

ТЕХНИЧЕСКИЕ НАУКИ

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

В. А. Кучуков [V. A. Kuchukov]
М. Г. Бабенко [M. G. Babenko]
Е. А. Кучукова [E. A. Kuchukova]
Н. Г. Гудиева [N. G. Gudieva]

УДК 004.315

АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ ДВИЖЕНИЯ В ЦИФРОВЫХ СИСТЕМАХ ВИДЕОНАБЛЮДЕНИЯ

THE ANALYSIS OF MOTION DETECTION METHODS IN DIGITAL VIDEO SURVEILLANCE SYSTEMS

ФГАОУ ВО «Северо-Кавказский Федеральный университет» в г. Ставрополь

В статье исследованы различные методы детектирования движения в видеопотоке. Проведено моделирование на микрокомпьютере Raspberry Pi 2. Исследована возможность применения системы остаточных классов для данной задачи, приведена схема устройства детектирования движения, работающего в системе остаточных классов.

The article examines various methods of motion detection in the video stream. The modeling on the microcomputer Raspberry Pi 2 was carried out. The possibility of applying the residue number system for this problem was investigated, the scheme of the motion detection device operating in the residue number system is given.

Ключевые слова: система остаточных классов, детектирование движения, китайская теорема об остатках.

Key words: residue number system, motion detection, the Chinese remainder theorem.

Работа выполнена при поддержке стипендий Президента РФ молодым ученым и аспирантам СП-2236.2018.5 и СП-1215.2016.5.

Введение. Одним из основных этапов видеоаналитики в цифровых системах видеонаблюдения является задача вычитания фона, которая служит для классификации пикселей кадра видеопотока на два класса: пиксели заднего (фон) и переднего плана.

Материалы и методы. Для выполнения операции обнаружения движущихся объектов необходима некоторая априорная информация.

Рассмотрим видеопоток, состоящий из пикселей со значениями $f(x, y, t)$, где x и y – координаты пикселя в изображении размером $M \times N$ и t – отсчеты, которые берутся в дискретные моменты времени. Для изображения в оттенках серого данное значение является скалярной величиной, в то время как для цветного – векторной, зависящей, например, от значений красного, зеленого и синего каналов в цветовом пространстве RGB. Исследуем некоторые варианты решения задачи обнаружения движения.

Простейшим вариантом детектирования движения является случай со статическим фоном, например, в помещении без окон с постоянным искусственным освещением. Тогда в качестве фона может быть использован кадр, не содержащий никаких объектов. Пусть такой кадр получен в момент времени t_0 , тогда критерием, что пиксель (x, y) в последующие моменты времени t_i не является фоном служит неравенство [1]:

$$|f(x, y, t_i) - f(x, y, t_0)| \neq 0.$$

Однако в реальных системах видеонаблюдения даже в таких «идеальных» условиях, как помещение с постоянным освещением, будут иметь место искажения фона за счет шумов телекамеры и за счет теней от движущихся объектов.

щихся объектов. Для проверки работоспособности алгоритмов было разработано программное обеспечение на языке Python для микрокомпьютера Raspberry Pi 2. Результат работы данного способа показан на рис. 1.

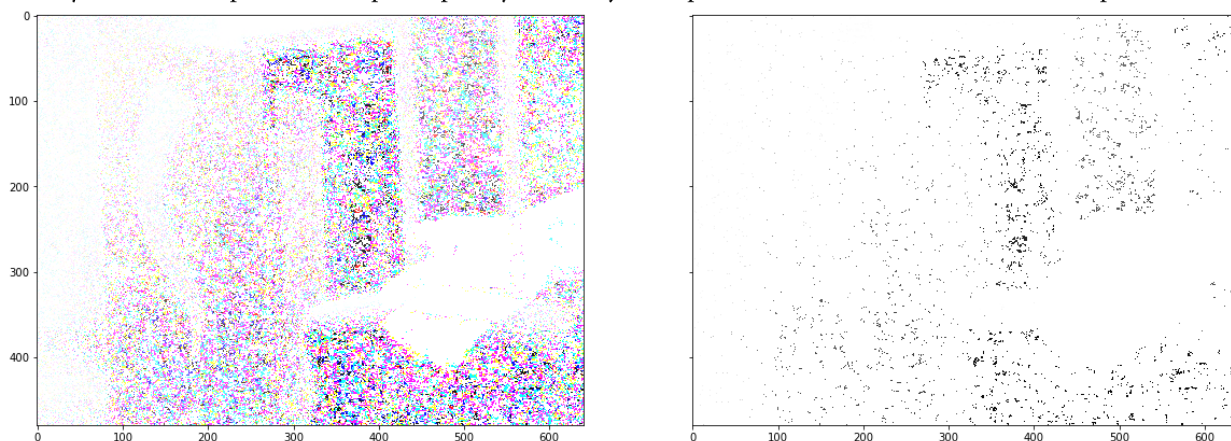


Рис. 1. Результат вычитания статического фона для цветного и монохромного изображения

Для цветного изображения операция вычитания фона на Raspberry Pi 2 занимала 0.130 с., в то время как для монохромного – 0.049 с. В то же время всего 2 % пикселей изображения в оттенках серого было отнесено к фону, у цветного изображения данный показатель составил около 30 %.

В большинстве случаев правильнее было бы говорить о квазистатическом фоне и сравнение должно вестись с некоторым порогом Z_{th} . Это связано с изменением самого фона, например, в случае поверхности воды, с изменением освещенности, как медленным в течение суток, так и быстрым из-за фар автомобиля или туч, заслонивших солнце. Также влияние может оказывать погодные условия, такие как дождь и туман. И конечно же помехи от электронных компонентов системы видеонаблюдения. Таким образом, пиксель будет относиться к фону, если выполняется неравенство:

$$|f(x, y, t_i) - f(x, y, t_0)| \leq Z_{th}. \quad (1)$$

Проблема выбора значения порога Z_{th} влияет на качество найденных движущихся объектов. Рассмотрим результаты моделирования вычитания фона (1) для различных значений порога. Поскольку абсолютные значения порога Z_{th} могут быть неэффективны для текущего уровня освещения (когда гистограмма изображения сдвинута в область темных или светлых пикселей), то одним из подходов к заданию порога может быть установка его в зависимости от средней яркости Z_{avg} . Увеличение значения порога будет приводить к большей помехоустойчивости, однако вероятность обнаружить объект заметно снизится. Результаты моделирования для различных значений порога показаны в табл. 1.

Таблица 1

Результат вычитания фона для различных пороговых значений

Значение порога Z_{th}	128	64	26	19	Z_{avg}	$\frac{1}{2}Z_{avg}$	$\frac{1}{4}Z_{avg}$	$\frac{1}{8}Z_{avg}$
Процент пикселей фона	99	98	93	92	99	96	93	91

При этом время выполнения операции было 0.10 с. для цветного изображения и 0.04 с. для изображения в оттенках серого. Для данного изображения объект занимает 7 % кадра. Результат работы для $Z_{th} = \frac{1}{4}Z_{avg}$ показан на рис. 2.

Можно заметить, что при меньшем времени работы качество выделения фона значительно улучшилось. Однако данная модель не обладает свойством адаптации к изменению фона, поскольку он выбирается один раз и используется на протяжении всего времени работы. Другим способом детектирования движения является использования в качестве фона предыдущие кадры видеопотока, при этом кадры могут быть несколько разнесены во времени. Недостатком такого способа является то, что объекты могут выглядеть размытыми или даже "двоиться".

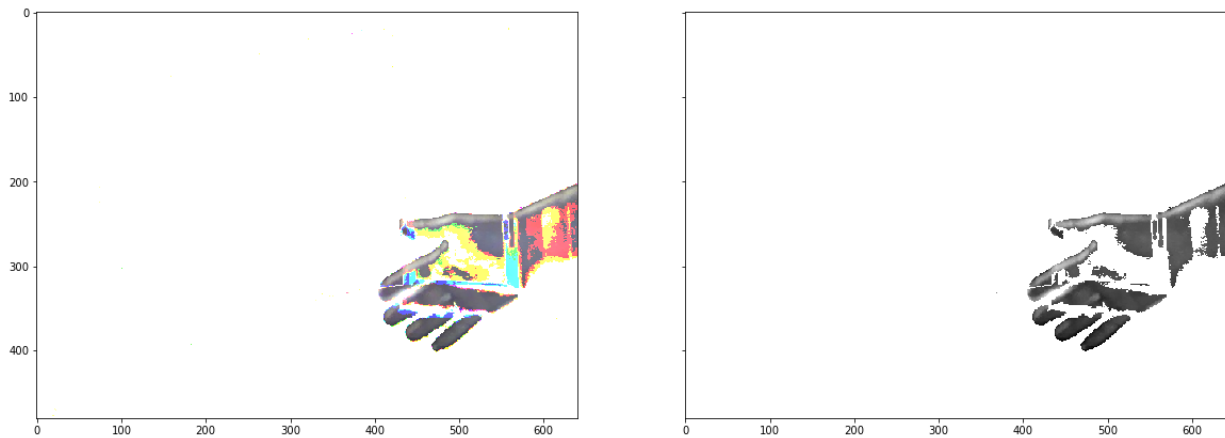


Рис. 2. Результат вычитания фона с заданным порогом для цветного и монохромного изображений

Ряд методов детектирования движения может быть основан на статистических методах [2]. Метод, показанный в статье [2], основан на следующем алгоритме:

1. По первым n кадрам видеопотока происходит обучение модели, т.е. оценка выборочного среднего значения m и среднего квадратичного отклонения σ для каждого пикселя по формулам:

$$m = \frac{1}{n} \sum_{i=1}^n f(x, y)_i, \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f(x, y)_i - m)^2},$$

где $f(x, y)_i$ – значения пикселя в точке (x, y) i -го кадра ($i = 1, \dots, n$).

2. Затем для каждого пикселя последующих кадров проверяют неравенство

$$\frac{|m - f(x, y)|}{\sigma} \leq e,$$

где e – некоторый порог. Если неравенство выполняется, то пиксель относится к фону, иначе – к переднему плану.

Данная модель может быть адаптивной, если производить следующее обновление параметров [2] для всех пикселей, отнесенных к фону:

$$m_t = (1 - \alpha_1) \cdot m_{t-1} + \alpha_1 f(x, y),$$

$$\sigma_t = \sqrt{(1 - \alpha_2) \sigma_{t-1}^2 + \alpha_2 (f(x, y) - m_t)^2},$$

где m_{t-1} , σ_{t-1} – значения выборочного среднего и среднего квадратичного отклонения на предыдущем кадре, m_t , σ_t – текущие параметры, α_1 , α_2 – параметры фильтра, отвечающие за скорость обучения модели. Для повышения производительности обновление параметров возможно производить через некоторое время, например, раз в несколько секунд.

Моделирование обучения данной модели на 5 кадрах заняло 0.804~с для цветного видеонаблюдения и 0.278~с для монохромного изображения. При этом время обработки одного кадра данным методом занимало 0.20~с для цветного и 0.069~с для монохромного изображений. Это в 1,5–2 раза больше, чем у предыдущего метода, при том что качество выделенного объекта несколько хуже, в чем можно убедиться на рис. 3.

Поскольку требованием к алгоритмам, работающим на маломощных вычислительных машинах, является простота и скорость работы, остановимся на методе с квазистатичным обновляемым фоном с порогом, зависящим от среднего значения яркости пикселей. Одними из методов повышения производительности является использование системы остаточных классов (СОК), которая позволяет заменить вычисления над большими числами на вычисления над вектором чисел меньшей размерности.

Рассмотрим вопрос выбора набора модулей для реализации системы видеонаблюдения в СОК. Поскольку в формуле (1) используется вычитание, то СОК должен единственным образом представлять числа в диапазоне $[-255, 255]$. Таким образом, рассмотренного в статье [3] набора $\{5, 7, 8\}$ уже недостаточно. Минимально необходимый динамический диапазон составляет 511.

Как показано в статье [4], оптимальным по производительности и занимаемым ресурсам при реализации на FPGA является набор модулей $\{2^n - 1, 2^n + 1, 2^k\}$. Для покрытия необходимого интервала чисел возьмем следующие параметры $n = 3$, $k = 4$. Тогда получим СОК $\{7, 9, 16\}$ с динамическим диапазоном $P = 1008$.

В случае выполнения более сложных арифметических операций понадобится другой набор модулей, например, $\{15, 16, 31\}$ с динамическим диапазоном $P = 7440$.

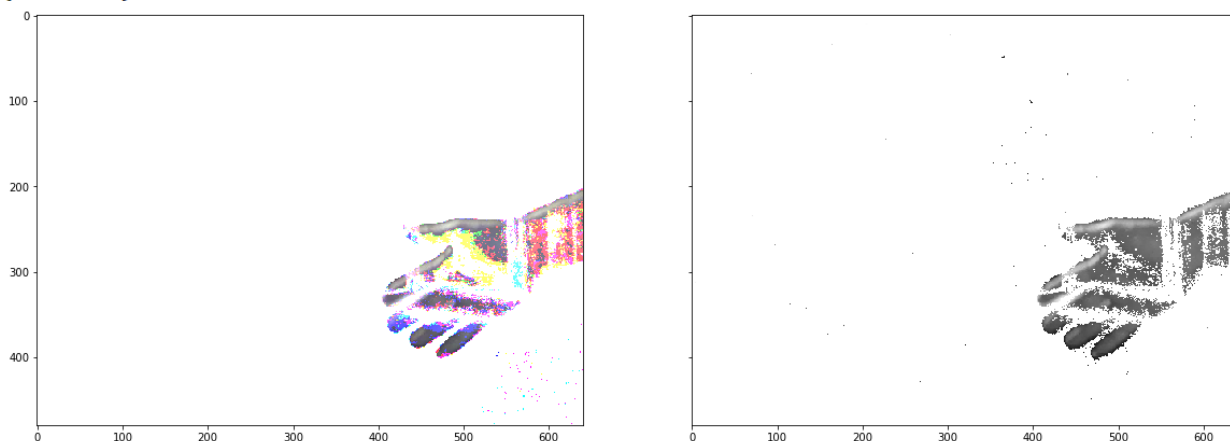


Рис. 3. Результат вычитания фона для цветного и монохромного изображения статистическим методом

Рассмотрим математическую модель вычитания фона в СОК на основе разработанных алгоритмов сравнения чисел.

Пусть задано изображение, значения пикселей которого представлены в системе остаточных классов с модулями $\{p_1, p_2, \dots, p_n\}$. Тогда для выполнения формулы (1) нам необходимо найти абсолютную величину $|f(x, y, t_i) - f(x, y, t_0)|$. Поскольку в СОК операция нахождения абсолютной величины является немодульной, поскольку необходимо определить знак числа, то для повышения эффективности работы найдем одновременно две разности $f(x, y, t_i) - f(x, y, t_0)$ и $f(x, y, t_0) - f(x, y, t_i)$, при этом определим знак первой разности, и если полученное значение больше нуля, возьмем первую разность, иначе вторую. Можно, конечно, для отрицательных чисел менять знак, вычисляя дополнения каждого остатка до модуля, но данная операция выполняется после определения знака, что несколько замедляет работу алгоритма. Классификация пикселей на фон и передний план также подразумевает сравнение с заданным порогом, поэтому из полученной после определения знака разности вычитают порог Z_{th} и снова определяют знак. Если число отрицательное, то разница яркостей исходных пикселей небольшая и пиксель $f(x, y, t_i)$ можно отнести к фону, в случае положительного результата пиксель принадлежит движущемуся объекту.

Для определения знака в данной модели возможно применение различных методов, описанных в статьях [5, 6]. Эффективным методом сравнения является приближенный метод на основе КТО:

$$\left\lfloor \frac{x}{p} \right\rfloor_1 = \left\lfloor \sum_{i=1}^n |P_i^{-1}|_{p_i} \cdot \frac{x_i}{p_i} \right\rfloor_1 = \left\lfloor \sum_{i=1}^n k_i \cdot x_i \right\rfloor_1, \quad (2)$$

где $k_i = \frac{|P_i^{-1}|_{p_i}}{p_i}$. С целью уменьшения количества операций в приближенном методе в [7] предложено использовать следующую формулу

$$C(X) = \left\lfloor \sum_{i=1}^n W_i x_i \right\rfloor_1, \quad (3)$$

где $W_i = \left\lfloor \frac{2^N |P_i^{-1}|_{p_i}}{p_i} \right\rfloor / 2^N$, $|x|_1$ – дробная часть числа x , $N = \lceil \log_2(P \cdot \rho) \rceil$ и $\rho = \sum_{i=1}^n (p_i - 1)$. Преимущество данного способа относительно формулы (2) состоит в том, что он не требует дополнительных операций округления вверх, однако при этом увеличились размеры операндов.

При этом порог Z_{th} может быть как предварительно вычисленным значением, так и вычисляться в ходе детектирования движения. Таким образом, схема детектора движения в системе остаточных классов будет выглядеть следующим образом (рис. 4). На вход данной схемы поступает текущий кадр, фон и порог, а на выход значения $\{0, 1\}$.

Рассмотрим пример реализации данной математической модели. Возьмем область монохромного изображения размером 5×5 и представим её в СОК. Для выполнения операции детектирования движения с заданным порогом по формуле (1) возможно использование небольшого набора модулей, поскольку в этом случае будут необходимы операции вычитания и сравнения. Возьмем СОК с набором модулей $\{7, 9, 16\}$ и фиксиро-

ванным порогом $Z_{th} = 26 = (5, 8, 10)$. Данная СОК может однозначно представить числа, удовлетворяющие двойному неравенству $-504 \leq X \leq 503$ и для определения знака необходимо будет сравнивать числа с серединой диапазона $503 = (6, 8, 7)$.

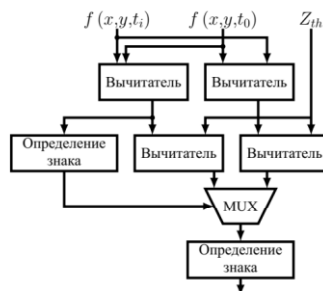


Рис. 4. Схема детектирования движения в системе остаточных классов

Точность, необходимая для корректных вычислений, составляет $N = 15$ знаков. Вычислим параметры W_i :

$$W_1 = \frac{\left\lfloor \frac{2^N |P_1^{-1}|_{p_1}}{p_1} \right\rfloor}{2^N} = \frac{\left\lfloor \frac{2^{15} \cdot 2}{7} \right\rfloor}{2^{15}} = 0.2857055664,$$

$$W_2 = \frac{\left\lfloor \frac{2^N |P_2^{-1}|_{p_2}}{p_2} \right\rfloor}{2^N} = \frac{\left\lfloor \frac{2^{15} \cdot 7}{9} \right\rfloor}{2^{15}} = 0.7777709961,$$

$$W_3 = \frac{\left\lfloor \frac{2^N |P_3^{-1}|_{p_3}}{p_3} \right\rfloor}{2^N} = \frac{\left\lfloor \frac{2^{15} \cdot 15}{16} \right\rfloor}{2^{15}} = 0.9375000000.$$

Тогда для определения знака необходимо сравнивать с $C(6, 8, 7) = 0.49890137$.

Пусть на вход подаются изображения:

$$f_0 = \begin{pmatrix} 21 & 23 & 23 & 23 & 23 \\ 21 & 23 & 22 & 23 & 24 \\ 21 & 23 & 22 & 23 & 24 \\ 22 & 24 & 23 & 24 & 24 \\ 23 & 24 & 23 & 24 & 23 \end{pmatrix} \text{ и } f_1 = \begin{pmatrix} 21 & 24 & 33 & 23 & 28 \\ 27 & 30 & 30 & 55 & 76 \\ 25 & 25 & 25 & 78 & 122 \\ 22 & 39 & 87 & 132 & 145 \\ 28 & 89 & 136 & 146 & 157 \end{pmatrix}.$$

Рассмотрим пиксели $f_1(0, 0) = 21 = (0, 3, 5)$ и $f_1(4, 4) = 157 = (3, 4, 13)$. Проведем вычисления для $f_1(0, 0)$:

$$f_1(0, 0) - f_0(0, 0) = (0, 3, 5) - (0, 3, 5) = (0, 0, 0),$$

$$C(0, 0, 0) = 0 < C(6, 8, 7) = 0.49890137 \rightarrow \text{положительное},$$

$$(0, 0, 0) - (5, 8, 10) = (2, 1, 6), \quad C(2, 1, 6) = 0.974182129 > C(6, 8, 7) \rightarrow \text{Фон}.$$

Таким образом, $f_1(0, 0)$ можно классифицировать как фон. Проверим $f_1(4, 4)$:

$$f_1(4, 4) - f_0(4, 4) = (3, 4, 13) - (2, 5, 7) = (1, 8, 6)$$

$$C(1, 8, 6) = 0.13287354 < C(6, 8, 7) = 0.49890137 \rightarrow \text{положительное}$$

$$(1, 8, 6) - (5, 8, 10) = (3, 0, 12), \quad C(3, 0, 12) = 0.10711670 < C(6, 8, 7) \rightarrow \text{Объект}$$

Таким образом, $f_1(4, 4)$ можно классифицировать как движущийся объект.

Заключение. В статье рассмотрены различные методы детектирования движения, проведено их моделирование на Raspberry Pi 2. Для повышения производительности предложена схема детектирования движения в СОК. Данная схема может быть реализована программно на FPGA с заранее заданным порогом. В случае вычисления порога, как некоторого среднего, для повышения эффективности использования СОК необходимо взять набор модулей с большим динамическим диапазоном, например, $\{15, 16, 31\}$ с динамическим диапазоном $P = 4590$.

ЛИТЕРАТУРА

1. Алексеев С. А., Волхонский В. В., Суханов А. В. Телевизионные системы наблюдения. Основы проектирования. СПб.: Университет ИТМО, 2015.

2. Нужный С. П., Червяков Н. И. Детектор движения в цифровой системе охранного видеонаблюдения // Электронный Интернет-журнал Graphicon [Электронный ресурс]. 2007. Режим доступа: www.graphicon.ru/2007/proceedings/Papers/Paper_7.pdf, свободный. 2007. Т. 13. №. 08.
3. Taleshmekeail D. K., Mousavi A. The use of residue number system for improving the digital image processing // Signal Processing (ICSP), 2010 IEEE 10th International Conference on. IEEE, 2010. С. 775-780.
4. Chervyakov N. I., Babenko M. G., Kuchukov V. A. Research of effective methods of conversion from positional notation to RNS on FPGA // Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017 IEEE Conference of Russian. IEEE, 2017. С. 277-281.
5. Van Vu T. Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding // IEEE Transactions on Computers. 1985. Т. 100. №. 7. С. 646-651.
6. Червяков Н. И. Методы, алгоритмы и техническая реализация основных проблемных операций, выполняемых в системе остаточных классов //Инфокоммуникационные технологии. 2011. Т. 9. №. 4. С. 4-12.
7. Chervyakov N. I. et al. Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem //International Journal of Computer Mathematics. 2017. Т. 94. №. 9. С. 1833-1849.

REFERENCES

1. Alekseev S. A., Volhonskij V. V., Suhanov A. V. Televizionnye sistemy nablyudeniya. Osnovy proektirovaniya. SPb.: Universitet ITMO, 2015.
2. Nuzhnyj S. P., Chervyakov N. I. Detektor dvizheniya v cifrovoj sisteme ohrannogo videonablyudeniya // EHlektronnyj Internet-zhurnal Graphicon [EHlektronnyj resurs]. 2007. Rezhim dostupa: www.graphicon.ru/2007/proceedings/Papers/Paper_7.pdf, svobodnyj. 2007. Т. 13. №. 08.
3. Taleshmekeail D. K., Mousavi A. The use of residue number system for improving the digital image processing // Signal Processing (ICSP), 2010 IEEE 10th International Conference on. IEEE, 2010. С. 775-780.
4. Chervyakov N. I., Babenko M. G., Kuchukov V. A. Research of effective methods of conversion from positional notation to RNS on FPGA //Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017 IEEE Conference of Russian. IEEE, 2017. С. 277-281.
5. Van Vu T. Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding // IEEE Transactions on Computers. 1985. Т. 100. №. 7. С. 646-651.
6. Chervyakov N. I. Metody, algoritmy i tekhnicheskaya realizaciya osnovnyh problemnyh operacij, vypolnyaemyh v sisteme ostatocnyh klassov // Infokommunikacionnye tekhnologii. 2011. Т. 9. №. 4. С. 4-12.
7. Chervyakov N. I. et al. Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem //International Journal of Computer Mathematics. 2017. Т. 94. №. 9. С. 1833-1849.

ОБ АВТОРАХ

Кучуков Виктор Андреевич, младший научный сотрудник Управления науки и технологий СКФУ, тел.: 89097537700, e-mail: vkuchukov@ncfu.ru

Kuchukov Viktor Andreevich, Junior Researcher, Department of Science and Technology, tel.: 89097537700, e-mail: vkuchukov@ncfu.ru

Бабенко Михаил Григорьевич, доцент кафедры прикладной математики и математического моделирования СКФУ, тел.: +79187535673, e-mail: mgbabenko@ncfu.ru

Babenko Mikhail Grigorievich, Associate Professor of Department of Applied Mathematics and Mathematical Modeling, tel.: +79187535673, e-mail: mgbabenko@ncfu.ru

Кучукова Екатерина Андреевна, студент кафедры высшей математики СКФУ, тел.: (8652)956510, e-mail: ekuchukova@ncfu.ru

Kuchukova Ekaterina Andreevna, student of department of Higher Mathematics, tel.: (8652)956510, e-mail: ekuchukova@ncfu.ru

Гудиева Наталья Григорьевна, аспирант кафедры прикладной математики и математического моделирования СКФУ, e-mail: whbear@yandex.ru

Gudieva Natalia Grigorievna, post-graduate student of the Department of Applied Mathematics and Mathematical Modeling, NCFU, e-mail: whbear@yandex.ru

АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ ДВИЖЕНИЯ В ЦИФРОВЫХ СИСТЕМАХ ВИДЕОНАБЛЮДЕНИЯ**В. А. Кучуков, М. Г. Бабенко, Е. А. Кучукова, Н. Г. Гудиева**

В статье исследованы различные методы детектирования движения в видеопотоке. Проведено моделирование на микрокомпьютере Raspberry Pi 2. Были рассмотрены ряд методов детектирования движения, основанный на статистических методах, и метод с квазистатичным обновляемым фоном с порогом, зависящим от среднего значения яркости пикселей. Для повышения производительности предложена схема детектирования движения в СОК. Данная схема может быть реализована программно на FPGA с заранее заданным порогом. В случае вычисления порога, как некоторого среднего, для повышения эффективности использования СОК необходимо взять набор модулей с большим динамическим диапазоном, например, {15, 16, 31} с динамическим диапазоном $P = 4590$.

ANALYSIS OF MOTION DETECTION METHODS IN DIGITAL VIDEO SURVEILLANCE SYSTEMS**V. A. Kuchukov, M. G. Babenko, E. A. Kuchukova, N. G. Gudieva**

In the article, various methods of detecting motion in a video stream are investigated. A simulation was performed on a Raspberry Pi microcomputer 2. A number of motion detection methods based on statistical methods and a method with a quasi-static updated background with a threshold depending on the average brightness of the pixels were considered. To improve performance, a motion detection scheme is proposed in the RNS. This scheme can be implemented programmatically on an FPGA with a predetermined threshold. In the case of calculating the threshold as an average, in order to increase the efficiency of using the RNS, it is necessary to take a set of modules with a large dynamic range, for example, {15, 16, 31} with a dynamic range $P = 4590$.