

21. *Turulin I.I.* Upravlyaemye tsifrovye fil'try [Controlled digital filters]. Taganrog: YuFU, 2009, 260 p. Rus. Dep. v VINITI 18.06.09 № 383-V2009.
22. *Turulin I.I., Bulgakova Yu.I.* Metodika proektirovaniya upravlyaemykh tsifrovyykh fil'trov na osnove metoda bilineynogo preobrazovaniya [The methodology of designing controlled digital filters based on the method of bilinear transformation], *Aviakosmicheskoe priborostroenie* [Aerospace instrumentation], 2011, No. 6, pp. 30-36.
23. *Turulin I.I., Mogheer H.S.* Analysis of Controlled Digital Recursive High-Pass Filters Structures with Infinite Non-Negative Impulse Response, *2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. IEEE, 2022, pp. 755-759.
24. *Mogheer H.S., Turulin I.I.* Reduction of Signal Overshooting Caused by Cutoff Frequency Changing in the Controlled Digital Butterworth Low Pass Filter, *2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. IEEE, 2022, pp. 783-788.
25. *Mogheer Hussein Sh., Turulin I.I.* Analysis of MATLAB System Applicability for Synthesis of Controlled Butterworth Digital Recursive IIR Filters. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2021, No. 3 (220), pp. 72-81.

Статью рекомендовал к опубликованию д.т.н., профессор Н.Н. Чернов.

Турулин Игорь Ильич – Южный федеральный университет; e-mail: iiturulin@sfedu.ru; г. Таганрог, Россия; тел.: +78634371632; кафедра теоретических основ радиотехники; д.т.н.; профессор.

Аль-Карави Хусейн Шукор Мукер – e-mail: alkaravi@sfedu.ru; кафедра теоретических основ радиотехники; аспирант.

Turulin Igor Ilich – Southern Federal University; e-mail: iiturulin@sfedu.ru; Taganrog, Russia; phone: +78634371632; the department of theoretical foundations of radio engineering dr. of eng. sc.; professor.

Hussein Shakor Mogheer – e-mail: alkaravi@sfedu.ru; the department of theoretical foundations of radio engineering; postgraduate student.

УДК 004.67

DOI 10.18522/2311-3103-2022-4-182-192

А.А. Чусов, М.А. Копаева

АЛГОРИТМ ПОТОКОВОЙ КОМПРЕССИИ ДАННЫХ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ ОБЕСПЕЧЕНИЯ НАУЧНЫХ ЭКСПЕРИМЕНТОВ

*Представлен оригинальный алгоритм и метод реализации однопроходной компрессии потока числовых данных с плавающей запятой. Целью работы является разработка и формализация алгоритмического метода однопроходной потоковой компрессии числовых значений с плавающей запятой, обеспечивающего высокую оперативность кодирования и декодирования, поскольку практика применения для этого существующих реализаций показывает их недостаточную оперативность, неприемлемую ресурсоемкость, низкую применимость для потоковой обработки большого объема данных с плавающей запятой в реальном времени. Для достижения этой цели были решены следующие задачи. Описана математическая модель и алгоритм компрессии потока скалярных числовых значений с плавающей запятой, а также результаты проведенного экспериментального исследования созданного метода реализации компрессии структур одномерных и двумерных научных данных. Модель опирается на распространенный метод реализации расширенного множества действительных чисел, отображаемых на числа с плавающей запятой двойной точности, представление которых регламентировано форматом *binary_64* стандарта IEEE-754. Алгоритм может быть использован в составе распределенных высокопроизводительных информационных систем обеспечения научных экспериментов для реализации задач, критичных к производительности*

операций ввода-вывода и инфокоммуникационного обмена данными. Производительность и применимость алгоритма в реализациях потоковой обработки данных обусловлены его однопроходным поведением, относительно низкими требованиями к априорно известному и фиксированному размеру истории, на основе которой построена работа предиктора алгоритма компрессии. Действительно, полученные результаты показывают сравнимую с более ресурсоемкими универсальными кодерами результативность компрессии при значительном росте оперативности. При синхронизации параметров компрессии и декомпрессии векторных данных в предположении взаимной корреляции магнитуд скаляров одного измерения, возможно применение параллелизма класса ОКМД (одна команда-множество данных) для дальнейшего увеличения производительности предиктора (а следовательно, компрессии и декомпрессии), если реализующее устройство обладает возможностью обращения к изменяемой памяти, реализующей историю, на основе сдвиговых значений, формируемых параллельно или векторно, например, с помощью инструкций класса VGATHER микропроцессоров Intel. Реализация алгоритма внедрена авторами в параллельно-распределенную систему моделирования волновых полей с целью снижения издержек, обусловленных операциями ввода-вывода при взаимодействии системы узлов через вычислительную сеть. Экспериментально показана более высокая оперативность компрессии при сравнимой результативности универсальных многопроходных кодеров RAR, ZIP и 7Z.

Компрессионное кодирование; кодирование источника; арифметика с плавающей запятой; потоки данных.

A.A. Chusov, M.A. Kopaeva

ALGORITHM FOR COMPRESSION OF FLOATING-POINT DATA IN SCIENTIFIC RESEARCH SUPPORT SYSTEMS

The paper presents an original algorithm and its implementation for single pass real-time compression of streams of numeric floating-point data. The purpose of the research is to develop and formalize a single-pass algorithm of stream floating-point data compression in order to increase performance of both encoding and decoding, because a use of existing implementations provides insufficient speed of compression, are too restrictive on hardware resources and limited in applicability to real-time stream data compression when it comes to floating-point data. For that, the following issues have been addressed. The developed mathematical model and the algorithm for compression of scalar floating-point data are described together with results of experimental research of the compression method applied to single-dimensional and two-dimensional scientific data. The model is based upon the commonly-used binary₆₄ representation, of the IEEE-754 standard, onto which extended real-line values are mapped. The algorithm can be implemented as part of high-performance distributed systems in which performance of input-output operations, as well as internetwork communication, are critical to overall efficiency. The performance and applicability of the algorithm in data stream compression result from its single-pass behaviour, relatively low requirements to a priori known and statically defined size of memory required to implement history of compression, which the predictor, used in compression and decompression, is based on. Indeed, the measured compression ratios are comparable to ones which result from using more resource-intensive universal coders but providing significantly lower latency. Provided synchronization of parameters of both compressor and decompressor applied to a stream of vector values and assuming a correlation between absolute values of scalars of the same dimension within the vectors, further improvement of the predictor performance can be attained by means of SIMD-class parallelism which, in turn, is beneficial for overall performance of compression and decompression, provided that the underlying hardware is capable of addressing random-access memory based on offsets in a vector register, such as by employment of the VGATHER class instructions of Intel processors. In order to reduce the bottlenecks associated with input-output, an implementation of the algorithm is employed by the authors as part of a computing system used for parallel simulation of wave fields which is distributed via a network. The experiments described in the paper demonstrate significant performance increase of the proposed coder compared to well-known universal compressors, RAR, ZIP and 7Z, while the achieved compression factors remain comparable.

Data compression; source compression coding; floating-point arithmetic; data streams.

Введение. При обработке экспериментальных научных данных, представленных в формате чисел с плавающей запятой, их передача в несжатом виде может иметь неприемлемо низкую производительность, например, в системах реального времени. Однако универсальные алгоритмы компрессии направлены на увеличение результативности сжатия, а не оперативности, поэтому в случаях, когда предъявляются требования к оперативности или энергоэффективности инфокоммуникационных систем, алгоритмы необходимо специализировать. Для сжатия данных с плавающей запятой без потерь существуют методы сжатия на основе прогнозирования. Такие методы рассматривают число с плавающей запятой как целое число той же разрядности и предсказывают целочисленное значение, используя целочисленную операцию. Эти алгоритмы вычисляют разность или результат логической операции «исключающее ИЛИ» между предсказанным и исходным значением в каждом наборе данных для кодирования в более короткие битовые строки, чем исходные.

Первыми статьями, в которых были представлены основанные на прогнозировании алгоритмы сжатия научных данных с плавающей запятой, были [1–3]. В [1] разработали метод сжатия без потерь для данных временных рядов с плавающей запятой с использованием дельта-сжатия, и также объясняется проблема с представлением в памяти данных с плавающей запятой: даже если десятичное представление двух чисел может быть очень похожим, это не обязательно будет так для машинного представления в памяти [4, 5]. Подход, предложенный в [2], сжимает любой линейный поток данных, создавая хеш-ключ на основе различий нескольких последних выборок с поиском повторяющихся шаблонов в данных. В [6] предлагается использовать комбинированный интер-экстраполяционный предиктор: экстраполяция оценивает будущие данные, исходя из последовательности прошлых данных, а интерполяция конструирует соответствующие данные как часть последовательности данных. Существует схема, которая содержит два предиктора [7]. Два спрогнозированных значения сравниваются с действительным, и для выполнения последующей операции «исключающее ИЛИ» выбирается тот прогноз, который ближе к действительному значению. В [8] приводится многопроходный алгоритм сжатия без потерь $ndzip$ для многомерных регулярных сеток данных с плавающей запятой одинарной и двойной точности. Алгоритм разработан для реализации на многоядерных процессорах с поддержкой ОКМД (одиночный поток команд, множественный поток данных) [9]. В области сжатия данных с плавающей запятой без потерь, существуют исследования аппаратных реализаций [10–15]. В [6] предложили аппаратную реализацию сжатия данных с плавающей запятой двойной точности путем группирования экспонент. В ряде работ [16–18] проведены исследования по синтезу алгоритмов компрессии.

Ниже представлен разработанный авторами настоящей работы алгоритм сжатия данных без потерь на основе прогнозирования с использованием хеш-таблицы.

Математическая модель кодера источника. Абстрактно работа кодеров с предикторами может быть сформулирована в виде функциональной диаграммы, представленной на рис. 1 [19].

В рамках настоящей работы предлагаются новые методы прогнозирования, структурирования и использования хеш-таблицы.

На рис. 2 отображены блок-схемы созданных алгоритмов компрессии и декомпрессии. На вход реализации компрессии поступает число с плавающей запятой двойной точности D . Далее функцией $hash$ производится вычисление хеш на основе имеющихся трех последних разностей, хранимых кодером.



Рис. 1. Функциональная диаграмма алгоритмов компрессии на основе предикторов

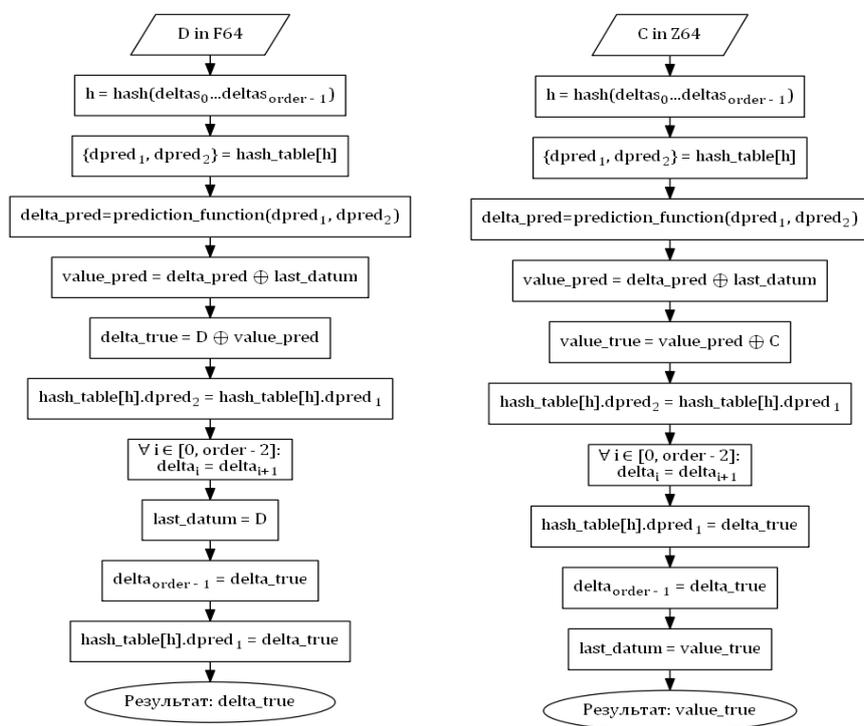


Рис. 2. Блок-схема алгоритмов предсказания при компрессии (слева) и декомпрессии (справа)

Блок-схема вычисления ключа h хеш-таблицы «hash_table» осуществляется на основе нескольких последних вычисленных полиномиальных (в кольце $\mathbb{Z}_2[x]$, то есть посредством операции «исключающее ИЛИ») разностей между предсказанными и вычисленными значениями. На рис. 2 эти разности обозначены переменными $delta_i$, где i – элемент индексного множества $[0, order - 1]$, а число этих разностей, в реализации принятое равным трем, определяет порядок предиктора. Таким образом предсказание осуществляется путем поиска истории процесса компрессии (или, соответственно, декомпрессии) на основе числа, равного порядку кодера, последних обработанных элементов данных.

Это достигается путем вычисления ключа хеш-таблицы, содержащей вероятные полиномиальные разности между истинными и предсказанными значениями. Тогда на этой основе и делается прогноз, который совпадает с последним ранее записанным в таблицу (по этому же ключу) значением. Реализуемость функции вычисления хеш-ключа обусловлена размером хеш-таблицы, который определяет разрядность ключа, а также тем, насколько вероятным должен быть успех поиска существующего элемента хеш-таблицы, так как в пределе, эта вероятность падает экспоненциально с увеличением порядка компрессора. В этом случае компрессор становится эквивалентным компрессору на основе последней, предыдущей, полиномиальной разности (в предположении, что хеш-таблица проинициализирована нулями). Однако вероятность успешного поиска может быть увеличена путем выбора в процессе вычисления ключа лишь наименее часто изменяемых старших бит числа с плавающей запятой.

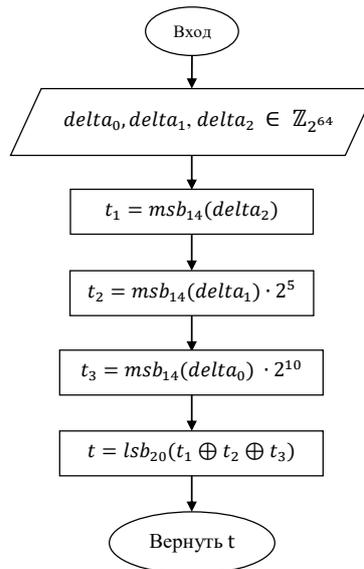


Рис. 3. Блок-схема вычисления hash

По этим причинам были получены удовлетворительные параметры функции $hash$, которая приведена на рис. 3. На вход поступают три целочисленные полиномиальные разности $delta_i$. Для вычислений используются старшие 14 бит каждой разности, возвращаемые функцией msb (most significant bits) как единый целочисленный скаляр. Самая последняя (недавняя) разность $delta_2$ не подвергается операции побитового сдвига. Разности $delta_1$ и $delta_0$ подвергаются операции побитового сдвига влево на 5 бит и 10 бит соответственно. Далее результаты по-

линомиально складываются, и 20 младших бит суммы, реализуемые на рис. 3 функцией *lsb* – least significant bits, возвращаются в качестве значения функции *hash* и ключа хеш-таблицы.

Использование двадцати битового ключа дает возможность реализовать таблицу в виде одномерного массива с 2^{20} элементами. Значением элемента является пара значений $dpred_1$ и $dpred_2$, которые являются последними шестидесяти четырёх битными полиномиальными разностями, изначально равными нулю либо полученными ранее для того же значения хеш-ключа.

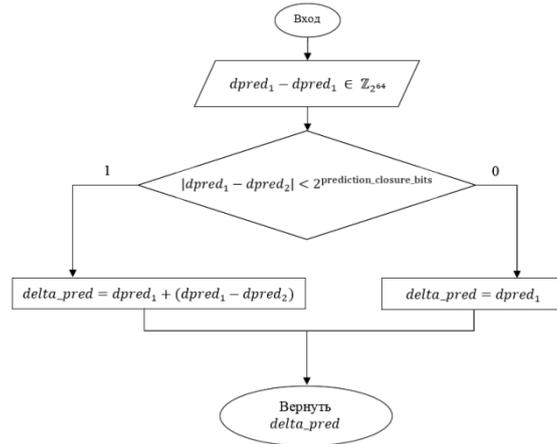


Рис. 4. Блок-схема функции *prediction_function*

В более простом варианте алгоритма в качестве предсказанного значения достаточно использование лишь одного элемента истории в таблицы. Однако лучшим себя показал модифицированный алгоритм (рис. 4) предсказания на основе двух элементов истории, если они достаточно похожи. В реализации алгоритм представлен функцией *prediction_function*, которая возвращает $dpred_1$, если два её аргумента – $dpred_1$ и $dpred_2$ – имеют менее заданного (управляющим параметром *prediction_closure_bits* кодера) количества одинаковых старших бит. Если же количество совпадающих старших бит больше либо равно этому значению, в качестве предсказанной разницы функция *prediction_function* возвращает значение $dpred_1 + (dpred_1 - dpred_2)$.

Выбор значения параметра *prediction_closure_bits* апостериорно обусловлен результативностью сжатия: для выбранных наборов данных экспериментально полученный оптимум составляет 14 бит.

На следующем этапе работы алгоритма компрессии производится обновление значений в хеш-таблице, соответствующих рассчитанному хеш. Также обновляются значения разностей $delta_i$.

Предшествующий элемент сжимаемых данных *last_datum* полиномиально суммируется с результатом операции *prediction_function*, и таким образом рассчитывается прогнозируемое значение *pred*. Результатом предсказания при декомпрессии является полиномиальная разница *delta_true* между прогнозом *pred* и истинным значением *D*. Результатом компрессии числа с плавающей точкой двойной точности – четырехбитовое число ведущих бинарных нулей в этой разнице, деленное на четыре, и оставшиеся младшие биты разницы *delta_true* в исходном виде.

Обратимость компрессии обусловлена когерентностью состояния декодера состоянию кодера при обработке каждого элемента данных и соответствующего ему кода переменной длины на стороне декодера. Эта когерентность в свою очередь наглядно наблюдается на рис. 1: действительно предсказанное значение $value_pred$ совпадает на кодере и декодере, поскольку оно рассчитано в результате выполнения идентичных операций, и в свою очередь это же значение детерминировано определяет состояние кодера и декодера при обработке последующих элементов данных.

Эксперименты. На рис. 5 и 6 отображены результаты проведенного экспериментального исследования. Для проведения исследования использованы файлы размером от 16 байт до 256 Мбайт, содержащие, с различными параметрами, одномерную линейную функцию, синусоиду, а также двумерную матрицу значений $\Re(Me^{2\pi i f d - \alpha d})$ (где M – магнитуа, f – пространственная частота, d расстояние до точки, которой соответствует выбранный элемент матрицы, α – коэффициент пространственного затухания, $i^2 = -1$).

На рис. 5 приводится сравнение работы созданного специального алгоритма компрессии с работой таких общепринятых реализаций универсального сжатия как 7Z и ZIP для сжатия синусоиды. ZIP использован с аргументом «-9», задающий максимальное сжатие. В качестве показателя оценки качества работы алгоритмов сжатия используется коэффициент сжатия [20], который устанавливает соотношение объема исходных к объему сжатых данных. Коэффициенты сжатия разработанного алгоритма компрессии варьируются от 1 до 1,6. Рис. 5 сравнивает результативности трех реализаций компрессии.

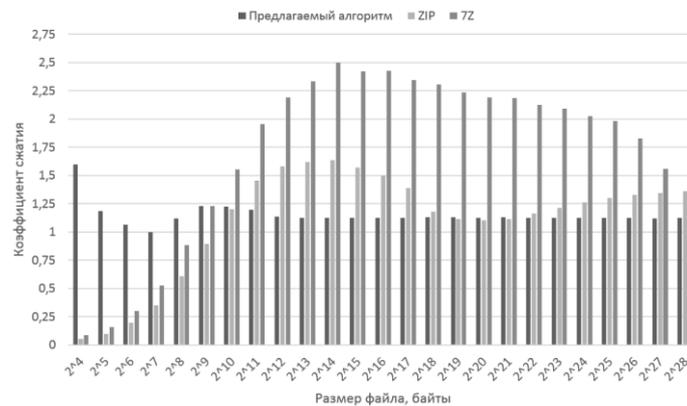


Рис. 5. Зависимость коэффициентов сжатия от размеров файлов для трех методов компрессии синусоидального сигнала

Однако в системах передачи данных, а также в системах с требованиями к оперативности информационного обмена в условиях низких пропускных способностей каналов связи или большого объема передаваемых данных необходимо обеспечение вычислительной эффективности компрессии и декомпрессии, которая определяется временными и связанными энергетическими затратами на обработку данных.

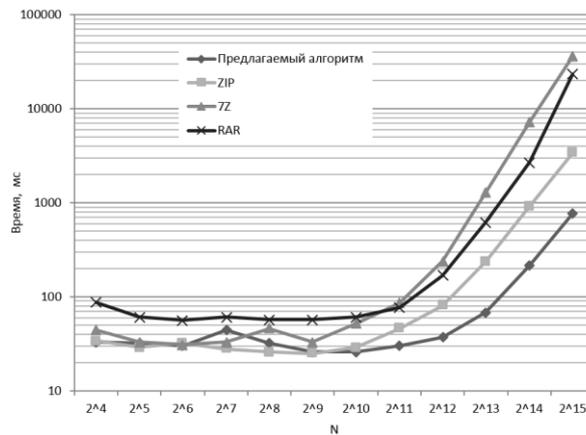


Рис. 6. Оперативность сжатия, измерение которой производилось на высокопроизводительном компьютере, разработанного алгоритма компрессии в сравнении с алгоритмами компрессии 7Z, ZIP, RAR в случае сжатия двумерных данных, из которых составлена квадратная матрица $N \times N$

Поэтому отдельно исследовались возможности по обеспечению представленным однопроходным алгоритмом требований к оперативности. На рис. 6 изображены полученные значения времени сжатия двумерных данных, описанных выше, в зависимости от размеров файлов для четырех методов компрессии. Измерение времени сжатия проводилось на высокопроизводительном компьютере с двумя 24-ядерными центральными процессорами Intel Xeon X12 E5-2695v2 и 128 Гбайт оперативной памяти под управлением операционной системой Microsoft Windows Server 2012 R2. При размерах файла от 32 байт до 128 Мбайт измеренная оперативность разработанного алгоритма варьируется от 26 мс до 763 мс и сравнительно с другими алгоритмами является наиболее высокой. 7Z при обработке больших файлов оказался наиболее медленным. Работа компрессоров ZIP и RAR заняла меньше времени по сравнению с 7Z, но больше, чем работа разработанного алгоритма.

Заключение. Результативность алгоритма, определяемая коэффициентом сжатия, превзошла результативность распространенной программы сжатия данных ZIP, однако уступила реализациям алгоритмов RAR и 7Z.

Вместе с тем наблюдается значительный прирост оперативности компрессии и декомпрессии созданного алгоритма, который является однопроходным, что дает возможность применять его для потоковой обработки данных в реальном времени, например, при передаче результатов измерений с удаленных датчиков, при реализации высокопроизводительного распределенного численного компьютерного моделирования физических процессов и явлений, математические модели которых опираются на вещественные и комплексные числовые множества, которые с достаточной достоверностью аппроксимируются числами с плавающей запятой.

Реализация функций предиктора в виде автомата с конечным состоянием и простота операций синтеза битовых кодов переменной длины позволила авторам работы инкапсулировать реализацию в функции обработки полиморфных классов, моделирующих потоки данных, и, во-первых, добиться указанных в настоящей работе результативности и оперативности компрессии, а также внедрить его в более составную систему имитационного моделирования волновых векторных полей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Engelson V., Fritzon D., Fritzon P.* Lossless compression of high-volume numerical data from simulations // In Data Compression Conference. – 2000. – P. 574-586.
2. *Ratanaworabhan P., Ke J., Burtcher M.* Fast lossless compression of scientific floating-point data // In Proceedings of Data Compression Conference. – 2006. – P. 133-142.
3. *Lindstrom P., Isenburg M.* Fast and efficient compression of floating-point data // IEEE Transactions on Visual and Computer Graphics. – 2006. – Vol. 12, No. 5. – P. 1245-1250.
4. IEEE 754: Standard for binary floating-point arithmetic, 2008.
5. *Gomez L.A., Cappello F.* Improving floating point compression through binary masks // In Proc. 2013 IEEE Int. Conf. Big Data. – 2013. – P. 326-331.
6. *Imai S., Fukuma S., Mori S.* A Floating Point Data Compression Using Inter-Extrapolative Predictor // IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS). – 2018. – P. 546-549.
7. *Burtcher M., Ratanaworabhan P.* FPC: A High-Speed Compressor for Double-Precision Floating-Point Data // IEEE transactions on computers. – 2009. – Vol. 58, No. 1. – P. 18-31.
8. *Knorr F., Thoman P., Fahringer T.* ndzip: A High-Throughput Parallel Lossless Compressor for Scientific Data // 2021 Data Compression Conference (DCC). – 2021. – P. 103-112.
9. *Hildebrandt J., Habich D. and Lehner W.* BOUNCE: Memory-Efficient SIMD Approach for Lightweight Integer Compression // 2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW). – 2022. – P. 123-128. – DOI: 10.1109/ICDEW55742.2022.00025.
10. *Tomari H., Inaba M., Hiraki K.* Compressing floating-point number stream for numerical applications // 2010 First International Conference on Networking and Computing. – 2010. – P. 112-119.
11. *Katahira K., Sano K., Yamamoto S.* FPGA-based lossless compressors of floating-point data streams to enhance memory bandwidth // In Proceedings of the International Conference on Application-specific Systems, Architectures and Processors. – 2010. – P. 246-253.
12. *Mondigo A., Ueno T., Tanaka D., Sano K., Yamamoto S.* Design and scalability analysis of bandwidth-compressed stream computing with multiple fpgas // In Proceedings of 2017 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC). – 2017. – P. 108-115.
13. *Ueno T., Kono Y., Sano K., Yamamoto S.* Parameterized Design and Evaluation of Bandwidth Compressor for Floating-Point Data Streams in FPGA-Based Custom Computing. – Berlin, Heidelberg: Springer Berlin Heidelberg. 2013. – P. 90-102.
14. *Ueno T., Sano K., Furusawa T.* Performance Analysis of Hardware-Based Numerical Data Compression on Various Data Formats // 2018 Data Compression Conference. – 2018. – P. 345-354.
15. *Ueno T., Sano K., Yamamoto S.* Bandwidth Compression of Floating-Point Numerical Data Streams for FPGA-Based High-Performance Computing // ACM Transactions on Reconfigurable Technology and Systems. – 2017. – Vol. 10, No. 3. – P. 1-22.
16. *Yang A., Mukka H., Hesaaraki F., Burtcher M.* MPC: A Massively Parallel Compression Algorithm for Scientific Data // IEEE International Conference on Cluster Computing. – 2015.
17. *Claggett S., Azimi S., Burtcher M.* SPDP: An Automatically Synthesized Lossless Compression Algorithm for Floating-Point Data // Data Compression Conference. – 2018.
18. *Burtcher M., Hesaaraki F., Mukka H., Yang A.* Real-Time Synthesis of Compression Algorithms for Scientific Data // ACM/IEEE International Conference for High-Performance Computing, Networking, Storage and Analysis. – 2016. – P. 264-275.
19. *Konaeva M.A., Чусов А.А.* Алгоритм и системная реализация компрессии потоков чисел с плавающей точкой при реализации обработки экспериментальных научных данных // Радиоэлектроника. Проблемы и перспективы развития: Сб. трудов Седьмой всероссийской молодежной научной конференции. – Тамбов: Изд. центр ФГБОУ ВО «ТГТУ», 2022.
20. *Ainsworth M., Klasky S., Whitney B.* Compression Using Lossless Decimation: Analysis and Application // SIAM J. Sci. Comput. – 2017. – Vol. 39 (4). – P. B732-B757.

REFERENCES

1. *Engelson V., Fritzson D., Fritzson P.* Lossless compression of high-volume numerical data from simulations, *In Data Compression Conference*, 2000, pp. 574-586.
2. *Ratanaworabhan P., Ke J., Burtscher M.* Fast lossless compression of scientific floating-point data, *In Proceedings of Data Compression Conference*, 2006, pp. 133-142.
3. *Lindstrom P., Isenburg M.* Fast and efficient compression of floating-point data, *IEEE Transactions on Visual and Computer Graphics*, 2006, Vol. 12, No. 5, pp. 1245-1250.
4. IEEE 754: Standard for binary floating-point arithmetic, 2008.
5. *Gomez L.A., Cappello F.* Improving floating point compression through binary masks, *In Proc. 2013 IEEE Int. Conf. Big Data*, 2013, pp. 326-331.
6. *Imai S., Fukuma S., Mori S.* A Floating Point Data Compression Using Inter-Extrapolative Predictor, *IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2018, pp. 546-549.
7. *Burtscher M., Ratanaworabhan P.* FPC: A High-Speed Compressor for Double-Precision Floating-Point Data, *IEEE transactions on computers*, 2009, Vol. 58, No. 1, pp. 18-31.
8. *Knorr F., Thoman P., Fahringer T.* ndzip: A High-Throughput Parallel Lossless Compressor for Scientific Data, *2021 Data Compression Conference (DCC)*, 2021, pp. 103-112.
9. *Hildebrandt J., Habich D. and Lehner W.* BOUNCE: Memory-Efficient SIMD Approach for Lightweight Integer Compression, *2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*, 2022, pp. 123-128. DOI: 10.1109/ICDEW55742.2022.00025.
10. *Tomari H., Inaba M., Hiraki K.* Compressing floating-point number stream for numerical applications, *2010 First International Conference on Networking and Computing*, 2010, pp. 112-119.
11. *Katahira K., Sano K., Yamamoto S.* FPGA-based lossless compressors of floating-point data streams to enhance memory bandwidth, *In Proceedings of the International Conference on Application-specific Systems, Architectures and Processors*, 2010, pp. 246-253.
12. *Mondigo A., Ueno T., Tanaka D., Sano K., Yamamoto S.* Design and scalability analysis of bandwidth-compressed stream computing with multiple fpgas, *In Proceedings of 2017 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2017, pp. 108-115.
13. *Ueno T., Kono Y., Sano K., Yamamoto S.* Parameterized Design and Evaluation of Bandwidth Compressor for Floating-Point Data Streams in FPGA-Based Custom Computing. Berlin, Heidelberg: Springer Berlin Heidelberg. 2013, pp. 90-102.
14. *Ueno T., Sano K., Furusawa T.* Performance Analysis of Hardware-Based Numerical Data Compression on Various Data Formats, *2018 Data Compression Conference*, 2018, pp. 345-354.
15. *Ueno T., Sano K., Yamamoto S.* Bandwidth Compression of Floating-Point Numerical Data Streams for FPGA-Based High-Performance Computing, *ACM Transactions on Reconfigurable Technology and Systems*, 2017, Vol. 10, No. 3, pp. 1-22.
16. *Yang A., Mukka H., Hesaaraki F., Burtscher M.* MPC: A Massively Parallel Compression Algorithm for Scientific Data, *IEEE International Conference on Cluster Computing*, 2015.
17. *Claggett S., Azimi S., Burtscher M.* SPDP: An Automatically Synthesized Lossless Compression Algorithm for Floating-Point Data, *Data Compression Conference*, 2018.
18. *Burtscher M., Hesaaraki F., Mukka H., Yang A.* Real-Time Synthesis of Compression Algorithms for Scientific Data, *ACM/IEEE International Conference for High-Performance Computing, Networking, Storage and Analysis*, 2016, pp. 264-275.
19. *Kopaeva M.A., Chusov A.A.* Алгоритм и системная реализация компрессии потоков чисел с плавающей точкой при реализации обработки экспериментальных научных данных [An algorithm and systematic approach to compression of floating-point data streams for processing of scientific data], *Radioelektronika. Problemy i perspektivy razvitiya: Sb. trudov Sed'moy vserossiyskoy molodezhnoy nauchnoy konferentsii* [Radioelectronics. Problems and future developments: Seventh Russian Youth Scientific Conference]. Tambov: Izd. tsentr FGBOU VO «TGTU», 2022.
20. *Ainsworth M., Klasky S., Whitney B.* Compression Using Lossless Decimation: Analysis and Application, *SIAM J. Sci. Comput.*, 2017, Vol. 39 (4), pp. B732-B757.

Статью рекомендовала к опубликованию д.т.н., г.н.с. Н.В. Злобина.

Чусов Андрей Александрович – ФГАОУ ВО Дальневосточный федеральный университет; e-mail: chusov.aa@dvfu.ru; г. Владивосток, Россия; тел.: +79147315896; доцент департамента электроники, телекоммуникации и приборостроения; к.т.н.

Кобаева Маргарита Александровна – e-mail: rikopae@yandex.ru; тел.: +79502982261; магистрант департамента электроники, телекоммуникации и приборостроения.

Chusov Andrei Aleksandrovich – Far-Eastern Federal University, e-mail: chusov.aa@dvfu.ru; Vladivostok, Russia; phone: +79147315896; associate professor of the department of electronics, telecommunication and hardware development; cand. of eng. sc.

Кобаева Margarita Aleksandrovna – e-mail: rikopae@yandex.ru; phone: +79502982261; graduate student at the department of electronics, telecommunication and hardware development.

УДК 004.89

DOI 10.18522/2311-3103-2022-4-192-200

В.А. Частикова, С.А. Жерлицын**НЕЙРОСЕТЕВАЯ МЕТОДИКА ИДЕНТИФИКАЦИИ ЛИЧНОСТИ
ПО РИСУНКУ ВЕН ЛАДОНИ**

Описывается работа по созданию нейросетевой методики идентификации личности, основанной на механизме сканирования и анализа рисунка вен ладони, как биометрического параметра. В рамках проведенного исследования описаны предпосылки, цели и причины, по которым разработка надежной системы биометрической идентификации является важным и актуальным направлением деятельности. Сформулирован ряд проблем, присутствующих существующим методам решения поставленной задачи: графовому методу и методу, основанному на вычислении расстояния, выраженного в различных интервальных метриках. Приведено описание принципов их работы. Сформулированы задачи, решаемые системами идентификации личности: сопоставление субъекта идентификации с его идентификатором, однозначно идентифицирующим этого субъекта в информационной системе. Описан механизм считывания рисунка вен с ладони, разработанный для анализа изображения, полученного с восприимчивой к излучению инфракрасного диапазона цифровой камеры. При нахождении в кадре ладони, подсвечиваемой светом ближнего ИК-диапазона, на изображении, полученном с камеры, становится заметен рисунок пролегающих под кожным покровом вен, сосудов и капилляров. В зависимости от организации, система идентификации может на основе предоставленного идентификатора определять соответствующий субъект доступа или проверять принадлежность того же идентификатора предполагаемому субъекту. Приведены 3 метода дальнейшего анализа биометрических данных и идентификации личности: подходы, основанные на категориальной классификации и бинарной классификации, а также комбинированный подход, при котором сначала используется идентификация по первому способу, а затем, по второму, но уже для известного идентификатора доступа, определённого на первом этапе. Приведена результирующая архитектура нейросети для категориальной классификации рисунка вен, описан способ вычисления количества параметров модели в зависимости от числа зарегистрированных субъектов. Представлены основные выводы и экспериментальные замеры точности работы системы при реализации различных методов, а также диаграммы изменения точности моделей во время обучения. Выявлены основные преимущества и недостатки приведённых методов.

Биометрическая идентификация личности; рисунок вен ладони; сверточная нейронная сеть; бинарная классификация; категориальная классификация; информационная безопасность.