добавление новой серии к множеству $S_a$

Серии в множестве  $S_a$  упорядочены следующим образом: серия  $s_i$  предшествует серии  $s_j$  тогда и только тогда, когда элемент вагона, которому соответствует серия  $s_i$ , появился в области видимости раньше, чем элемент вагона, которому соответствует серия  $s_j$ . Обратим внимание на то, что выражение  $-\text{sign}(v_c) \cdot \text{dist}(s_i, s_j)$ , используемое в процедуре добавления, ведёт себя в соответствии с указанным правилом: оно положительно, если элемент вагона серии  $s_i$  появился раньше элемента вагона серии  $s_j$ , и отрицательно в противном случае.

Следующая процедура добавляет новую серию  $s_n$  в множество  $S_a$  так, чтобы указанный порядок не был нарушен. Приведём её описание.

Процедура *ADD-NEW-SERIES*( $s_n$ ).

Входные данные: новая серия  $s_n$ .

В ходе выполнения данной процедуры изменяется множество  $S_a$ .

1. Если  $S_a = \emptyset$ , то добавить серию  $s_n$  в  $S_a$  и завершить процедуру.
2. Если  $\exists s \in S_a : -\text{sign}(v_c) \cdot \text{dist}(s, s_n) > 0$ , то
  - пусть  $p = \max\{1 \leq k \leq |S_a| : -\text{sign}(v_c) \cdot \text{dist}(S_a(k), s_n) > 0\}$ ;
  - вставить серию  $s_n$  в множество  $S_a$  после серии  $S_a(p)$  и перед серией  $S_a(p+1)$  (если такая имеется).
3. Иначе вставить серию  $s_n$  в множество  $S_a$  перед серией  $S_a(1)$ .

### 9.5.1.3 Поддержание множества последних серий $S_l$

Следующая процедура предназначена для поддержания множества последних серий  $S_l$  в актуальном состоянии. Серии внутри данного множества упорядочены в соответствии с тем же правилом, что и в множестве  $S_a$  (см. разд. 9.5.1.2).

Приведём теперь саму процедуру.

Процедура *UPDATE-LAST-SERIES*( $s_n$ ).

Входные данные: новая серия  $s_n$ .

В ходе выполнения данной процедуры изменяется множество  $S_l$ .

1. Вставить серию  $s_n$  в множество  $S_l$  таким образом, чтобы не был нарушен порядок данного множества.
2. Удалить из множества  $S_l$  все такие серии  $s$ , для которых выполняется неравенство
 
$$-\text{sign}(v_c) \cdot \text{dist}(s, s_n) > W_a + L_{t\_max}.$$

Детали выполнения первого шага можно увидеть в описании процедуры *ADD-NEW-SERIES* (см. разд. 9.5.1.2). Константа  $L_{t\_max} = 1200$  обозначает максимально возможную длину вагона.

#### 9.5.1.4 Построение накопительного фонового кадра

В данном разделе будет описана процедура добавления некоторой области текущего кадра к накопительному фоновому кадру.

Процедура *ADD-FIELD* ( $f_c, x_l, x_r$ ).

Входные данные: текущий неподготовленный кадр  $f_c = b(x, y)$ , левая граница добавляемой области  $x_l$ , правая граница добавляемой области  $x_r$ .

В ходе выполнения данной процедуры могут измениться значения переменных  $b_b$  и  $c_{b\_b}$ .

1.  $c_{b\_b}(x) := c_{b\_b}(x) + 1$  для  $x_l \leq x \leq x_r$ .
2.  $b_b(x, y) := b_b(x, y) + b(x + W_l - 1, y)$  для  $x_l \leq x \leq x_r, 1 \leq y \leq H$ .

#### 9.5.2 Процедура регистрации новой пометки

В этом разделе будет описана процедура учёта очередной пометки.

Процедура *ADD-MARK* ( $f_c, ind(f_c), x_l, x_r$ ).

Входные данные: новая пометка  $m = (ind(f_c), x_l, x_r)$ , текущий неподготовленный кадр  $f_c$ .

В ходе выполнения данной процедуры могут измениться значения переменных  $b_b, c_{b\_b}, S_a, S_l, F$ .

1. Если  $l(m) \geq l_{min}$ , то выполнить *ADD-FIELD* ( $f_c, x(m) - \Delta l, x(m) + \Delta l$ ).
2.  $S_s := \{ s \in S_l : |dist(s(1), m)| \leq d_{s\_max} \}$ .
3. Если  $S_s \neq \emptyset$ , то
  - $s^* := \arg \min_{s \in S_s} |dist(s(1), m)|$ ;
  - $s^* := s^* \cup \{m\}$ ;
  - завершить выполнение процедуры.
4.  $s_n := \{m\}$ .
5.  $P(s_n) := \{ s \in S_l : L_{t\_min} \leq -\text{sign}(v_c) \cdot dist(s, s_n) \leq L_{t\_max} \}$ .
6. *ADD-NEW-SERIES* ( $s_n$ ).
7. *UPDATE-LAST-SERIES* ( $s_n$ ).

Прокомментируем работу приведённой процедуры. На первом шаге часть области пометки, если её ширина достаточно велика, добавляется к накопительному фоновому кадру. На втором шаге выполняется поиск тех серий, к которым может принадлежать данная пометка (т.е. серий, соответствующих тому же элементу вагона, что и данная пометка). Если было найдено несколько таких серий, то среди них выбирается наиболее *близкая*, пометка добавляется к ней и процедура завершается (шаг 3). Если не удалось найти ни одной подходящей серии, то пометка

добавляется в новую серию (шаг 4). Далее для этой серии определяются родительские серии (шаг 5). Процедуры *ADD-NEW-SERIES* и *UPDATE-LAST-SERIES* на шестом и седьмом шагах вызываются для того, чтобы поддерживать множества  $S_a$  и  $S_l$  соответственно в актуальном состоянии.

Приведём теперь назначения и величины используемых констант. Константа  $d_{s\_max} = 35$  обозначает половину максимально возможного смещения между элементами вагонов, которым соответствуют пометки одной и той же серии. Константы  $L_{t\_min} = 300$  и  $L_{t\_max} = 1200$  обозначают минимальную и максимальную длины вагонов соответственно. Константа  $l_{min} = 20$  определяет, насколько широкой должна быть пометка, чтобы некоторая часть её изображения (величину которой определяет константа  $\Delta l = 5$ ) была добавлена к накопительному фоновому кадру.

## 10 Обработка накопленных данных на заключительном этапе

Задача, решаемая на данном этапе, состоит в построении некоторого основного течения, которое будет определять искомое разбиение железнодорожного состава на отдельные вагоны, и последующем определении координаты отделяющей вертикальной линии для всех кадров, на которых можно видеть край какого-либо вагона, и только для них (см. разд. 1). Эта задача решается нами следующим образом. Каждой серии сопоставляется вес так, что серии с бóльшим весом чаще соответствуют просветам, а серии с меньшим — иным элементам вагона. Для определения веса некоторой серии используются различные эмпирические правила (эвристики), подробно порядок его вычисления описан в разд. 10.3. Каждому основному течению также может быть сопоставлен вес, величина которого тем меньше, чем в большей степени это течение удовлетворяет двум требованиям: во-первых, оно должно содержать серии с большим весом и не содержать серий с малым весом; во-вторых, длины вагонов, определяемые данным течением, должны быть примерно одинаковыми. Однако нами рассматриваются не все возможные основные течения (формально порядок вычисления веса основных течений описан в разделах 10.2 и 10.1). Итак, основное предположение состоит в том, что основное течение, определяющее верное разбиение состава на вагоны, обладает наименьшим весом среди всех рассмотренных нами основных течений, заканчивающихся последней серией из множества  $S_a$ . После того, как искомое основное течение будет найдено, оно используется для определения координат разделяющей вертикали.

Процедура, решающая поставленную задачу, описана в разд. 10.1.

### 10.1 Основная процедура заключительного этапа

В данном разделе будет описана процедура, выполняемая на заключительном этапе обработки данных. Используемые в ней функция  $w_p(s_p, s)$  и вспомогательная процедура *DEFINE-POS* будут описаны в последующих подразделах.

Процедура *FINAL-PROCESS*.

Входные данные: переменные  $S_a, F$ .

Выходные данные: множество пар вида  $(ind(f), x)$ , где величина  $x$  определяет координату разделяющей вертикали на кадре с номером  $ind(f)$ .

1. Пусть

$$T^*(s) = \begin{cases} T^*(p^*(s)) \cup \{s\}, & \text{если } p^*(s) \neq nil; \\ \{s\}, & \text{иначе.} \end{cases}$$

2. Для каждой серии  $s \in S_a$ , начиная с первой и заканчивая последней, выполнить следующие шаги.

(a) Если  $P(s) \neq \emptyset$ , то  
 $p^*(s) := \arg \min_{s_p \in P(s)} w_p(s_p, s)$ .

(b) Если  $P(s) = \emptyset$ , то  
 $p^*(s) := nil$ .

(c)  $w(T^*(s)) := w_p(p^*(s), s)$ .

3.  $L_{avg} := \frac{1}{|T_d|-1} \sum_{k=1}^{|T_d|-1} |dist(T_d(k), T_d(k+1))|$ , где  $T_d = T^*(s_e(S_a))$ .

4. Повторить шаг 2.

5.  $R := DEFINE-POS(T^*(s_e(S_a)))$ .

6. Возвратить множество  $R$  в качестве результата.

Прокомментируем работу приведённой процедуры. Первый шаг служит для того, чтобы показать принцип построения искомого основного течения, и непосредственно не выполняется. На втором шаге для каждой серии  $s \in S_a$  определяется серия  $p^*(s)$ , в некотором отношении более подходящая на роль родителя, чем другие серии; кроме того, определяется вес течения, построенного так, как было указано на первом шаге, и ведомого серией  $s$ . На третьем шаге оценивается средняя длина вагона в составе (на втором шаге эта величина ещё не была известна). На четвёртом шаге определение искомого основного течения выполняется заново с использованием более точной оценки средней длины вагона, после чего на пятом шаге найденная последовательность серий используется для определения координаты разделяющей вертикали для тех кадров, на которых можно видеть край какого-либо вагона.

## 10.2 Определение условного веса течения

В этом разделе описана функция  $w_p(s_p, s)$ , определяющая вес течения  $T^*(s_p) \cup \{s\}$ , где  $s_p \in P(s)$ , через вес течения  $T^*(s_p)$ . Приведём её общий вид.

$$w_p(s_p, s) = \begin{cases} w_{penalty}(s_p, s), & \text{если } s_p = nil; \\ w_{penalty}(s_p, s) + w_{shift}(s_p, s) + w(T^*(s_p)), & \text{если } s_p \neq nil. \end{cases}$$

Прежде чем привести описания используемых вспомогательных функций и комментариев к ним, сделаем замечание. Обратим внимание на то, что всякое основное

течение соответствует всему железнодорожному составу или его части: первая серия предположительно соответствует тому концу состава, который первым попадает в область видимости видеокамеры, последняя серия может соответствовать другому концу состава, просвету между вагонами или некоторому элементу очередного вагона. Основное течение задаёт разбиение состава (или его части) на вагоны следующим образом: всякий элемент вагона, для которого в данном течении существует серия, ему соответствующая, считается просветом, а *промежутки* между этими элементами — вагонами. Разумеется, это разбиение может быть и неверным: например, одним вагоном могут считаться части двух соседних (настоящих). Далее под *вагонами некоторого основного течения* будем понимать последовательность (потенциальных) вагонов, на которые данное основное течение разбивает часть состава, которой оно соответствует, указанным выше образом.

Перейдём теперь к описанию используемых вспомогательных функций.

1.  $w_{penalty}(s_p, s) = \sum_{s_i \in S_i} (w(s_i) - k_{w(s)} w_{avg})$ , где  $w_{avg} = \frac{1}{|S_a|} \cdot \sum_{s \in S_a} w(s)$ ,  $S_i \subseteq S_a$  — это множество серий, такое что для каждой  $s_i \in S_i$  верны следующие неравенства:

- $-\text{sign}(v_c) \cdot \text{dist}(s_p, s_i) > 0$ ;
- $-\text{sign}(v_c) \cdot \text{dist}(s_i, s) > 0$ .

Если  $s_p = nil$ , то первое из двух указанных выше неравенств опускается. Используемое значение константы  $k_{w(s)}$  равно 0,25.

Таким образом, данная функция вычисляет сумму весов (уменьшенных на некоторую часть от среднего веса) всех серий, расположенных *между* сериями  $s_p$  и  $s$ . Благодаря наличию компоненты  $w_{penalty}$  вес основного течения становится тем больше, чем меньше серий с большим весом (большим  $k_{w(s)} w_{avg}$ ) из множества  $S_a$  оно содержит (и наоборот для серий с малым весом).

2.

$$lcl(s, \Delta g) = \begin{cases} \Delta g \cdot \widehat{L}_{avg}, & \text{если } P(s) = \emptyset; \\ |\text{dist}(m_e(p^*(s)), m_1(s))|, & \text{если } P(s) \neq \emptyset \text{ и } \Delta g = 1; \\ lcl(s, 1) + lcl(p^*(s), \Delta g - 1), & \text{если } P(s) \neq \emptyset \text{ и } \Delta g > 1. \end{cases}$$

Функция  $lcl$  предназначена для определения суммарной длины  $\Delta g$  последних вагонов основного течения  $T^*(s)$ . Длины отсутствующих вагонов (если данное течение слишком коротко) заменяются на среднюю длину вагона по умолчанию, равную  $\widehat{L}_{avg} = 550$ .

3.  $w_{shift}(s_p, s) = k_h \cdot |\text{dist}(m_e(s_p), m_1(s)) + \text{sign}(v_c) l_{avg}|$ , где

$$l_{avg} = \begin{cases} L_{avg}, & \text{если } L_{avg} \neq nil; \\ (c_{lcl})^{-1} \cdot lcl(s_p, c_{lcl}), & \text{если } L_{avg} = nil. \end{cases}$$

Функция  $w_{shift}$  предназначена для того, чтобы учитывать различия длин вагонов основного течения  $T^*(s_p) \cup \{s\}$ . Выражение под знаком модуля — разница

между длиной вагона, заключённого между сериями  $s_p$  и  $s$ , и средней длиной (средней для всего состава или для последних  $c_{lcl} = 5$  вагонов основного течения  $T^*(s_p)$ ). Чем бóльшим будет различие длин вагонов, вычисленное указанным образом, тем бóльшим будет вес данного основного течения. Коэффициент  $k_h = 5,0$  необходим для того, чтобы регулировать соотношение вкладов функций  $w_{shift}$  и  $w_{penalty}$  в общий вес основного течения.

### 10.3 Определение веса серии

В данном разделе будет описана функция веса серии  $w(s)$ . Приведём вначале её общий вид.

$$w(s) = w_{len}(s) + w_{count}(s) + w_{sim}(s).$$

Теперь опишем отдельно каждую компоненту веса.

#### 1. Компонента $w_{len}$ .

$$w_{len}(s) = k_{len} \cdot \left( \frac{\max_{m \in s} l(m)}{\min_{m \in s} l(m)} - 1 \right).$$

Значение используемой константы  $k_{len}$  равно 100,0. Данная компонента предназначена для того, чтобы отличать светлые полосы на вагонах от просветов. Предположение состоит в том, что ширина области просвета в кадре изменяется в большей степени во время приближения этого просвета к видеокамере (или удаления от неё), чем ширина светлой полосы на поверхности вагона в аналогичных условиях; дело в том, что, чем дальше от видеокамеры располагается просвет, тем бóльшую его часть в кадре занимает торцевая поверхность одного из вагонов.

#### 2. Компонента $w_{count}$ .

$$w_{count}(s) = k_{count} |v_{s\_avg}| \cdot |s| f_g,$$

где

$$v_{s\_avg} = \frac{1}{|s|} \cdot \sum_{m \in s} v_{avg}(ind(f(m))).$$

Предположение, побудившее нас включить данную компоненту в общий вес, заключается в том, что серии, содержащие малое количество пометок, по сравнению с остальными сериями с большей вероятностью соответствуют светлым вертикальным полосам на поверхности вагонов, причём таким, яркость которых незначительно отличается от яркости окружающей поверхности. Если разница яркостей близка к пороговой (см. разд. 9.4), то такая полоса может обнаруживаться процедурой *FIND-CHINKS* не на всех кадрах, на которых она видна. В то же время просветы в большинстве случаев характеризуются тем,



что их яркость значительно отличается от яркости окружающей поверхности. Множители  $|v_{s\_avg}|$  и  $f_g$  добавлены для того, чтобы сделать вклад данной компоненты независимым от скорости движения состава в данный момент времени и величины межкадрового пропуска. Значение используемой константы  $k_{count}$  равно 12,5.

### 3. Компонента $w_{sim}$ .

Данная компонента необходима для отличия серий, соответствующих просветам, от иных серий путём сравнения изображений составляющих их пометок с накопительным фоновым кадром. Прежде чем привести формальное определение этой компоненты, сделаем некоторые замечания. В приведённых далее соотношениях переменные  $x$  и  $y$  будут лежать внутри области очередной пометки  $m$ , т.е.  $x_l(m) \leq x \leq x_r(m)$ ,  $1 \leq y \leq H$ . Матрица  $b(m, x, y)$  — это часть неподготовленного кадра, на котором была обнаружена пометка  $m$ , лежащая внутри области этой пометки; величина  $c_w(m)$  обозначает количество ненулевых элементов вектора  $c_{b\_b}$  в тех же пределах для  $x$ . Приведём теперь формальное описание компоненты  $w_{sim}$ .

$$w_{sim}(s) = \begin{cases} k_{sim}|s^+|^{-1} \cdot \sum_{m \in s^+} w(m), & \text{если } |s^+|^{-1} \cdot \sum_{m \in s^+} c_w(m) \geq c_{w\_min}; \\ 0, & \text{иначе,} \end{cases}$$

где  $s^+ = \{m \in s \mid c_w(m) > 0\}$ ;

$$w(m) = \begin{cases} \sqrt{\frac{\sum (q(m, x, y) - q_{avg}(m))^2}{c_w(m) \cdot H}}, & \text{если } c_w(m) > 0; \\ 0, & \text{иначе;} \end{cases}$$

$$q_{avg}(m) = \frac{\sum q(m, x, y)}{c_w(m) \cdot H};$$

$$q(m, x, y) = \begin{cases} \ln \frac{\max\{1, b(m, x, y)\}}{\max\{1, b_r(x, y)\}}, & \text{если } c_{b\_b}(x) > 0; \\ 0, & \text{иначе;} \end{cases}$$

$$b_r(x, y) = \begin{cases} \lfloor b_b(x, y) / c_{b\_b}(x) \rfloor, & \text{если } c_{b\_b}(x) > 0; \\ 0, & \text{иначе.} \end{cases}$$

Приведём значения используемых констант:  $k_{sim} = -3500,0$ ;  $c_{w\_min} = 20,0$ .

## 10.4 Определение координаты разделяющей вертикали

В данном разделе будет описана процедура определения абсциссы вертикали, отделяющей вагоны друг от друга или от фона, для каждого кадра фильма, на котором можно видеть край очередного вагона.

Процедура *DEFINE-POS* ( $T^*$ ).

Входные данные: основное течение  $T^*$ .

Выходные данные: множество пар  $R$  вида  $(ind(f), x)$ , где величина  $x$  определяет координату вертикали на кадре с номером  $ind(f)$ .

1.  $R := \emptyset$ .
2. Для каждой серии  $s \in T^*$  выполнить следующие шаги.
  - (a)  $x_c := x(m_1(s)), i_f := ind(f(m_1(s)))$ .
  - (b) Пока  $1 \leq \lfloor x_c \rfloor \leq W_a$  и  $i_f \geq 1$ , выполнять
    - $R := R \cup \{(i_f, \lfloor x_c \rfloor)\}$ ;
    - $x_c := x_c - v_{avg}(i_f)$ ;
    - $i_f := i_f - 1$ .
  - (c)  $m_l := m_1(s), x_c := x(m_1(s))$ .
  - (d) Для каждого  $2 \leq k \leq |s|$  выполнить следующие шаги.
    - i.  $m_c := s(k)$ .
    - ii. Для каждого  $ind(f(m_l)) + 1 \leq i_f < ind(f(m_c))$  выполнить
      - $x_c := x_c + v_{avg}(i_f)$ ;
      - $R := R \cup \{(i_f, \lfloor x_c \rfloor)\}$ .
    - iii.  $R := R \cup \{(ind(f(m_c)), x(m_c))\}$ .
    - iv.  $x_c := x(m_c)$ .
    - v.  $m_l := m_c$ .
  - (e)  $i_f := ind(f(m_e(s))) + 1$ .
  - (f) Если  $v_{avg}(i_f) \neq nil$ , то  $x_c := x_c + v_{avg}(i_f)$ , иначе  $x_c := x_c + v_c$ .
  - (g) Пока  $1 \leq \lfloor x_c \rfloor \leq W_a$ , выполнять
    - $R := R \cup \{(i_f, \lfloor x_c \rfloor)\}$ ;
    - $i_f := i_f + 1$ ;
    - если  $v_{avg}(i_f) \neq nil$ , то  $x_c := x_c + v_{avg}(i_f)$ , иначе  $x_c := x_c + v_c$ .
3. Вернуть множество  $R$  в качестве результата.

## 11 Описание проведённых экспериментов и результатов работы алгоритма

Предлагаемый алгоритм был протестирован с использованием 16 фильмов, один из которых был снят ночью, остальные — днём. Все указанные фильмы были получены с помощью видеокамеры, установленной на территории предприятия «Славнефть-ЯНОС» (г. Ярославль). В каждом фильме изображено движение одного состава, в среднем состоящего из 26,5 вагонов. Среднее количество кадров в фильмах — 1960.

Приблизительное количество кадров и количество вагонов в составе для каждого фильма указаны в табл. 1. Заметим, что при обработке фильма №16, снятого ночью, между первым и вторым шагами процедуры *PREPARE-PIC* выполнялась инверсия изображения кадра, т.е. яркость каждого пикселя заменялась на противоположную величину.

Таблица 1. Характеристики фильмов

#	Кадров	Вагонов	#	Кадров	Вагонов
1	1150	34	9	1150	18
2	2200	34	10	2300	32
3	940	10	11	2310	34
4	2680	28	12	2320	34
5	1830	21	13	3540	35
6	2340	33	14	2830	34
7	960	10	15	1575	21
8	1140	18	16	2200	33

В ходе тестирования алгоритм не пропустил ни одного края и не сообщил о лишнем крае (т.е. о таком, которого не было в действительности), однако расположение вертикальной линии, отделяющей вагоны, на части кадров было определено неточно. Один из возможных путей улучшения алгоритма — более точная оценка мгновенной скорости состава.

## Список литературы

1. Прэтт У. Цифровая обработка изображений: Пер. с англ. под ред. Д. С. Лебедева: в 2 кн. Кн. 2. М.: Мир, 1982. 480 с.: ил.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений: Пер. с англ. под ред. П. А. Чочиа. М.: Техносфера, 2005. 1072 с.
3. Волотовский С. Г., Казанский Н. Л., Попов С. Б., Хмелев Р. В. Система технического зрения для распознавания номеров железнодорожных цистерн с использованием модифицированного коррелятора в метрике Хаусдорфа // Компьютерная оптика. 2005. В. 27. С. 177 – 184.
4. Карлин А. К., Малков А. Н., Тимофеев Е. А., Штерн Г. П. Распознавание номеров железнодорожных цистерн с использованием корреляционного алгоритма // Математика, кибернетика, информатика: Труды международной научной конференции, посвящённой памяти профессора А. Ю. Левина (Ярославль, 25 – 26 июня, 2008) / Под ред. С. А. Кащенко, В. А. Соколова; Яросл. гос. ун-т. Ярославль: ЯрГУ, 2008. С. 103 – 110.

## The Decomposition of a Train into its Cars

Malkov A.N., Shtern G.P., Mikhaylov I.A.

**Keywords:** automatic train registration, train decomposition into cars, heuristic algorithm

The problem of the decomposition of a train into its cars is considered in this paper. This problem arises while making automatic systems of train registration. The input data is a film that visualizes the train motion. An algorithm for solving the problem is proposed. An experiment is conducted using 16 different films to test the algorithm.

### **Сведения об авторах:**

**Малков Алексей Николаевич,**

Ярославский государственный университет им. П.Г. Демидова,  
доцент;

**Штерн Геннадий Павлович,**

Ярославский государственный университет им. П.Г. Демидова,  
доцент;

**Михайлов Иван Александрович,**

Ярославский государственный университет им. П.Г. Демидова,  
аспирант.