

УДК 519.684.6

DOI: 10.22213/2410-9304-2019-3-4-115-119

СРАВНЕНИЕ РЕАЛИЗАЦИЙ ВЫЧИСЛЕНИЯ ВЕКТОРОВ СМЕЩЕНИЯ ВИДЕОИЗОБРАЖЕНИЙ
ВО ВСТРАИВАЕМЫХ СИСТЕМАХ

П. А. Ушаков, доктор технических наук, профессор, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
А. Ю. Печенкин, кандидат технических наук, доцент, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия

В работе рассмотрены алгоритмы вычисления векторов смещения видеоизображений, которые находят применение в смартфонах, цифровых фотокамерах, различных специализированных изделиях для стабилизации изображений, контроля дорожного движения, при аэрофотосъемках, при сопровождении объектов и в других приложениях. Для сравнительной оценки эффективности использования вычислительных ресурсов при реализации алгоритмов вычисления векторов смещения были выбраны: алгоритм на основе метода Лукаса – Кэнейда и алгоритм, основанный на согласованной фильтрации изображения. Определялось глобальное смещение изображения с использованием аппаратной платформы – системы на кристалле Xilinx семейства Ultrascale+. Матричные вычисления алгоритма Лукаса – Кэнейда были реализованы 4 одинаковыми аппаратными блоками, что позволило вычислять смещения между кадрами одновременно для 4 точек. В этом случае время вычисления оптического потока составило в среднем 7,5 мс. В реализации алгоритма согласованной фильтрации изображения на базе ДДПФ также использовалось 4 аппаратных блока, позволяющих распараллелить вычисления. В результате время работы алгоритма составило 2,9 мс, при этом процессорное время, необходимое для работы алгоритма, составило 0,2 мс. В процессе реализации определены слабые места алгоритмов, необходимое количество аппаратных ресурсов системы на кристалле. Показано, что при наличии достаточных аппаратных ресурсов определение смещения соседних кадров видеоряда гораздо эффективнее реализуется на базе согласованной фильтрации, чем алгоритмом Лукаса – Кэнейда.

Ключевые слова: алгоритм Лукаса – Кэнейда, трекинг ключевых точек, оптический поток, согласованная фильтрация изображений.

Введение

Вычисление векторов смещения видеоизображений в настоящее время широко применяется в алгоритмах стабилизации изображений, системах контроля дорожного движения, при аэрофотосъемках, при сопровождении объектов и в других приложениях [1–7]. Эти алгоритмы находят применение в смартфонах, цифровых фотокамерах, различных специализированных изделиях. Однако вычисление векторов смещения – задача достаточно тяжелая и требует значительных вычислительных ресурсов [8].

Самый распространенный метод вычисления вектора смещения изображения – метод Лукаса – Кэнейда (Lucas-Kanade – L-K) [9, 10].

Известен также алгоритм, основанный на согласованной фильтрации изображения, описанный, например, в [11]. Согласованная фильтрация изображения позволяет определить положение фрагмента изображения на кадре, используя прямое и обратное дискретное двумерное преобразование Фурье (ДДПФ).

Кроме того, существует реализация алгоритма Лукаса – Кэнейда в среде Vivado HLS [12]. Однако в нашей работе эта реализация не рассматривается, поскольку она вычисляет лишь направление, но не величину смещения.

Метод Лукаса – Кэнейда позволяет определить вектор смещения точки на двух соседних кадрах в пределах некоторой окрестности (ок-

на), используя исходные изображения и производные от них, вычисленные с помощью оператора Шарра. При этом определяется только направление смещения, а не величина смещения. Чтобы определить величину смещения, с определенной точностью используют итеративную процедуру Ньютона – Рафсона, оценивая максимальный градиент оптического потока после каждой итерации [13]. Это требует многократного вычисления оптического потока, в большинстве случаев для обычного видео 5–10 итераций. Максимальное смещение, которое можно определить таким методом, – это смещение в пределах окна. В известных нам реализациях размер окна выбирался в пределах 4–20 пикселей по вертикали и по горизонтали. Для расширения максимальной области поиска смещения используют пирамидальный итерационный метод Лукаса – Кэнейда. В этом случае создают уменьшенные копии исходных кадров, между которыми требуется определить смещение (Гауссову пирамиду [14]), и определение смещения начинается с самого низкого разрешения. Затем смещение уточняется на изображении с большим разрешением (на следующем уровне пирамиды) и так до исходного размера изображения. При этом максимальное детектируемое таким алгоритмом смещение определяется размером окна на изображении самого низкого разрешения. В этом случае вычисление оптического по-

тока итерационным методом Лукаса – Кэнейда требуется на каждом уровне пирамиды изображений. Учитывая, что для надежного определения смещения требуется обычно от 20 до 200 точек, и для каждой точки оптический поток должен быть вычислен итерационно и на нескольких уровнях пирамиды, и для каждого уровня пирамиды необходимо вычислить производные Шарра, то задача определения смещения объекта на изображении или изображения в целом требует значительных вычислительных затрат.

Алгоритм, основанный на согласованной фильтрации изображения, реализуется следующим образом [15]. На двух соседних кадрах вычисляется положение фрагмента изображения вокруг точки, для которой необходимо определить смещение. Область изображения, в пределах которой необходимо найти заданный фрагмент изображения, подвергается ДДПФ. Если область изображения размером $I_1 \times I_2$ пикселей обозначить как

$$x(i_1, i_2) = x_{i_1, i_2}, \quad i_1 = \overline{0, I_1 - 1}, \quad i_2 = \overline{0, I_2 - 1},$$

то ДДПФ этой области изображения вычисляется как

$$\tilde{X}_{k_1, k_2} = \sum_{i_1=0}^{I_1-1} \sum_{i_2=0}^{I_2-1} \tilde{x}_{i_1, i_2} W_1^{i_1, k_1} W_2^{i_2, k_2},$$

$$W_1 = \exp\left(j \frac{2\pi}{I_1}\right), \quad W_2 = \exp\left(j \frac{2\pi}{I_2}\right),$$

где \tilde{x}_{i_1, i_2} – циклическое продолжение по вертикали и горизонтали конечного фрагмента $x(i_1, i_2)$.

В результате получается двумерный спектр \tilde{X}_{k_1, k_2} области изображения. Для того чтобы искомым фрагмент мог использоваться в качестве согласованного фильтра, комплексный спектр такого фильтра должен быть равен выражению, комплексно сопряженному с выражением, описывающим спектр обнаруживаемого фрагмента. Поэтому производится ДДПФ искомого фрагмента аналогичным образом и формируется комплексно-сопряженный спектр этого фрагмента. Обозначим его \tilde{Y}_{k_1, k_2}^* . Далее спектр области изображения и комплексно-сопряженный спектр искомого фрагмента перемножаются:

$$\tilde{Z}_{k_1, k_2} = \tilde{Y}_{k_1, k_2}^* \cdot \tilde{X}_{k_1, k_2},$$

результат подвергается обратному ДДПФ:

$$\tilde{z}_{i_1, i_2} =$$

$$= \frac{1}{I_1 I_2} \sum_{k_1=0}^{I_1-1} \sum_{k_2=0}^{I_2-1} \tilde{Z}_{k_1, k_2} \exp\left(j \frac{2\pi}{I_1} i_1 k_1 + j \frac{2\pi}{I_2} i_2 k_2\right).$$

В результате получаем изображение \tilde{z}_{i_1, i_2} , в котором степень яркости каждого пикселя соответствует степени схожести искомого фрагмента с фрагментом изображения по координатам этого пикселя.

Целью работы является сравнение различных реализаций алгоритмов вычисления векторов смещения видеоизображений для определения наиболее эффективного алгоритма, обеспечивающего минимальные вычислительные затраты во встраиваемых системах.

Реализация и сравнение алгоритмов

В качестве сравниваемых алгоритмов вычисления векторов смещения были выбраны: алгоритм на основе метода Лукаса – Кэнейда и алгоритм, основанный на согласованной фильтрации изображения. Определялось глобальное смещение изображения с использованием аппаратной платформы – системы на кристалле Xilinx семейства Ultrascale+.

Алгоритм Лукаса – Кэнейда был реализован со следующими параметрами:

- размер окна – 4×4 пикселя;
- количество уровней пирамиды – 6 (включая исходный размер изображения);
- максимальное количество итераций – 20;
- количество точек для определения смещения – 81 (равномерная сетка 9×9);
- разрешение изображения – 720×575;
- максимально детектируемое смещение – ± 128 пикселей.

Сначала алгоритм был реализован полностью программно, на одном встроенном ядре Cortex A53. Среднее время работы алгоритма на 1 кадре составило 220 мс. Надо отметить, что поскольку алгоритм Лукаса – Кэнейда итерационный, то на каждой паре кадров определение смещения данным методом занимает различное время. Очевидно, что для обработки изображений в реальном времени такое время обработки одного кадра не приемлемо.

На следующем этапе расчет производных Шарра для изображений, а также масштабирование изображений было реализовано аппаратными блоками и выполнялось независимо от процессорного ядра. В этом случае время вычисления составило 18 мс.

Затем матричные вычисления алгоритма Лукаса – Кэнейда были реализованы 4 одинаковыми аппаратными блоками, что позволило вы-

числять смещения между кадрами одновременно для 4 точек. В этом случае время вычисления оптического потока составило в среднем 7,5 мс. Дальнейшее распараллеливание вычислений оказалось неэффективным, поскольку в процедуре Ньютона – Рафсона присутствуют ветвления, которые эффективнее реализовать программно. При дальнейшем увеличении параллельных веток вычисления процессор не успевал производить программные операции процедуры Ньютона – Рафсона.

Для сравнения с семейством алгоритмов Лукаса – Кэнейда был реализован алгоритм согласованной фильтрации изображения на базе ДДПФ. Было реализовано полностью аппаратное вычисление ДДПФ для зоны поиска фрагмента, для самого фрагмента и обратное ДДПФ для определения положения фрагмента в зоне поиска. В реализации алгоритма использовалось 4 аппаратных блока, позволяющих распараллелить вычисления.

Каждый аппаратный блок был реализован со следующими параметрами:

- размер зоны поиска смещения – 256×256 пикселей;
- размер фрагмента изображения – 8×8 пикселей;
- количество точек для определения смещения – 81 (равномерная сетка 9×9);

В результате время работы алгоритма с указанными параметрами составило 2,9 мс, при этом процессорное время, необходимое для работы алгоритма, составило 0,2 мс. Остальное время аппаратные блоки работали независимо от процессорного ядра.

На рис. 1–3 приведены результаты основных этапов алгоритма.

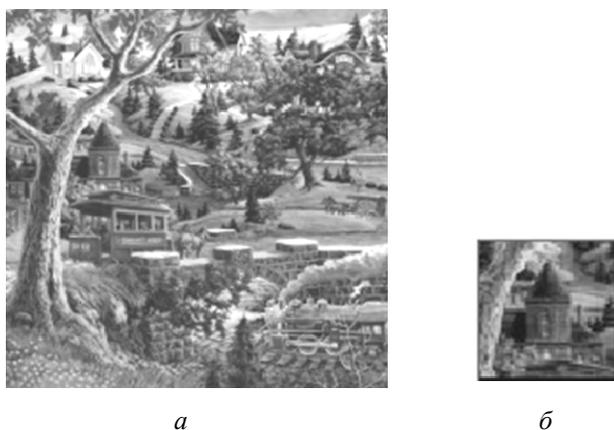


Рис. 1. Изображения: *a* – зоны поиска фрагмента; *б* – искомого фрагмента

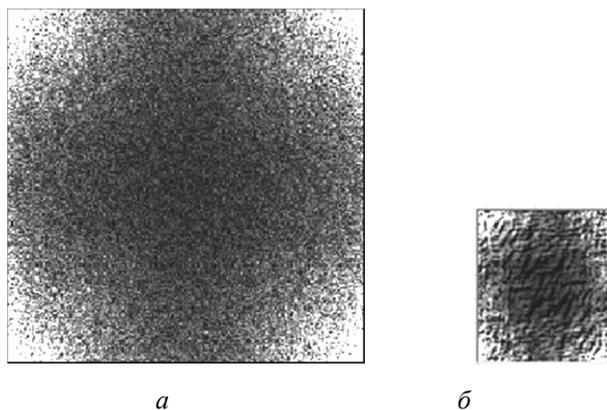


Рис. 2. ДДПФ: *a* – зоны поиска фрагмента; *б* – искомого фрагмента

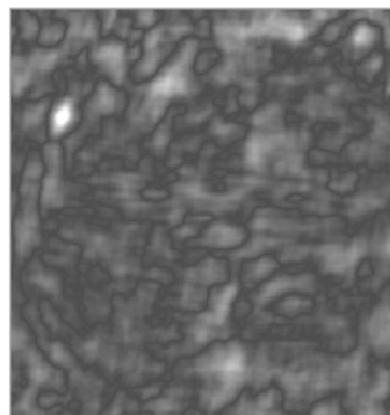


Рис. 3. Результат согласованной фильтрации изображения

На рис. 3 виден пиксель с максимальной яркостью в том месте зоны поиска, где верхний левый угол искомого фрагмента совпадает с фрагментом зоны поиска.

Сравнительная характеристика рассмотренных различных реализаций алгоритмов определения векторов смещения видеоизображений сведена в табл. 1.

Видно, что программная обработка не позволяет реализовать вычисление векторов смещения с указанными параметрами в реальном времени. Аппаратные блоки значительно могут ускорить вычисления в случае с алгоритмом L-K, однако наилучший результат достигается при использовании алгоритма на основе согласованной фильтрации.

Параметры, определяющие эффективность сравниваемых алгоритмов, сведены в табл. 2.

Таблица 1. Сравнительная характеристика реализаций алгоритмов определения векторов смещения видеоизображений

Параметр/реализация	Время вычисления, мс	Количество ресурсов ПЛИС, LUT/ Registers	Количество точек	Зона поиска смещения	Размер фрагмента	Процент занятости процессорного ядра A53
Программная реализация алгоритма L-K на ядре CortexA53 (1,3ГГц)	220	–	81	256×256	4×4	100 %
Программная реализация алгоритма L-K на ядре CortexA53 (1,3ГГц) + аппаратные блоки вычисления производных и масштабирования изображения	18	3580/3510	81	256×256	4×4	100 %
Программная реализация алгоритма L-K на ядре CortexA53 (1,3ГГц) + аппаратные блоки вычисления производных, масштабирования изображения и матричных операций	7,5	9631/9456	81	256×256	4×4	100 %
Программная реализация алгоритма согласованной фильтрации на ядре CortexA53 + специализированные аппаратные блоки ДДПФ	2,9	10736/11226	81	256×256	32×32	<7 %

Таблица 2. Параметры эффективности алгоритмов определения векторов смещения видеоизображений

Параметр	Алгоритм	
	Алгоритм L-K	Согласованная фильтрация
Время работы алгоритма	Не фиксированное, зависит от изображения	Фиксированное
Адаптируемость под архитектуру FPGA	Частично адаптируемый	Возможна работа полностью независимая от процессорного ядра
Эффективность использования контроллера DDR	Не очень эффективное: много чтений малых фрагментов (4–8 байт) по произвольным адресам	Эффективное, чтение большими фрагментами (1024 байта)

Заключение

Наши сравнения показали, что при наличии достаточных аппаратных ресурсов определение смещения соседних кадров видеоряда гораздо эффективнее реализуется на базе согласованной фильтрации, чем алгоритмом L-K.

Библиографические ссылки

1. CHANG T. H. TRACKING MULTIPLE PEOPLE WITH A MULTI-CAMERA SYSTEM / T.H. CHANG, S. GONG // PROCEEDINGS 2001 IEEE WORKSHOP ON MULTI-OBJECT TRACKING. 2001. P. 19-26. DOI:10.1109/MOT.2001.937977.
2. Pan J. Robust Occlusion Handling in Object Tracking / J. Pan, B. Hu // 2007 IEEE Conference on Computer Vision and Pattern Recognition : Minneapolis, MN, 17 - 22 June 2007; [CVPR 2007; proceedings]; Vol. 4.
3. Bewley A. Simple online and realtime tracking / A. Bewley, G. Zongyuan, L. Ott, F. Ramos, B. Uproft // 2016 IEEE International Conference on Image Processing (ICIP 2016), Phoenix, Arizona, USA 25-28 September 2016. Pp. 3464-3468.
4. Волосатова Т. М., Яблоков В. Е. Слежение за выделенными объектами на протяжении видеоряда // Инженерный вестник. 2015. №7. С. 518–531.
5. Sanjay S., Chandra S., Anil V. Real-Time FPGA-Based Object Tracker with Automatic Pan-Tilt Features for Smart Video Surveillance Systems // J. Imaging 2017, 3, 18; doi:10.3390/jimaging3020018
6. Peng G., Ruyue Y., Zhicong L., Linsheng Z., Yan Z. A Novel Low-cost FPGA-based Real-time Object Tracking System // arXiv:1804.05535v2 DOI:10.1109/ASICON.2017.8252560
7. Baoyan Z., Xiaofeng L., Qiaoyuan W., Siqi Z., Qianyun L. A Hardware Architecture of Target Tracking System on FPGA // Proceedings of the 2nd International Conference on Computer Science and Application Engineering 2018. Article No. 65 ISBN: 978-1-4503-6512-3 DOI:10.1145/3207677.3278009.
8. Milan A. MOT 16: a benchmark for multi-object tracking / A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler. // arXiv preprint arXiv:1603.00831 (2016). Pp. 1-12.
9. Bruce D. An Iterative Image Registration Technique with an Application to Stereo Vision / D. Bruce, Lucas and Takeo Kanade // IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence. Vol. 2, 1981. Pp. 674-679.
10. Сакович И. О., Белов Ю. С. Применение метода Лукаса – Канаде для вычисления оптического потока // Инженерный журнал: наука и инновации, 2014, вып. 7. URL: <http://engjournal.ni/catalog/pribor/optica/1275.html>.
11. Прэнт У. Цифровая обработка изображений / пер. с англ. М. : Мир, 1982. Кн. 2. 480 с.

12. Bagni B. Demystifying the Lucas-Kanade Optical Flow Algorithm with Vivado HLS / D. Bagni, P. Kannan, S. Neuendorffer: URL: https://www.xilinx.com/support/documentation/application_notes/xapp1300-lucas-kanade-optical-flow.pdf.

13. Поляк Б. Т. Метод Ньютона и его роль в оптимизации и вычислительной математике // Труды ИСА РАН. 2006. Т. 28. С. 48–66.

14. Цифровая обработка изображений в информационных системах: Учебное пособие / И. С. Грузман, В. С. Киричук, В. П. Косых, Г. И. Перетягин, А. А. Спектор. Новосибирск : Изд-во НГТУ, 2002. 352 с.

15. Прэнтт У. Цифровая обработка изображений / пер. с англ. М. : Мир, 1982. Кн. 2. 480 с.

References

1. CHANG T.H., GONG S. [TRACKING MULTIPLE PEOPLE WITH A MULTI-CAMERA SYSTEM]. *PROCEEDINGS 2001 IEEE WORKSHOP ON MULTI-OBJECT TRACKING*. 2001. PP. 19-26. DOI:10.1109/MOT.2001.937977.

2. Pan J., Hu B. [Robust Occlusion Handling in Object Tracking]. *IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, MN, 2007, vol.4, pp 17-22.

3. Bewley A., Zongyuan G., Ott L., Ramos F., Uppcroft B. [Simple online and realtime tracking]. *IEEE International Conference on Image Processing (ICIP 2016)*. Phoenix, Arizona, USA, 2016, pp. 3464-3468.

4. Volosatova T.M. [Tracking selected objects throughout the video sequence]. *Inzhenernyy vestnik*. 2015, vol. 7, pp. 518-531 (in Russ.).

5. Sanjay S., Chandra S., Anil V. [Real-Time FPGA-Based Object Tracker with Automatic Pan-Tilt Features for Smart Video Surveillance Systems]. *J. Imaging* 2017, 3, 18; DOI:10.3390/jimaging3020018.

6. Peng G., Ruyue Y., Zhicong L., Linsheng Z., Yan Z. [A Novel Low-cost FPGA-based Real-time Object Tracking System]. arXiv:1804.05535v2 (2017) DOI:10.1109/ASICON.2017.8252560.

7. Baoyan Z., Xiaofeng L., Qiaoyuan W., Siqi Z., Qianyun L. [A Hardware Architecture of Target Tracking System on FPGA] *Proceedings of the 2nd International Conference on Computer Science and Application Engineering* 2018. Article No. 65 DOI:10.1145/3207677.3278009.

8. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler [A benchmark for multi-object tracking]. arXiv preprint arXiv:1603.00831 (2016). pp. 1-12.

9. Bruce D., Lucas and Takeo Kanade [An Iterative Image Registration Technique with an Application to Stereo Vision]. [IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence]. 1981, vol. 2, pp. 674-679.

10. Sakovich I.O., Belov Y.S. [An application of the Lucas-Canada method for calculating the optical flux]. *Inzhenernyy zhurnal: nauka i innovacii*. 2014, vol. 7 (in Russ.).

11. Pratt W. *Tsifrovaya obrabotka izobrazhenii* [Digital image processing]. Moscow, Mir Publ., 1982, vol.2, 480 p. (in Russ.).

12. D. Bagni, P. Kannan, S. Neuendorffer [Demystifying the Lucas-Kanade Optical Flow Algorithm with Vivado HLS] Available at: https://www.xilinx.com/support/documentation/application_notes/xapp1300-lucas-kanade-optical-flow.pdf

13. Poljak B.T. *Metod N'jutona i ego rol' v optimizacii i vychislitel'noj matematike* [Newton's method and its role in optimization and computational mathematics]. *Trudy ISA RAN* [Proc. ISA RSA]. 2006, vol. 28, pp. 48-66 (in Russ.).

14. Gruzman I.S., Kirchuk V.S., Kosyh V.P., Peretjagin G.I., Spector A.A. *Cifrovaja obrabotka izobrazhenij v informacionnyh sistemah* [Digital image processing in information systems]. Novosibirsk, NGTU Publ., 2002, 352 p. (in Russ.).

15. Pratt W. *Tsifrovaya obrabotka izobrazhenii* [Digital image processing]. Moscow, Mir Publ., 1982, vol.2, 480 p. (in Russ.).

Comparison of Implementation of Computation of Video Image Displacement Vectors in Built-in Systems

P. A. Ushakov, DSc in Engineering, Professor, Kalashnikov ISTU, Izhevsk, Russia

A. Yu. Pechenkin, PhD in Engineering, Associate Professor, Kalashnikov ISTU, Izhevsk, Russia

Various algorithms of video image displacement vectors calculation are considered. These algorithms are used in smartphones, digital photo cameras, different special-purpose devices in order to stabilize the image, traffic control, air photographic survey, object tracking and other applications. In order to compare the efficiency of hardware resources consumption to perform video images displacement vectors calculation algorithms there were considered: Lucas-Kanade method-based algorithm and the image matched filtering algorithm. Global image displacement was estimated by means of a hardware platform – Ultrascale+ Xilinx SoC. Matrix calculations of the Lucas-Kanade algorithm were performed with four equal hardware blocks, and it made it possible to calculate the displacements of the frames for four points at the same time. The acquired average optical flow calculation time was 7.5 ms. Four hardware blocks were also used to perform parallel calculations to implement the 2D-DFT-based image matched filtering. The resulting operation time of the algorithm was 2.9 ms, and the CPU time used by the algorithm was 0.2 ms. The implementations revealed weak spots of the algorithms, and the required hardware resources of SoPC were determined. The matching filtering algorithm proved to be more efficient as compared to the Lucas-Kanade algorithm to estimate the video sequence consecutive frames displacement upon enough hardware resources.

Keywords: Lucas-Kanade algorithm, feature point tracking, optical flow, image matched filtering.