

УДК 004.932

П.В. Бабаян, В.В. Стротов

РЕАЛИЗАЦИЯ АЛГОРИТМА СЛЕЖЕНИЯ ЗА МНОЖЕСТВОМ ОБЪЕКТОВ НА ОСНОВЕ РАЗБИЕНИЯ ДВУДОЛЬНОГО ГРАФА В МАЛОГАБАРИТНЫХ БОРТОВЫХ СИСТЕМАХ ВИДЕОСЛЕЖЕНИЯ

Предложена реализация алгоритма слежения за множеством движущихся объектов в системе видеослежения на основе ПЛИС Xilinx с реализованным на ней софт-процессорным блоком MicroBlaze. Данная реализация предполагает отказ от рекурсивных процедур в пользу процедур на основе таблиц. Экспериментальные исследования показывают увеличение производительности в 5 – 9 раз по сравнению с оригинальной реализацией.

Ключевые слова: бортовые системы видеослежения, ПЛИС, слежение за несколькими объектами, двудольный граф.

Введение. Современные бортовые системы наземных, воздушных и космических аппаратов решают большое количество задач, таких как автономная навигация, предотвращение столкновения с препятствиями, обнаружение, выделение объектов и слежение за ними. Структура аппаратной части такой системы зависит от типа носителя, круга решаемых задач и условий эксплуатации. Наиболее гибкими и высокопроизводительными могут считаться системы, содержащие одновременно процессоры разных типов и несколько ПЛИС в качестве основных вычислительных узлов. Программируемая логика используется для проведения «тяжелых» вычислений, таких как фильтрация изображений, их преобразование, поиск объекта по эталонному изображению, пороговая бинаризация, разметка изображений и решение СЛАУ. Процессор в подобных системах используется для проведения специальных вычислений и как контрольный узел [1].

В том случае, когда система проектируется для малогабаритного носителя, на такие её параметры, как размеры, вес, энергопотребление и тепловыделение, накладываются серьёзные ограничения. Для эксплуатации в особо тяжелых условиях, таких как, например, большие высоты, учитываются такие параметры, как диапазон рабочих температур и радиационная стойкость. В такой ситуации разработчики сильно ограничены в выборе вычислительных узлов. Фактически заданным условиям удовлетворяют считанные микросхемы типа «система-на-чипе». Если пойти по другому пути и отказаться от связки процессор+ПЛИС, то появляется возможность ис-

пользования единственной программируемой схемы, на основе которой создается так называемый софт-процессорный блок, или программно реализуемый процессор. Примером могут служить блоки Xilinx MicroBlaze и Altera Nios II. Данные процессоры обладают достойными характеристиками, например, MicroBlaze – это полноценный 32 разрядный RISC процессор с гарвардской архитектурой. Он имеет пятиуровневый конвейер и выделенные области памяти для данных и команд.

Достоинствами предложенной схемы являются простота и гибкость её реализации, так как софт-процессор может быть настроен сообразно решаемым задачам. Недостаток этой схемы логично вытекает из её достоинств. Это более низкая производительность по сравнению с полноценными, аппаратно реализованными процессорами. Фактически максимальная частота такого процессора может быть порядка 330 МГц (при условии использования ПЛИС Xilinx серии Virtex7), а максимальный объем кэш-памяти ограничен 64 кБ. Это означает, что некоторые алгоритмы, чувствительные к характеристикам процессора (например, рекурсивные алгоритмы или алгоритмы с большим количеством ветвлений), не смогут быть эффективно реализованы.

В таблице приведена примерная классификация алгоритмов обработки изображений по принципу пригодности к реализации в системах различного вида. Алгоритмы сгруппированы относительно решаемой ими задачи.

Одновременное слежение за множеством объектов широко используются в различных прикладных областях, таких как навигация лета-

тельных аппаратов, мониторинг дорожного движения, анализ потоков людей в толпе и т.д.

Классификация алгоритмов обработки изображений по принципу пригодности к реализации

Решаемая задача	Алгоритмы, пригодные для систем на процессорах	Алгоритмы, пригодные для систем на ПЛИС
Сопоставление изображений	Сопоставление особенностей	Сопоставление участков изображений
Фильтрация изображений	Адаптивная фильтрация	Фильтрация на основе свертки, морфологическая фильтрация
Сегментация изображений	Сегментация на основе активных контуров	Пороговая сегментация, статистическая сегментация
Слежение за объектами	Слежение на основе графов	Слежение на основе матриц
Распознавание объектов	На основе деревьев	Нейронные сети, нечеткие классификаторы

Задача отслеживания нескольких объектов определяется здесь как одновременная оценка траектории движения нескольких объектов в последовательности изображений видеокadres [2]. Основные трудности в решении данной задачи (по сравнению с задачей отслеживания одного объекта) связаны с возможными исчезновениями, появлениями и заслонениями объектов. Кроме того, объекты могут быть временно перекрыты элементами сцены, становясь при этом невидимыми для наблюдателя.

Существует много подходов к решению задачи слежения за множеством объектов, в частности подходы на основе теории графов или на основе матричного представления перемещения объектов [2-5]. Рассматриваемая задача в значительной мере близка к хорошо известной задаче связывания данных [6].

В представленной работе для слежения за объектами предлагается использовать алгоритм, основанный на биении двудольного графа. На вход данный алгоритм получает список параметров бинарных сегментов, найденных алгоритмом выделения объектов. На выходе алгоритма слежения – список параметров объектов, обязательно включающий их координаты объектов и, опционально, размер, средняя яркость и т.д. Каждый элемент списка обозначен уникальным номером.

Целью работы является создание реализации алгоритма слежения за несколькими объектами, основанного на разбиении двудольного

графа, для малогабаритных систем видеослежения, основным вычислительным элементом в которых является ПЛИС.

Математические модели. Рассмотрим модель наблюдаемого изображения в виде (1):

$$l(i, j, n) = r(i, j, n) \cdot h(i, j, n) + (1 - r(i, j, n)) \cdot g(i, j, n) + \xi(i, j, n), \quad (1)$$

где i, j – пространственные координаты, $n = \overline{1, N}$ – номер текущего кадра, $l(i, j, n)$ – наблюдаемое изображение, $g(i, j, n)$ и $h(i, j, n)$ – неизвестные значения яркости фона и объектов соответственно, $\xi(i, j, n)$ – белый гауссов шум, $r(i, j, n)$ – бинарная маска движущихся объектов. Изображение фона изменяется с течением времени. Его изменения связаны с движением камеры или её носителя. Упрощенно преобразование фона можно описать как совокупность Евклидовых преобразований (сдвиг и поворот) и однородного преобразования масштаба. Они описываются выражением (2):

$$g(i, j, n) = g_0(T_g(i, j, n)), \quad (2)$$

$$T_g(i, j, n) = (i \cos \psi(n) + j \sin \psi(n) + v_x(n), -i \sin \psi(n) + j \cos \psi(n) + v_y(n)),$$

где $g_0(i, j)$ – исходное изображение фона, $v_x(n), v_y(n)$ – параметры смещения по горизонтали и вертикали соответственно, $\psi(n)$ – угол поворота, $\lambda(n)$ – коэффициент изменения масштаба в n -м кадре.

Пусть K – общее число объектов, наблюдаемых в сцене, тогда маска $r(i, j, n)$ может быть представлена как:

$$r(i, j, n) = \max_{k=1, K} r_k(i, j, n), \quad (3)$$

где $r_k(i, j, n)$ – бинарная маска k -го объекта. Если объект невидим на кадре n , то $r_k(i, j, n) = 0, \forall i, j$.

Постановка задачи. Результатом работы алгоритмов выделения объектов, а также разметки и параметризации является список, включающий координаты и размеры найденных сегментов [2,7,8]. На кадре n список содержит $M(n)$ измерений. Каждое из них соответствует связному сегменту бинарного изображения. Возможны три ситуации, каждая из которых определяет собственную модель измерений.

1. Некоторый m -й сегмент соответствует k -му объекту, при этом модель будет иметь вид:

$$\begin{aligned} \tilde{x}_m(n) &= x_k(n) + \xi_m^x(n), \quad \tilde{y}_m(n) = y_k(n) + \xi_m^y(n), \\ \tilde{w}_m(n) &= w_k(n) + \xi_m^w(n), \quad \tilde{h}_m(n) = h_k(n) + \xi_m^h(n), \end{aligned} \quad (4)$$

где $x_m(n), y_m(n)$ – координаты центра объекта на кадре n ; $w_k(n), h_k(n)$ – его размеры, $\tilde{x}_m(n), \tilde{y}_m(n), \tilde{w}_m(n), \tilde{h}_m(n)$ – результат измерения соответствующих величин, $\xi_m^x, \xi_m^y \sim N(0, \sigma_y^2)$, $E(\xi_m^x, \xi_m^y) = 0$ – белый гауссов шум.

2. Некоторый m -й сегмент является результатом объединения объектов с номерами k_1 и k_2 . Из предположения, что оба они имеют прямоугольную форму, получаем следующую модель:

$$\begin{aligned} \tilde{x}_m(n) &= \frac{x_m^{\min}(n) + x_m^{\max}(n)}{2} + \xi_m^x(n), \\ \tilde{y}_m(n) &= \frac{y_m^{\min}(n) + y_m^{\max}(n)}{2} + \xi_m^y(n), \\ \tilde{w}_m(n) &= x_m^{\max}(n) - x_m^{\min}(n) + \xi_m^w(n), \\ \tilde{h}_m(n) &= y_m^{\max}(n) - y_m^{\min}(n) + \xi_m^h(n), \end{aligned} \quad (5)$$

где

$$\begin{aligned} x_m^{\min}(n) &= \min(x_{k_1}(n) - w_{k_1}(n)/2, x_{k_2}(n) - w_{k_2}(n)/2), \\ x_m^{\max}(n) &= \max(x_{k_1}(n) + w_{k_1}(n)/2, x_{k_2}(n) + w_{k_2}(n)/2). \end{aligned}$$

Переменные $y_m^{\min}(n)$ and $y_m^{\max}(n)$ определяются схожим образом.

3. Некоторый m -й сегмент является результатом ложного выделения. В этом случае $\tilde{x}_m(n), \tilde{y}_m(n)$ являются равномерно распределенными случайными величинами.

Исключим из рассмотрения ситуации, когда три и более объекта объединяются в один сегмент или один объект разделяется на три и более сегмента.

Задача слежения за объектами заключается в оценивании величин $x_k(n), y_k(n), k = \overline{1, K}$.

Алгоритм слежения за объектами. Пусть Q – множество связных областей, найденных на кадре n , S – множество объектов, известных на текущем кадре. Очевидно, на первом кадре $S = \emptyset$. Каждая область $q_m \in Q$ определяется вектором измерений $(\tilde{x}_m(n), \tilde{y}_m(n), \tilde{w}_m(n), \tilde{h}_m(n))$. Наиболее важная задача рассматриваемого алгоритма – определение соответствия между сегментами и объектами. Соответствие между сегментами $q_k \in Q$ и объектами $s_m \in S$ может быть описано двудольным графом G (рисунок 1). Присутствие дуги между вершинами s_m и q_k графа означает, что сегмент q_k соответствует объекту s_m , вес дуги представляет собой меру соответствия между сегментом и объектом. Этот граф будем называть траекторным.

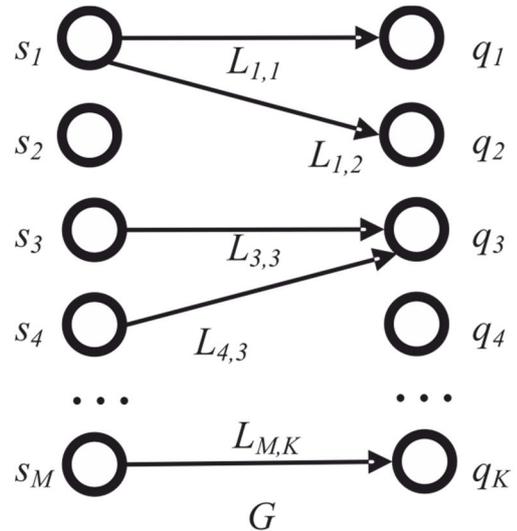


Рисунок 1 – Пример траекторного графа

Для получения информации о слиянии, разделении, появлении и исчезновении объектов граф необходимо разбить на связные компоненты. Пусть $\Gamma = \{G_i\}$ – множество этих компонент. На рисунке 2 показан пример разбиения графа G на связные компоненты для случая $M = 5$ и $K = 5$. Компонента G_1 соответствует разделению объекта, G_2 – слиянию объектов, G_3 – перемещению объекта, G_4 – пропаданию объекта, G_5 – появлению объекта. Эти компоненты будем называть “элементарными”.

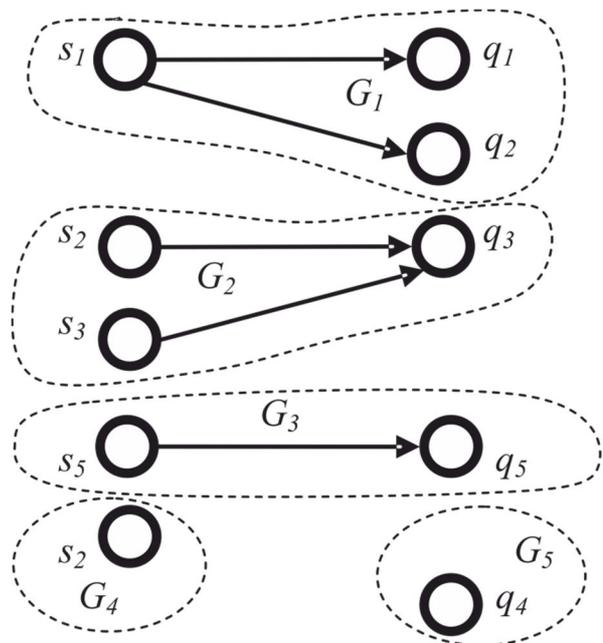


Рисунок 2 – Разделение двудольного графа на связные компоненты

В качестве меры соответствия используется функция:

$$L_{m,k} = 1 / d(m, k), \quad (6)$$

где $d(m, k)$ – расстояние s_m между q_k :

$$d(m, k) = \min_{c_x, c_y} \left[\sqrt{(\bar{x}_k(n-1) - c_x)^2 + (\bar{y}_k(n-1) - c_y)^2} \right], \quad (7)$$

где $c_x : \left[|\tilde{x}_m(n) - c_x| \leq \frac{\tilde{w}_m(n)}{2} \right], c_y : \left[|\tilde{y}_m(n) - c_y| \leq \frac{\tilde{h}_m(n)}{2} \right]$.

Каждый элемент множества Q описывается следующими параметрами: уникальный идентификационный номер объекта $Id_k(n)$; оценка положения центра объекта $\hat{x}_k(n), \hat{y}_k(n)$ и скорости его движения $\hat{\dot{x}}_k(n), \hat{\dot{y}}_k(n)$; оценки геометрических размеров объекта $\hat{w}_k(n), \hat{h}_k(n)$ и скорость их изменения $\hat{\dot{w}}_k(n), \hat{\dot{h}}_k(n)$; время (количество кадров) наблюдения t_k и время нахождения в прогнозе t_k^{pr} ; если элемент множества является результатом слияния объектов, то сюда же включается вся информация о траектории, размерах и временах для слитых объектов. Также присутствует счетчик нахождения в состоянии слияния t_k^m .

Алгоритм слежения за множеством объектов можно разделить на несколько шагов.

1. Построение полного графа. Он содержит все дуги, где мера соответствия превышает указанный порог L_{min} . Пример такого графа представлен на рисунке 3,а. Это не траекторный граф. Он состоит из двух связных компонент G_1 и G_2 .

2. Разбиение графа на связные компоненты.

3. Обработка каждой связной компоненты в зависимости от её типа. Для каждой не элементарной компоненты дуга с минимальным весом удаляется. Рекурсивная процедура повторяется для каждого полученного подграфа согласно шагам 2 и 3. Например, «наихудшая» дуга в G_1 – это дуга (s_1, q_2) с весом $L_{1,2} = 0,3$ (пунктирная линия на рисунке 3,а). После её удаления компонента G_1 разбивается на элементарные компоненты G_3 и G_4 (рисунок 3,б).

После разбиения двудольного графа получается список элементарных компонент. На основе этого списка выполняется оценивание координат объектов, при этом можно выделить пять различных случаев.

3.1 «Обычное» перемещение объекта.

При этом параметры объекта определяются с использованием фильтра Калмана. Счетчик t_k увеличивается на единицу, t_k^{pr} устанавливается в ноль. Если объект является результатом слияния, счетчик t_k^m также увеличивается на единицу. По

истечении заранее определенного времени информация о слиянии удаляется из описания объекта.

3.2 Возникновение объекта.

В этом случае создается новая запись, которой присваивается новый идентификационный номер, в множество S включается новый элемент.

3.3 Пропадание объекта.

При этом положение и размеры объекта прогнозируются с использованием Калмановской фильтрации. Счетчик t_k^{pr} увеличивается на единицу. Прогнозирование прекращается по истечении заданного времени.

3.4 Слияние объектов.

В этом случае в множество S включается новый элемент. Положение и размер сегмента становятся начальными параметрами нового объекта. Объекты, подвергшиеся слиянию, удаляются из S , однако информация об их траекториях сохраняется в описании нового объекта в течение определенного времени. Счетчик t_k^m стартует с единицы.

3.5 Разделение объектов.

Если информация о слиянии объектов отсутствует, то тот из сегментов, который находится ближе к месту разделения, становится носителем исходного идентификационного номера. Параметры второго определяются согласно п. 3.2. В случае, если информация о разделении присутствует, то прогнозируется положение слитых сегментов, а затем соответствие объектов и сегментов определяется исходя из анализа расстояний. Параметры объектов определяются согласно п. 3.1

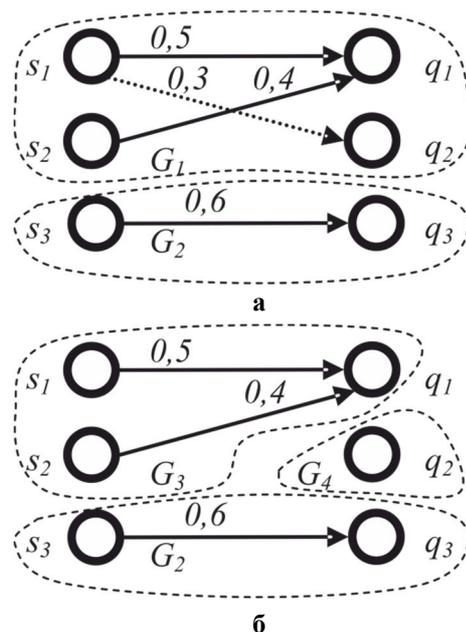


Рисунок 3 – Пример обработки графа: а – полный граф; б – траекторный граф

4. Выделение информации обо всех объектах из множества S . Главными параметрами считаются положение, скорость, времена наблюдения и слежения за объектом [9].

Реализация алгоритма в системе видеослежения на основе ПЛИС. Предложенный алгоритм в том виде, который представлен в предыдущем разделе, фактически нереализуем в системах видеослежения на основе ПЛИС. В частности, можно выделить следующие проблемы [10].

1. Процедура биения графа является рекурсивной, что означает, что она не может быть реализована в ПЛИС. Реализация данной процедуры на основе софт-процессорного ядра возможна, но на практике скорость работы чересчур низка. Данная ситуация вызвана малой скоро-

стью работы процедур, связанных с выделением памяти. Также неоправданно высок риск переполнения стека.

2. Процедуры генерации графа и определения элементарности компонент занимают около 60 и 35 процентов времени вычисления соответственно.

Описанные проблемы привели к созданию новой реализации алгоритма. Её схема приведена на рисунке 4. На данном рисунке тонкие черные линии обозначают простые (непосредственные) связи между блоками, а толстые белые линии – буферизированные линии передачи информации.

Реализация алгоритма может быть условно разделена на несколько шагов.

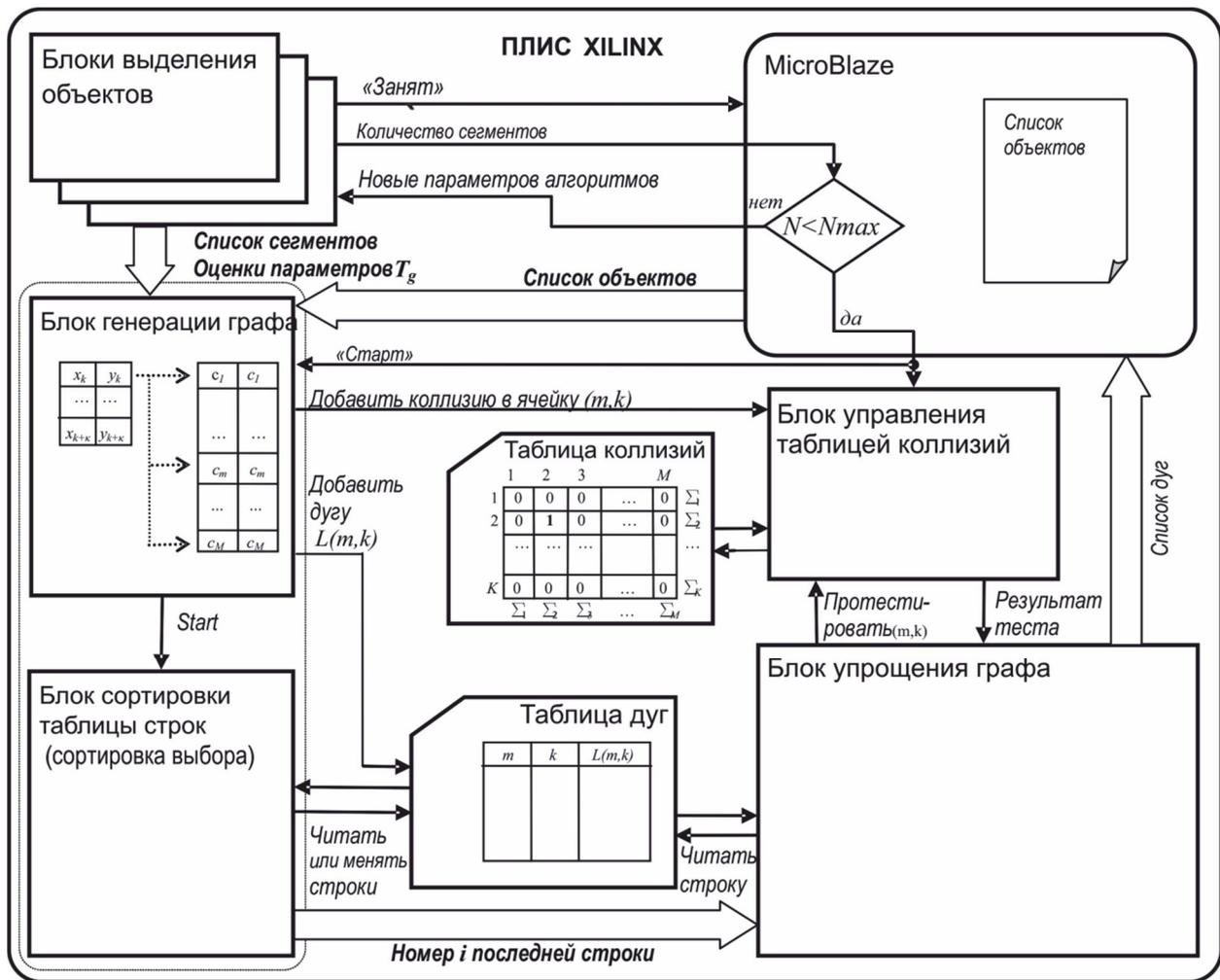


Рисунок 4 – Схема аппаратной реализации алгоритма

1. Подготовка данных.

Описываемый алгоритм стартует одновременно с алгоритмами выделения объектов и разметки. Для этого анализируется сигнал «Занят» от соответствующего блока. Пока производится обработка вновь принимаемого изображения, софт-процессорное ядро простаивает, а значит,

оно может быть задействовано для передачи актуального списка объектов в блок генерации графа.

2. Построение полного траекторного графа.

Блок генерации графа начинает получать информацию о найденных на текущем кадре сегментах. Полученные координаты «на лету»

переводятся в систему координат, связанную с первым изображением. Для этого используются оценки параметров преобразования фонового изображения $T_g(i, j, n)$. После этого вычисляются расстояния между каждым сегментом и каждым объектом. Если дистанция не превышает заданный порог, то данный блок создает новую дугу в таблице дуг. Кроме того, увеличиваются на единицу счетчики связей для сегмента P_k и объекта P_m . Если размер ПЛИС позволяет, то можно производить k сопоставлений сегментов и объектов одновременно.

После того как все расстояния измерены, блок генерации графа формирует так называемую таблицу коллизий. Она заполняется на основе таблицы дуг и счетчиков связей. В данной работе под термином «коллизия» понимается ситуация, когда у двух вершин, соединенных дугой, общее количество связей больше трех. Пример коллизии можно видеть на рисунке 3,а. Здесь суммарное число связей у вершин s_1 и q_1 равно четырем. Значение элемента в таблице коллизий определяется по правилу:

$$C(m, k) = \begin{cases} P_m + P_k - 3, & \text{if } (P_m + P_k > 3) \wedge (L(m, k) > 0); \\ 0, & \text{иначе} \end{cases} \quad (8)$$

3. Сортировка таблицы дуг.

На следующем этапе производится сортировка таблицы дуг методом выбора. Такой метод был выбран из-за того, что после каждого просмотра данный блок увеличивает количество отсортированных строк на единицу. Это позволяет объединить данный блок с блоком упрощения графа в единый конвейер.

4. Генерация траекторного графа.

Цель работы данного блока – определение всех лишних дуг в графе. Данный блок получает номер последней строки в отсортированной части таблицы дуг и генерирует запрос к таблице коллизий, включающий номера вершин (m, k) . Блок контроля таблицы коллизий проверяет суммы по m -му столбцу (Σ_m) и k -й строке (Σ_k). Если обе они равны нулю, то блок возвращает положительный результат. Тогда значение данной строки таблицы дуг передается в MicroBlaze. В противном случае блоком контроля коллизий формируется отрицательный результат и производится операция уменьшения всех положительных элементов по m -му столбцу и k -й строке. При получении отрицательного результата блок упрощения графа ожидает прихода номера следующей строки.

5. Обновление списка объектов.

Последний шаг алгоритма выполняется на софт–процессорном ядре. На основе траекторного графа строится новый список объектов согласно пп. 3.1–3.5 оригинального алгоритма.

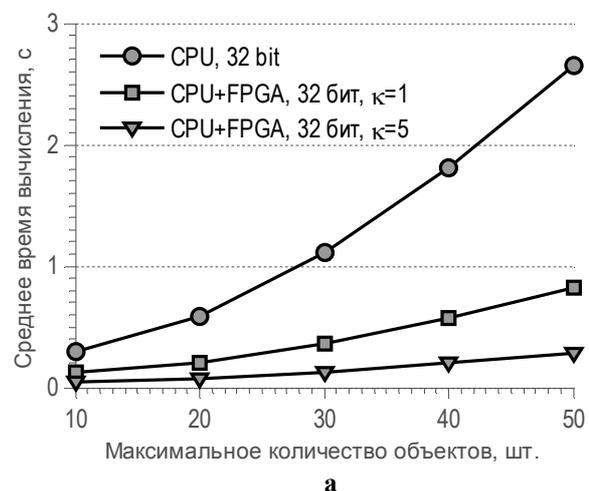
Экспериментальные исследования. Целью проведения экспериментальных исследований было получение ответа на два вопроса. Во-первых, исследовались ошибки слежения путем сравнения списков объектов на соответствующих кадрах для различных реализаций алгоритма. Во-вторых, измерялось среднее время, затраченное на вычисление списка.

Исследования выполнялись на наборе предварительно сгенерированных полутонных последовательностей изображений. Для получения списка объектов были использованы алгоритмы пороговой бинаризации, оценки параметров геометрических преобразований фона, морфологической фильтрации, разметки и параметризации бинарных изображений [2, 11, 12]. Результат работы этих алгоритмов использовался для подготовки входных данных изображений.

Параметры алгоритма были выбраны следующими: минимальная мера соответствия равна $1/18$ (максимальная дистанция равна 18 пикселей); максимальное время сохранения информации о слиянии – 10 кадров; максимальное время нахождения объекта в прогнозе – 10 кадров.

В результате экспериментов установлено, что оба алгоритма выдают одинаковый список объектов при использовании одинаковой точности расчетов. Это позволяет утверждать, что данная реализация соответствует исходному алгоритму.

Результаты измерения среднего времени, затраченного на формирование списка объектов на одном кадре, приведены на рисунках 5,а и 5,б. Точность задания координат – числа с фиксированной точкой 32 и 16 бит соответственно.



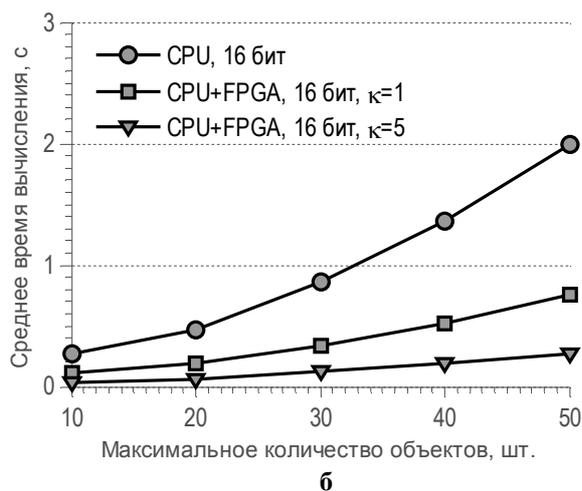


Рисунок 5 – Результаты экспериментальных исследований

Заключение. В статье описан алгоритм слежения за несколькими объектами. В основе данного алгоритма лежит идея построения и биения двудольного траекторного графа. Данный алгоритм позволяет следить за значительным числом целей притом, что объекты могут появляться, пропадать, распадаться и сливаться. Предложена новая реализация алгоритма, позволяющая использовать данный алгоритм в видеоинформационных системах реального времени на основе ПЛИС. Результаты, полученные в данной работе, показывают, что время вычисления может быть снижено в 5 – 10 раз, в зависимости от параметров реализации.

Благодарности. Публикация подготовлена в рамках выполнения научных исследований, осуществляемых ФГБОУ ВПО "РГРТУ" в соответствии с государственным заданием (№ государственной регистрации НИР 115011560084).

Библиографический список

1. Alpatov B., Korepanov S., Strotov V. "A composite algorithm for variable size object tracking for high performance FPGA-based on-board vision systems" Proc. of SPIE 9247, 92470A (2014).

2. Алпатов Б.А., Бабаян П.В., Балашиов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. – М.: Радиотехника, 2008. – 176 с.: ил.

3. Han M., Gong Y. Real-Time Object Tracking and Anomaly Detection // Proc. SPIE 5682, 173-182 (2005).

4. Lim Y., Lee M., Lee C., Kwon S., Lee J. Integrated Position and Motion Tracking Method for Online Multi-vehicle Tracking-by-detection // Optical Engineering 50 (7), 077203 (2011).

5. Konstantinova P., Udvarov A., Semerdjev T. A study of a target tracking algorithm using global nearest neighbor approach // Proc. Of Computer Systems and Technologies, pp. 290-295, New York, USA (2003).

6. Yang T., Li S., Pan Q., Li J. Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes // Proc. of IEEE, San Diego, CA, June 20-26 2005.

7. Алпатов Б.А., Бабаян П.В. Выделение движущихся объектов в последовательности мультиспектральных изображений при наличии геометрических искажений // Вестник Рязанского государственного радиотехнического университета. 2008. № 23. С. 42-48.

8. Alpatov B.A., Babayan P.V. Extraction of the objects observed on a non-uniform background during sensor motion // Proc. of SPIE 7835, 78350N (2010).

9. Alpatov B.A., Babayan P.V. "Multiple object tracking based on the partition of the bipartite graph" Proc. of SPIE 8186, 81860B (2011).

10. Alpatov B., Babayan P., Strotov V. The implementation of multiple objects tracking algorithm based on partition of bipartite graph in FPGA-based onboard vision systems // Proc. SPIE 9646, High-Performance Computing in Remote Sensing V, 964602 (2015).

11. Алпатов Б.А., Бабаян П.В., Стротов В.В. Анализ точностных характеристик методов слежения за фоновым изображением для бортовой видеоинформационной системы // Вестник Рязанского государственного радиотехнического университета. 2007. № 20. С. 3-10.

12. Alpatov B., Strotov V. "An estimation algorithm of the multispectral image geometric transformation parameters based on multiple reference area tracking," Proc. of SPIE 8713, 87130Q (2013).