УДК 004.932

МОДИФИЦИРОВАННЫЙ МЕТОД ВЫЧИСЛЕНИЯ ПЛОТНОГО ОПТИЧЕСКОГО ПОТОКА НА ПЛИС В РЕАЛЬНОМ ВРЕМЕНИ

А. В. Братулин, аспирант РГРТУ, Рязань, Россия;
orcid.org/0000-0003-3745-296X, e-mail: bratulin@mail.ru
М. Б. Никифоров, к.т.н., доцент кафедры ЭВМ РГРТУ, член-корреспондент академии информатизации образования, Рязань, Россия;
orcid.org/0000-0002-4796-0776, e-mail: nikiforov.m.b@evm.rsreu.ru
А.И. Ефимов, к.т.н., доцент кафедры ЭВМ РГРТУ, Рязань, Россия;
orcid.org/0000-0002-4014-8718, e-mail: lexie62rus@mail.ru

Одной из важных проблем в области систем технического зрения, является проблема реализации функций обнаружения и выделения движущихся объектов на видеоизображении высокой четкости в реальном времени. Данные функции реализуются с использованием понятия «оптический поток». В статье рассмотрен метод аппаратного вычисления плотного оптического потока в темпе поступления пикселей на основе корреляционно-экстремального алгоритма на видеоизображении высокой четкости. ***Целью работы*** *является разработка модификации метода вычисления плотного оптического потока на основе корреляционно-экстремального алгоритма в реальном масштабе времени на видеоизображении высокой четкости. Модифицированная версия метода должна обладать не меньшей производительностью по сравнению с исходной, но при этом занимать меньше аппаратных ресурсов. Предложен более оптимальный вариант устройства блока вычисления сумм абсолютных разностей для каждой точки установки окна эталона. Модификация содержит меньший размер внутренней памяти и меньшее количество сдвиговых регистров. Подробно рассмотрен принцип распараллеливания и конвейеризации, позволяющий минимизировать объем вычислений. Проанализированы аппаратные затраты при реализации метода на ПЛИС Xilinx Zynq Ultra Scale+ из состава отладочной платы MPSoC ZCU102 Evaluation Kit.*

Ключевые слова: *оптический поток, ПЛИС, видеоизображение, пиксель, корреляционный алгоритм, критериальная функция, эталон, зона поиска, окно эталона, абсолютная разность, конвейер, SAD, сдвиговый регистр.*

DOI: 10.21667/1995-4565-2019-70-146-157

Введение

В настоящее время системы машинного зрения – это одно из главных средств развития автоматических систем управления движением в условиях, когда объём априорной информации не достаточен и для решения задач управления необходим анализ внешней обстановки в режиме реального времени [1].

Вычисление оптического потока является одной из ключевых проблем в машинном зрении и используется для выделения объектов, сегментации, компрессии видео, 3D-реконструкции, технического зрения роботов [2].

Оптический поток – это изображение видимого движения т.е. сдвиг каждой точки между двумя кадрами видеоизображения. Он представляет собой поле скоростей (так как сдвиг эквивалентен мгновенной скорости), и для каждой точки предыдущего кадра видеоизображения находится такой сдвиг $(\Delta x, \Delta y)$, при котором исходной точке текущего кадра видеоизображения соответствовала точка на следующем кадре видеоизображения. Оптический поток, вычисленный для всех пикселей видеоизображения, называется плотным, для отдельных – разреженным [3].

Современные ПЛИС обладают большими логическими ресурсами и высокой производительностью для реализации широкого класса задач машинного зрения [4]. В отличие от дру-

гих видов элементной базы в ПЛИС можно в кратчайшие сроки организовать сложные параллельные алгоритмы цифровой обработки на аппаратном уровне. При этом быстродействие цифровой обработки резко возрастает [5, 6]. Поэтому ПЛИС является идеальной базой для прототипирования при разработке новых алгоритмов.

При реализации вычислителей оптического потока на ПЛИС чаще всего применяют либо метод Лукаса-Канаде (например: пирамидальный алгоритм вычисления оптического потока методом Лукаса-Канаде, предложенный Бедетом [7], оценка производительности которого приведена в [8], а одна из наиболее популярных версий реализации на ПЛИС описана в [9]), либо корреляционно-экстремальные алгоритмы [10] (например: метод аппаратного вычисления плотного оптического потока в реальном времени на ПЛИС, описанный в [11]). Однако эти методы при высокой производительности обладают существенным недостатком – высокие аппаратные затраты.

Целью данной **работы** является разработка модифицированной версии метода вычисления плотного оптического потока в реальном времени, реализация которого на ПЛИС обладает существенно меньшими аппаратными затратами.

Теоретическая часть

Как известно, одним из наиболее распространенных алгоритмов, основанным на сопоставлении точек, является алгоритм поиска минимальной суммы абсолютных разностей – **Sum of Absolute Difference (SAD)**. В этом алгоритме для каждой точки (x, y) кадра $N-1$ видеоизображения осуществляется поиск соответствующей точки $(x+\xi, y+\eta)$ в кадре N того же видеоизображения, которая является точкой лучшего совмещения фрагмента кадра $N-1$ видеоизображения – эталона, центрированного относительно текущей точки (x, y), и некоторой окрестности текущей точки кадра N видеоизображения – зоны поиска, т.е. определяются значения ξ, η, при которых достигается минимум критериальной функции $SAD(x, y, \xi, \eta)$ [12]:

$$SAD(x, y, \xi, \eta) = \sum_{i=-\frac{(w-1)}{2}}^{\frac{(w-1)}{2}} \sum_{j=-\frac{(w-1)}{2}}^{\frac{(w-1)}{2}} AD(x, y, \xi, \eta, i, j),$$

$$AD(x, y, \xi, \eta, i, j) = \left| I_{N-1}(x+i, y+j) - I_N(x+i+\xi, y+j+\eta) \right|,$$

где $AD(x, y, \xi, \eta, i, j)$ – абсолютная разность (Absolute Differences) яркостей точек $I_{N-1}(x+i, y+j)$ и $I_N(x+i+\xi, y+j+\eta)$; $I_{N-1}(x+i, y+j)$ – яркость точки $(x+i, y+j)$ кадра $N-1$ видеоизображения; $I_N(x+i+\xi, y+j+\eta)$ – яркость точки $(x+i+\xi, y+j+\eta)$ кадра N видеоизображения; w – размер эталона, центрированного относительно текущей точки с координатами (x, y).

Диапазон изменения ξ по x и η по y определяют зону поиска.

Таким образом, методика вычисления плотного оптического потока с использованием алгоритма поиска минимальной суммы абсолютных разностей следующий.

1. Эталон размером $w \times w$ центрируется относительно текущей точки (x, y) в кадре $N-1$. В начальный момент времени $x = \frac{w-1}{2}, y = \frac{w-1}{2}$.

2. В качестве текущей точки установки окна эталона в зоне поиска выбирается точка $(x+\xi, y+\eta)$, где $\xi = -\frac{k-1}{2}, \eta = -\frac{k-1}{2}$.

3. Вычисляется значение критериальной функции $SAD(x, y, \xi, \eta)$.

4. Окно эталона смещается вправо на 1 точку в зоне поиска, т.е. $\xi = \xi + 1$, и вычисляется значение критериальной функции $SAD(x, y, \xi, \eta)$ для новой точки установки окна эталона.

5. Далее с каждым шагом окно эталона движется по зоне поиска слева направо (инкремент ξ), сверху вниз (инкремент η) и для каждой новой точки установки окна эталона вычисляется значение критериальной функции $SAD(x, y, \xi, \eta)$ (рисунок 1).

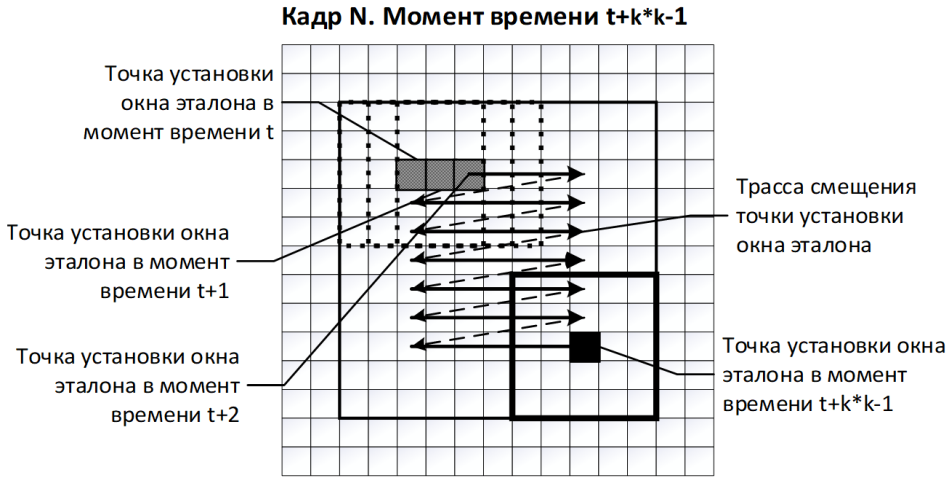

Рисунок 1 – Движение окна эталона по зоне поиска слева направо сверху-вниз
Figure 1 – Movement of the reference window in the search area from left to right from top to bottom

Если размер зоны поиска $(k+w-1)\times(k+w-1)$, то количество точек установки окна эталона $k\times k$. При этом каждая точка эталона сравнивается с $k\times k$ точками зоны поиска $\left((\xi,\eta)\in\left[-\frac{k-1}{2};\frac{k-1}{2}\right]\right)$. Таким образом, вычисляются $k\times k$ значений критериальной функции $SAD(x, y, \xi, \eta)$.

6. Выбирается окно, для которого $SAD(x, y, \xi, \eta)$ принимает минимальное значение, и центральная точка этого окна с координатами $(x+\xi, y+\eta)$ в кадре N видеоизображения считается соответствующей точке (x, y) в кадре $N-1$ видеоизображения. В результате получаем вектор оптического потока с началом в точке (x, y) и концом в $(x+\xi, y+\eta)$.

7. После этого осуществляется сдвиг текущей центральной точки эталона в кадре $N-1$ и зоны поиска в кадре $N(x, y)$ на одну координату вправо по горизонтали и выполняется процесс поиска минимального значения критериальной функции $SAD(x+1, y, \xi, \eta)$ для точки $(x+1, y)$.

Если размер кадра видеоизображения $X\times Y$ при частоте *N кадр/с,* то для поиска векторов оптического потока всех точек всех кадров видеоизображения за 1 секунду необходимо вычислить $X\times Y\times N$ векторов. Это означает, что для видеоизображения высокой четкости (разрешение 1920×1080) с прогрессивной разверткой и частотой кадров 50 кадр/с каждые 9,6 наносекунд необходимо формировать очередной вектор оптического потока. А чтобы вычислить один вектор, необходимо найти $k\times k$ значений критериальной функции $SAD(x, y, \xi, \eta)$ и $w\times w$ значений абсолютных разностей $AD(x, y, \xi, \eta, i, j)$ для каждой $SAD(x, y, \xi, \eta)$. Таким образом, чтобы выполнить указанное выше требование, необходимо сократить объем вычислений. Это можно сделать, например, путем повторного использования промежуточных результатов, полученных в процессе поиска минимума критериальной функции $SAD(x, y, \xi, \eta)$, при поиске минимума критериальной функции $SAD(x+1, y, \xi, \eta)$, $SAD(x+2, y, \xi, \eta)$ и т.д., как предложено в [11]. Суть данного метода заключается в следующем.

Предположим, что видеоизображение имеет размер 15×15 точек. На предыдущем шаге вычислен вектор оптического потока для точки (6, 7), эталон размером 5×5 и зона поиска размером 11×11 сдвинулись вправо на одну точку с целью вычисления вектора оптического потока для точки (7,7) $x=7, y=7$ (рисунок 2).

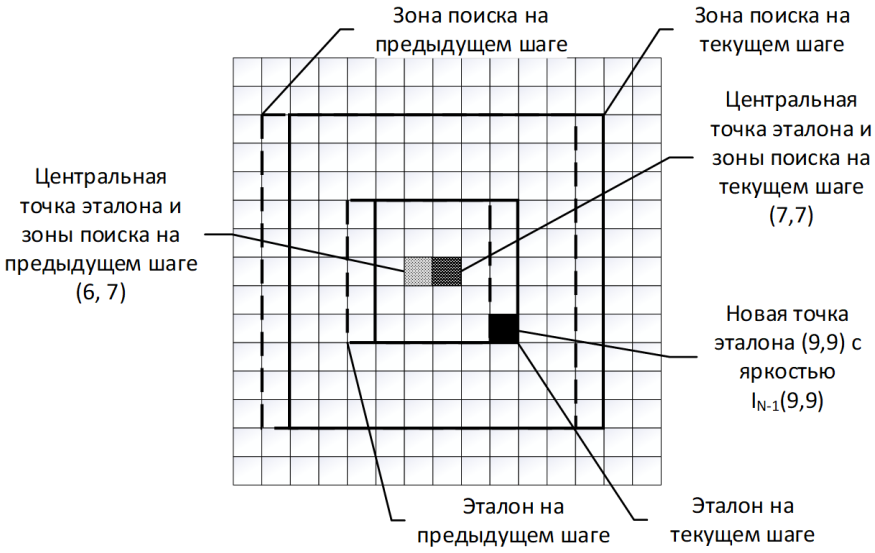

Рисунок 2 – Эталон 5x5 и зона поиска 11x11
Figure 2 – 5x5 reference and 11x11 search area

Далее эталон сравнивается с 7×7 (в общем случае $k\times k$) областями зоны поиска.

С целью достижения максимальной производительности вычисления плотного оптического потока в каждой точке строится конвейер из параллельно функционирующих 7×7 (в общем случае $k\times k$) блоков $SAD[m,n]$, где $m\in\{0,1,...,6\}, n\in\{0,1,...,6\}$ (в общем случае $m\in\{0,1,...,k-1\}, n\in\{0,1,...,k-1\}$ (рисунок 3)), каждый из которых соответствует определенной точке установки окна эталона в зоне поиска и вычисляет соответствующее значение критериальной функции $SAD(x,y,\xi,\eta)$ т.е. блок $SAD[m,n]$ вычисляет значение критериальной функции $SAD\left(x,y,m-\frac{k-1}{2},n-\frac{k-1}{2}\right)$. На выходе данного конвейера выполняется вычисление минимума среди полученных значений критериальной функции $SAD(x,y,\xi,\eta)$.

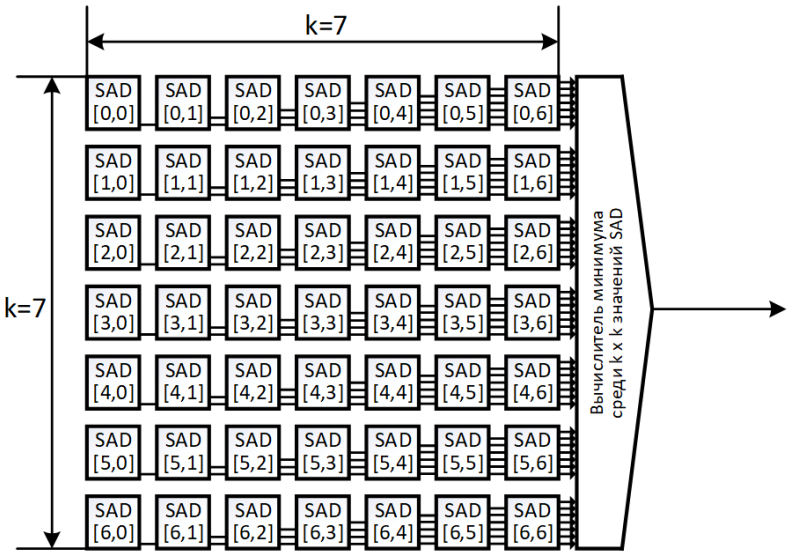

Рисунок 3 – Конвейер вычисления минимального значения критериальной функции $SAD(x,y,\xi,\eta)$

Figure 3 – Pipeline for calculating the minimum value of the criterion function $SAD(x,y,\xi,\eta)$

На рисунке 4 изображено устройство блока $SAD[1,0]$. Остальные блоки $SAD[m,n]$, где $m\in\{0,1,...,6\}, n\in\{0,1,...,6\}$, построены по такому же принципу. Обозначения следующие:

$AD_{1,0}(7,6)$ – это абсолютная разность яркости новой точки эталона $(9,9)$ в кадре $N-1$ и яркости соответствующей ей точки зоны поиска $(7,6)$ в кадре N при установке окна эталона в точке $(5,4)$, соответствующему блоку $SAD[1,0]$), вычисленная по формуле

$$AD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}, y+n-\frac{k-1}{2}+\frac{w-1}{2}\right)=$$
$$=AD\left(x,y,m-\frac{k-1}{2},n-\frac{k-1}{2},\frac{w-1}{2},\frac{w-1}{2}\right)= \quad (1)$$
$$=\left|l_{N-1}\left(x+\frac{w-1}{2},y+\frac{w-1}{2}\right)-l_N\left(x+m-\frac{k-1}{2}+\frac{w-1}{2},y+n-\frac{k-1}{2}+\frac{w-1}{2}\right)\right|.$$

Таким образом получим:

$$AD_{1,0}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2},7+0-\frac{7-1}{2}+\frac{5-1}{2}\right)=AD_{1,0}(7,6)=AD(7,7,-2,-3,2,2)=$$
$$=\left|I_{N-1}(7+2,7+2)-I_N(7+2-2,7+2-3)\right|=\left|I_{N-1}(9,9)-I_N(7,6)\right|. \quad (2)$$

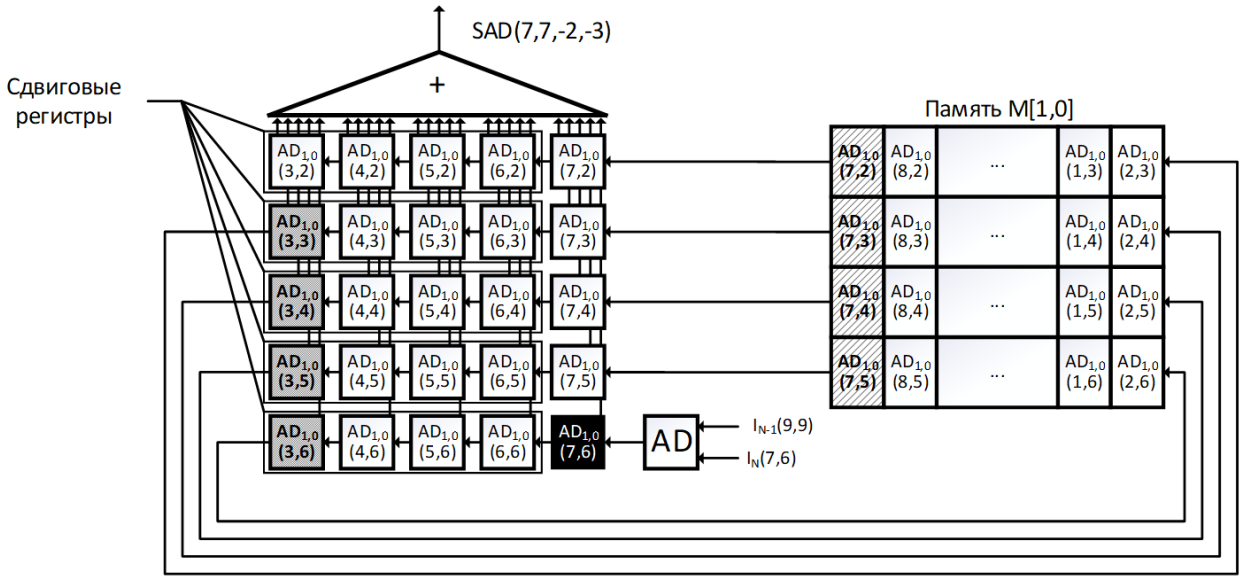

Рисунок 4 – Устройство блока SAD[1, 0]
Figure 4 – Block device SAD[1, 0]

На рисунке 5 изображено содержимое памятей абсолютных разностей $M[1,0]$ и $M[5,6]$ блоков $SAD[1,0]$ и $SAD[5,6]$ соответственно.

Процедура вычисления нового значения критериальной функции $SAD\left(x,y,m-\frac{k-1}{2},n-\frac{k-1}{2}\right)$ каждым блоком $SAD[m,n]$ следующая (на примере блока $SAD[1,0]$).

1. Из памяти $M[m,n]$ считываются $w-1$ значений абсолютных разностей, вычисленные на предыдущих шагах.

В данном случае из $M[1,0]$ считываются 4 значения абсолютных разностей $AD_{1,0}(7,j)$, где $j\in\{2,3,4,5\}$ (квадраты, заштрихованные косой чертой на рисунках 4, 5).

2. Вычисляется новая абсолютная разность $AD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2},y+n-\frac{k-1}{2}+\frac{w-1}{2}\right)$ по формуле (1).

Для данного примера получим (2).

Следует отметить, что $l_{N-1}(9,9)$ подается на вход всех блоков $SAD[m,n]$. Причем для каждого блока $SAD[m,n]$ точка, соответствующая новой точке эталона, $(9,9)$ своя, т.е. для $SAD[1,0]$ – точка $(7,6)$, для $SAD[5,6]$ – точка (C,C). Таким образом, необходим буфер на 6 строк длиной, 9 элементов и 7 сдвиговых регистров на 6 элементов (рисунок 6) (в общем случае $k-1$ строк длиной $X-k+1$ элементов и k сдвиговых регистров на $k-1$ элемент).

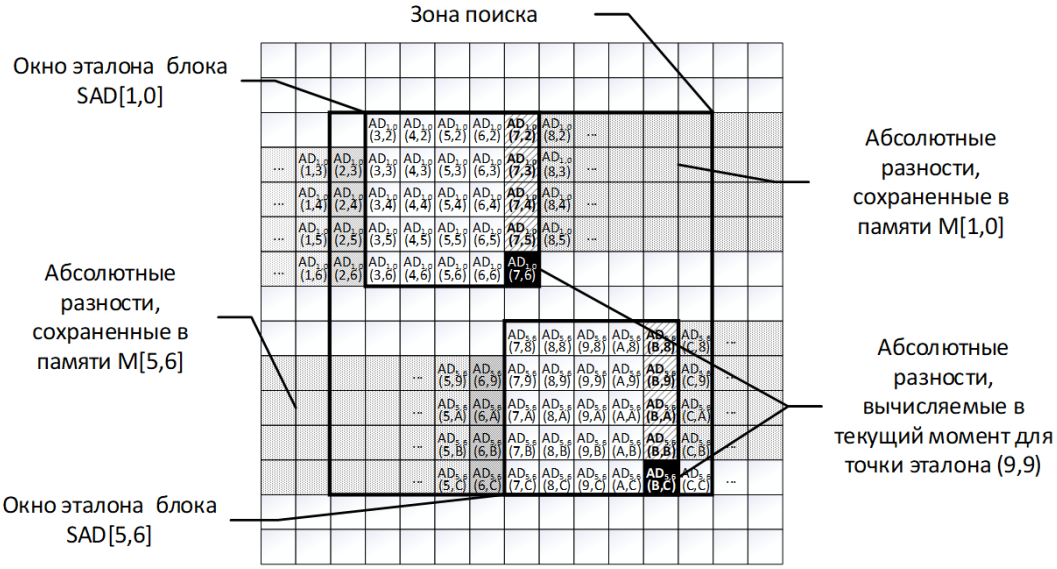

Рисунок 5 – Содержимое памятей абсолютных разностей $M[1,0]$ и $M[5,6]$
Figure 5 – Absolute difference memory content $M[1,0]$ и $M[1,0]$

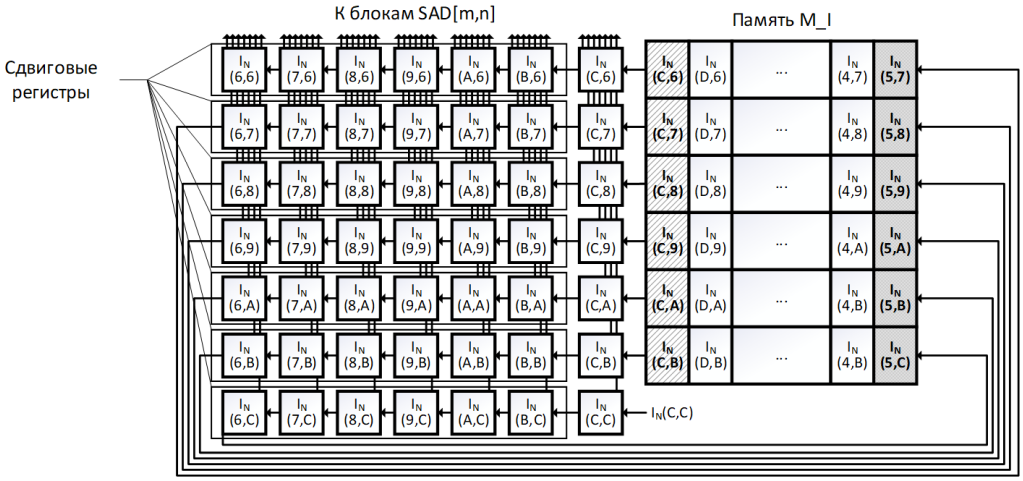

Рисунок 6 – Буфер с параллельным доступом к 7×7 элементам
Figure 6 – Buffer with parallel access to 7×7 elements

3. Абсолютные разности, считанные из памяти $M[m,n]$, задерживаются на $w-1$ такт на w сдвиговых регистрах для обеспечения одновременного доступа к $w \times w$ абсолютным разностям. При расчете нового значения критериальной функции значения абсолютных разностей продвигаются дальше по сдвиговому регистру.

В данном примере $AD_{1,0}(7,j)$ ($j \in \{2,3,...,6\}$) задерживаются на 4 такта в блоке $SAD[1,0]$ на сдвиговом регистре для обеспечения одновременного доступа к 5×5 абсолютным разностям.

4. Эти $w \times w$ абсолютных разностей с выходов сдвиговых регистров поступают на сумматор для вычисления нового значения критериальной функции $SAD\left(x, y, m-\frac{k-1}{2}, n-\frac{k-1}{2}\right)$.

В данном примере 5×5 абсолютных разностей $AD_{1,0}(i,j)$ $(i\in\{2,3,...,6\}, j\in\{3,4,...,7\})$ с выходов сдвиговых регистров поступают на сумматор для вычисления нового значения критериальной функции $SAD(7,7,-2,-3)$.

5. Значения абсолютных разностей, выталкиваемые из $w-1$ сдвиговых регистров, записываются обратно в память $M[m,n]$.

В данном примере $AD_{1,0}(3,j)$ $(j\in\{3,4,5,6\})$ – квадраты, заштрихованные плотной косой чертой на рисунках 4 и 5, выталкиваются из 4-сдвиговых регистров и записываются обратно в память $M[1,0]$.

Таким образом, каждая абсолютная разность $AD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}, y+n-\frac{k-1}{2}+\frac{w-1}{2}\right)$ суммируется w раз, пока она в сдвиговом регистре и сохраняется и считывается из памяти $w-1$ раз, т.е. используется в процессе вычислений $w\times w$ значений критериальной функции $SAD(x,y,\xi,\eta)$.

Путем повторения описанной выше процедуры для всех конвейеризированных и параллельно работающих $k\times k$ блоков $SAD[m,n]$ будем получать новый вектор оптического потока в каждом такте системной частоты.

Необходимо отметить, что память $M[0,1]$, а следовательно, и другие памяти $M[m,n]$ блоков $SAD[m,n]$ должны хранить абсолютные разности яркостей точек $w-1$ строк длиной $X-w+1$. Таким образом, общее количество параллельно работающих банков памяти должно быть $k\times k$ размером $(w-1)\times(X-w+1)$. Это является главным недостатком данного метода.

Для устранения данного недостатка предлагается модифицировать устройство каждого из блоков $SAD[m,n]$ путем сокращения объема памяти $M[m,n]$ и количества сдвиговых регистров. Для этого в памяти $M[m,n]$ будем хранить только X частичных сумм $w-1$ абсолютных разностей по столбцу. Устройство модифицированного блока $SAD[1,0]$ изображено на рисунке 7.

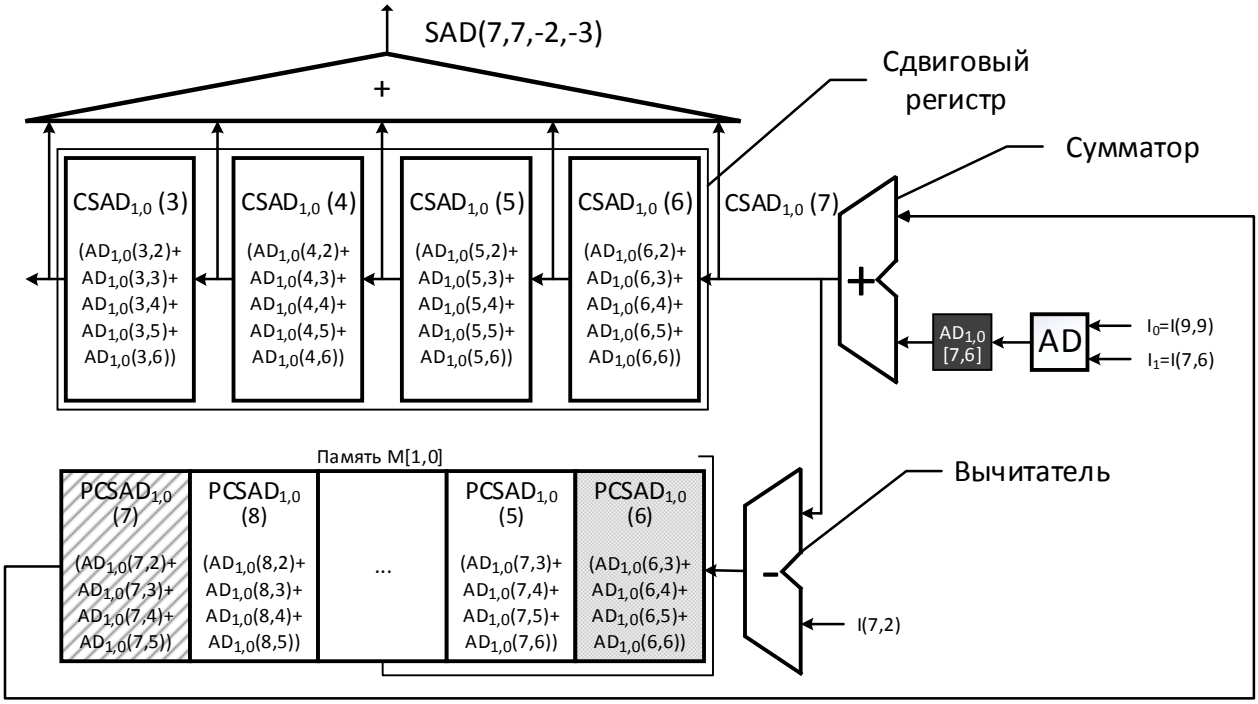

Рисунок 7 – Устройство модифицированного блока SAD[0,1]
Figure 7 – The device is a modified block SAD[0,1]

Таким образом, процедура вычисления нового значения критериальной функции $SAD\left(x, y, m-\frac{k-1}{2}, n-\frac{k-1}{2}\right)$ каждым модифицированным блоком $SAD[m,n]$ следующая (на примере блока $SAD[1,0]$).

1. Из памяти $M[m,n]$, функционирующей по принципу FIFO, считывается частичная сумма абсолютных разностей по текущему столбцу $PCSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$.

В данном случае из $M[0,1]$ считывается значение частичной суммы абсолютных разностей по текущему столбцу 7 $PCSAD_{1,0}(7)=AD_{1,0}(7,2)+AD_{1,0}(7,3)+AD_{1,0}(7,4)+AD_{1,0}(7,5)$.

2. Вычисляется новая абсолютная разность $AD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}, y+n-\frac{k-1}{2}+\frac{w-1}{2}\right)$ по формуле (1).

Для данного примера получим (2).

Следует отметить, что $I_{N-1}(9,9)$ подается на вход всех блоков $SAD[m,n]$. Причем для каждого блока $SAD[m,n]$ точка, соответствующая новой точке эталона $(9,9)$ своя т.е. для $SAD[1,0]$ – точка $(7,6)$, для $SAD[5,6]$ – точка (C,C). Таким образом необходим буфер на $k-1$ строк длиной $X-k+1$ элементов и k сдвиговых регистров на $k-1$ элемент (рисунок 7).

3. Сумматор вычисляет сумму абсолютных разностей по текущему столбцу $CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$ по формуле:

$$CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)=$$

$$=PCSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)+AD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}, y+n-\frac{k-1}{2}+\frac{w-1}{2}\right).$$

Таким образом для данного примера получим:

$$CSAD_{1,0}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}\right)=CSAD_{1,0}(7)=$$

$$PCSAD_{1,0}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}\right)+AD_{1,0}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}, 7+0-\frac{7-1}{2}+\frac{5-1}{2}\right)=$$

$$=PCSAD_{1,0}(7)+AD_{1,0}(7,6).$$

4. Вычитатель вычисляет обновленную частичную сумму абсолютных разностей по текущему столбцу $PCSAD^*_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$ по формуле:

$$PCSAD^*_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)=CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)-$$

$$-I_N\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}, y+n-\frac{k-1}{2}-\frac{w-1}{2}\right).$$

Для данного примера получим:

$$PCSAD^*_{m,n}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}\right)=PCSAD^*_{m,n}(7)=CSAD_{m,n}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}\right)-$$

$$-I_N\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}, 7+0-\frac{7-1}{2}-\frac{5-1}{2}\right)=CSAD_{m,n}(7)-I_N(7,2).$$

5. Обновленное значение $PCSAD^{*}_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$ записывается в память $M[m,n]$, выталкивая из нее устаревшее значение $PCSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$ и проталкивая остальные хранящиеся значения $PCSAD^{*}_{m,n}(i)$, где $i\in\left[0;x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)\cup$ $\cup\left(x+m-\frac{k-1}{2}+\frac{w-1}{2};X\right]$ влево (рисунок 7).

Для данного примера $PCSAD^{*}_{1,0}(7)$ записывается в память $M[1,0]$, выталкивая из нее устаревшее значение $PCSAD_{1,0}(7)$, проталкивая значения $PCSAD_{1,0}(8)...PCSAD_{1,0}(5)$, $PCSAD_{1,0}(6)$ дальше влево.

6. Сумма абсолютных разностей по текущему столбцу $CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}\right)$, вычисленная на шаге 3, подается на вход сдвигового регистра глубиной $w-1$ элементов для обеспечения одновременного доступа к w суммам абсолютных разностей по w столбцам $CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}-i\right)$, где $i\in\{0,1,...,w-1\}$. При вычислении нового значения критериальной функции значения сумм абсолютных разностей по столбцам продвигаются дальше по сдвиговому регистру.

В данном примере сумма абсолютных разностей по текущему столбцу $CSAD_{1,0}(7)$, вычисленная на шаге 3, подается на вход сдвигового регистра глубиной 4 элемента. В результате обеспечивается одновременный доступ к 5 суммам абсолютных разностей по 5 столбцам: $CSAD_{1,0}(7-i)$, где $i=0(1)4$.

7. w сумм абсолютных разностей по столбцам окна эталона $CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}-i\right)$, где $i\in\{0,1,...,w-1\}$, с выходов сдвигового регистра поступают на сумматор для вычисления нового значения критериальной функции $SAD\left(x,y,m-\frac{k-1}{2},n-\frac{k-1}{2}\right)$ по формуле:

$$SAD\left(x,y,m-\frac{k-1}{2},n-\frac{k-1}{2}\right)=\sum_{i=0}^{w-1}CSAD_{m,n}\left(x+m-\frac{k-1}{2}+\frac{w-1}{2}-i\right).$$

Для данного примера получим:

$$SAD\left(7,7,1-\frac{7-1}{2},0-\frac{5-1}{2}\right)=SAD(7,7,-2,-3)=$$

$$=\sum_{i=0}^{4}CSAD_{1,0}\left(7+1-\frac{7-1}{2}+\frac{5-1}{2}-i\right)=\sum_{i=0}^{4}CSAD_{1,0}(7-i).$$

Практическая часть

Данный модифицированный метод был реализован в виде отдельного модуля на ПЛИС Xilinx Zynq UltraScale+, входящей в состав отладочной платы Xilinx Zynq UltraScale+ MPSoC ZCU102 EvaluationKit [13], с параметрами, приведенными в таблице 1.

Выбор размеров эталона и зоны поиска, приведенных в таблице 1, осуществлялся эмпирическим путем. Установлено, что данных размеров достаточно, например, для реализации

функции стабилизации входного видеоизображения (устранение смещений, вызванных тряской источника видеосигнала), а также обнаружения движущихся объектов.

Сравнительный анализ объема занимаемых ресурсов ПЛИС Xilinx Zynq UltraScale+ модулями, реализующими исходный [11] (метод 1), модифицированный метод (метод 2) вычисления плотного оптического потока и пирамидальный метод Лукаса-Канаде (метод 3) [10], приведен в таблице 2.

Таблица 1 – Параметры разработанного модуля
Table 1 – Developed module parameters

№	Наименование параметра	Значение параметра
1	Разрешение видео	1920x1080
2	Частота кадров видео	60 Гц
3	Частота пикселей видео	148,500 МГц
4	Системная тактовая частота	150 МГц
5	Размер эталона	7x7
6	Размер зоны поиска	15x15

Таблица 2 – Объем занимаемых модулем ресурсов в ПЛИС Xilinx Zynq UltraScale+
Table 2 – The amount of resources occupied by the module in the FPGA Xilinx Zynq UltraScale +

Наименование ресурса	Тип метода	Объем доступного ресурса	Объем занимаемого ресурса	Доля занимаемого ресурса
SystemLogicCells (K)	1	599,550	34,313	5,7 %
	2		3,018	0,5 %
	3		34,007	5,6 %
Memory (Kb)	1	32,100	7,588	23,6 %
	2		2,196	6,8 %
	3		7,075	22,0 %
DSP Slices	1	2,520	0	0
	2		0,147	5,8 %
	3		0,196	7,8 %

Из таблицы 2 видно, что модифицированный метод вычисления плотного оптического потока по сравнению с исходным и пирамидальным методом Лукаса-Канаде занимает в 11 раз меньше логических ячеек (System Logic Cells) (3,018 против 34,313 и 34,007 соответственно), в 3,5 раза меньше памяти (Memory) (2,196 против 7,588 и 7,075 соответственно), но использует арифметические блоки (DSP Slices).

Заключение

В данной статье предложен модифицированный метод вычисления плотного оптического потока в темпе поступления видео (в реальном времени) на ПЛИС, предложенного в [11]. Благодаря изменению устройства и логике функционирования блока вычисления суммы абсолютной разности для заданной точки установки окна эталона, удалось достигнуть уменьшения аппаратных затрат, необходимых для реализации данного метода на ПЛИС, в 11 раз и 3,5 раза с точки зрения логических ячеек и памяти.

Высокое быстродействие, простота реализации и низкая ресурсоемкость метода позволяют использовать его в различных задачах машинного зрения и реализовывать совместно с другими функциональными модулями системы такими как модуль улучшения видения, формирования знакографической информации, аффинных преобразований. Компромисс между затратами ресурсов ПЛИС и производительностью позволяет гибко и быстро модифицировать предложенную схему под конкретное применение, что позволяет отказаться от использования дорогостоящих БИС.

Работа выполнена в рамках гранта Президента Российской Федерации для государственной поддержки молодых российских ученых, соглашение о предоставлении субсидии № 075-15-2019-703 от 11.06.2019.

Библиографический список

1. Техническое зрение в системах управления мобильными объектами-2010 // Труды научно-технической конференции-семинара. Вып. 4 / под ред. Р.Р. Назирова. М.: КДУ, 2011. 5 с.

2. **Елесина С. И., Никифоров М. Б., Логинов А. А., Костяшкин Л. Н.** Совмещение изображений в корреляционно-экстремальных навигационных системах / моногр. под ред. Л. Костяшкина, М. Никифорова. – М.: Радиотехника, 2015. 208 с.

3. **Абдухаликов А. А., Беляков П. В., Никифоров М. Б.** Реализация на ПЛИС алгоритма поиска ключевых точек на изображении // Международная научно-техническая и научно-методическая конференция «Современные технологии в науке и образовании». 2016. С. 103-108.

4. **Никифоров М. Б., Муратов Е. Р., Новиков А. И.** и др. Обработка изображений в авиационных системах технического зрения / под ред. Л. Н. Костяшкина, М. Б. Никифорова. М.: ФИЗМАТЛИТ, 2016. С. 28-32.

5. **Агафонов В.Ю.** Применение методов оптического потока для оценки совмещения изображений. / В. Ю. Агафонов // Образование и наука в России и за рубежом. 2017. № 6.

6. **Ефимов А. И., Устюков Д. И., Колчаев Д. А.** Особенности параллельной реализации операций математической морфологии на ПЛИС. Известия Тульского государственного университета. Технические науки. Вып. 3. Тула: Изд-во ТулГУ, 2019. С. 25-29.

7. **Алпатов Б. А., Хлудов С. Ю.** Алгоритмы последовательных испытаний в задаче совмещения двумерных изображений // Изв. вузов. Сер. Электромеханика. 1988. №7. С. 87-91.

8. **Bouguet J. Y. et al.** Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm // Intel Corporation. 2001. Vol. 5, no. 1-10, pp. 4.

9. **Babayan P. et al.** Real-time pyramidal Lukas-Kanade tracker performance estimation // Real-Time Image Processing and Deep Learning 2019. International Society for Optics and Photonics, 2019. Т. 10996. С. 109960L. DOI: 10.1117/12.2519274

10. **Monson J., Wirthlin M., Hutchings B. L.** Implementing high-performance, low-power FPGA-based optical flow accelerators in C // 2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors. IEEE, 2013. – С. 363-369.

11. **Bratulin A. V., Nikiforov M. B., Belyakov P. V., Kholopov E. Y.** Method for Computing Dense Optical Flow on FPGA in Real Time. Sovremennye informacionnye tehnologii i IT-obrazovanie. Modern Information Technologies and IT-Education. 2019; 15(2):320-330. DOI: 10.25559/SITITO.15.201902.320-330

12. **Алпатов Б. А., Бабаян П. В., Балашов О. Е., Степашкин А. И.** Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. М.: Радиотехника, 2008. 176 с.

13. Xilinx Zynq UltraScale+ MPSoCZCU102 Evaluation Kit [Электронный ресурс] // Xilinx – Adaptable. Intelligent, 2019. URL: https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html (дата обращения: 25.06.2019).

UDC 004.932

MODIFIED METHOD FOR REAL-TIME COMPUTING DENSE OPTICAL FLOW ON FPGA

A. V. Bratulin, post-graduate student, RSREU, Ryazan, Russia;
orcid.org/0000-0003-3745-296X, e-mail: bratulin@mail.ru
M. B. Nikiforov, Ph.D. (Tech.), associate professor, RSREU, Ryazan, Russia;
orcid.org/0000-0002-4796-0776, e-mail: nikiforov.m.b@evm.rsreu.ru
A. I. Efimov, Ph.D. (Tech.), associate professor, RSREU, Ryazan, Russia;
orcid.org/0000-0002-4014-8718, e-mail: lexie62rus@mail.ru

One of the important problems in the field of vision systems is the problem of implementing the functions of detection and selection of moving objects on the video image of high definition in real time. These func-

tions are implemented using the concept of optical flow. The article deals with the method of hardware calculation of dense optical flow in the rate of arrival of pixels on the basis of correlation-extreme algorithm on high-definition video image. ***The aim of the work*** *is to develop a modified version of the method for calculating the dense optical flow based on the correlation-extreme algorithm in real time on a high-definition video image. This modified version of the method should have no less performance than the original, but it takes less hardware costs. A more optimal variant of the device of the unit for calculating the sums of absolute differences for each installation point of the standard window is proposed. This version contains a smaller internal memory size and fewer shift registers. The principle of parallelization and pipelining, which allows to minimize the amount of calculations, is considered in detail. Hardware costs are analyzed when implementing the method on XILINX zynq Ultra Scale + FPGA from the MPSoC zcu102 Evaluation Kit debug Board.*

Key words: *optical flow, FPGA, video, pixel, correlation algorithm, criterion function, pattern, search zone, pattern window, absolute difference, pipeline, SAD, shift register.*

DOI: 10.21667/1995-4565-2019-70-146-157

References

1. Tehnicheskoe zrenie v sistemah upravlenija mobil'nymi obektami-2010 (Technical Vision in Mobile Object Management Systems-2010). *Trudy nauchno-tehnicheskoj konferencii-seminara*. Vol. 4. Pod red. R.R. Nazirova. M.: KDU, 2011. 5 p. (in Russian).

2. **Elesina S. I., Nikiforov M. B., Loginov A. A., Kostjashkin L. N.** *Sovmeshhenie izobrazhenij v korreljacionno-jekstremal'nyh navigacionnyh sistemah* (Images combining in correlation-extreme navigation systems) / monogr. pod red. L. Kostjashkina, M. Nikiforova. M.: Radiotehnika. 2015, 208 p. (in Russian).

3. **Abduhalikov A. A., Beljakov P. V., Nikiforov M. B.** Realizacija na PLIS algoritma poiska kljuchevyh tochek na izobrazhenii. *Mezhdunarodnaja nauchno-tehnicheskaja i nauchno-metodicheskaja konferencija «Sovremennye tehnologii v nauke i obrazovanii»*. 2016, pp. 103-108 (in Russian).

4. **Nikiforov M. B., Muratov Ye. R., Novikov A. I. and etc.** *Obrabotka izobrazhenij v aviacionnyh sistemah tehnicheskogo zrenija* **(**FPGA Implementation algorithm for the search for key points in the image). pod red. L. N. Kostjashkina, M. B. Nikiforova. M.: FIZMATLIT. 2016, pp. 28-32. (in Russian).

5. **Agafonov V. Ju.** Primenenie metodov opticheskogo potoka dlja ocenki sovmeshhenija izobrazhenij. *Obrazovanie i nauka v Rossii i za rubezhom*. 2017, no 6. (in Russian).

6. **Efimov A. I., Ustyukov D. I., Kolchayev D. A.** Osobennosti parallel'noy realizatsii operatsiy matematicheskoy morfologii na PLIS. Izvestiya *Tul'skogo gosudarstvennogo universiteta. Tekhnicheskiye nauki*. Issue 3. Tula: Izd-vo TulGU. 2019, pp. 25-29. (in Russian).

7. **Alpatov B. A., Hludov S. Yu.** Algoritmy posledovatel'nyh ispytanij v zadache sovmeshcheniya dvumernyh izobrazhenij. *Izv. vuzov. Ser. Elektromekhanika*. 1988, no. 7, pp. 87-91. (in Russian).

8. **Bouguet J.Y. et al.** Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*. 2001, vol. 5, no. 1-10, pp. 4.

9. **Babayan P. et al.** Real-time pyramidal Lukas-Kanade tracker performance estimation. *Real-Time Image Processing and Deep Learning 2019. International Society for Optics and Photonics*. 2019, vol. 10996, pp. 109960L. DOI: 10.1117/12.2519274

10. **Monson J., Wirthlin M., Hutchings B.L.** Implementing high-performance, low-power FPGA-based optical flow accelerators in C. *IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*. 2013. pp. 363-369.

11. **Bratulin A. V., Nikiforov M. B., Belyakov P. V., Kholopov E. Y.** Method for Computing Dense Optical Flow on FPGA in Real Time. *Sovremennye informacionnye tehnologii i IT-obrazovanie* – Modern Information Technologies and IT-Education. 2019; 15(2):320-330. DOI: 10.25559/SITITO.15.201902.320-330 (in Russian).

12. **Alpatov B. A., Babajan P. V., Balashov O.E., Stepashkin A.I.** *Metody avtomaticheskogo obnaruzhenija i soprovozhdenija ob'ektov. Obrabotka izobrazhenij i upravlenie* (Methods for automatic detection and tracking of objects. Image Processing and Management). M.: Radiotehnika, 2008. 176 p. (in Russian).

13. Xilinx Zynq UltraScale+ MPSoCZCU102 Evaluation Kit [Electronic resource]. Xilinx – Adaptable. Intelligent, 2019. URL: https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html (date of the request: 25.06.2019).