DOI: https://doi.org/10.26642/ten-2025-2(96)-117-127
UDC 004.932.2:004.272.2:004.272.32

Yu.S. Pohuliaiev, Postgraduate Student
K.S. Smelyakov, Doctor of Science, Professor
Kharkiv National University of Radio Electronics

Software development technology for high-performance biometric data comparators using selective SIMD vectorization

This research introduces a systematic software development methodology for high-performance biometric data comparators grounded in selective SIMD vectorization principles. The technology solves the important problem of choosing the best optimization strategy by creating a theoretical framework that tells you when and where to use vectorization instead of regular parallelization at different stages of an algorithm. The technology combines the Roofline model for computational limits and the Universal Scalability Law for testing parallel systems to give developers efficiency taxonomies and decision criteria for whether or not to use vectorization in biometric systems that process millions of samples. The technology validation utilized a four-stage minutiae-based fingerprint comparison algorithm, executed through various computational methods: traditional thread-based parallelization, SIMD vectorization utilizing SSE and AVX instructions, and hybrid methodologies that integrate both techniques. The experimental validation employed Intel Core i9 architecture alongside the FVC 2000 database, which comprises 80 fingerprint images and 4,421 minutiae points, utilizing stringent statistical methodologies across multiple iterations. The algorithm includes extracting minutiae from centroid squares, calculating local Euclidean structures, analyzing geometric distances and angles, and matching globally with displacement calculations. The results showed unexpected results that went against common ideas about optimization. The hybrid implementation showed consistent performance improvements in all cases, with a 2.66 times speedup with 32 threads and a 3.40 times improvement with complete datasets. This was expected because full SIMD vectorization didn't work as well as traditional parallelization because of high algorithmic interdependencies. Profiling analysis showed that the parallel and hybrid versions had the same computational stages. This means that the performance gains came from secondary optimization effects, such as better memory alignment, better cache use through uniform dictionary distribution, and compiler-induced optimizations during object field access. The research creates a technology framework for selective vectorization in biometric applications. This gives software developers a systematic way to make high-performance identification systems that can be used in real-world situations where they need to be used on a large scale.

Keywords: *biometric identification; vectorization; SIMD; parallelization; performance; AVX; SSE; software development technology.*

Topic relevance. Modern biometric identification systems have high performance standards for biometric data matching systems, especially when they are working with databases that have millions of samples. Horizontal scaling, which means running more threads at the same time, is a common method used in the development of biometric identification software. The applicability of vectorization technology in this domain remains inadequately explored. Vectorization technologies have many benefits, such as speeding up processing by handling more data at once. However, when they are used in real life, they often produce unexpected results. Speeding up thread-based scaling isn't always possible, especially when tasks have a lot of objects that depend on each other. These factors are the main reasons why we need to talk about technology for developing software that is computationally efficient for biometric template identification.

Modern deployment scenarios show how important it is to optimize performance in biometric systems. International airports' border control systems handle thousands of passengers every hour and need to respond to identification requests in less than two seconds per query against databases with more than ten million records. Financial institutions that use biometric authentication to approve transactions face similar problems, as delays in processing directly affect the customer experience and operational costs. Forensic analysts in law enforcement need to quickly match partial or degraded fingerprints against national criminal databases with millions of records. These real-world needs put a lot of pressure on software development teams to come up with solutions that make the most of the existing hardware infrastructure's computing power. The economic aspect is also important: improving algorithms to optimize performance can cut server infrastructure needs by two to three times, which can save a lot of money on both capital and operating costs compared to horizontal scaling by buying more hardware.

From a software engineering point of view, building systems that work so well requires a systematic approach to technology instead of trying to optimize things on the fly. Developers have to make strategic choices between different ways to optimize: thread-based parallelization, SIMD vectorization with SSE and AVX instruction sets,

hybrid methods that use both techniques, and GPU acceleration. The lack of systematic methods for choosing an optimization strategy often results in suboptimal implementations. This is because developers either use vectorization indiscriminately at all stages of an algorithm or don't use it at all because they are unsure of the results and the implementation is too complicated. This research fills this methodological gap by suggesting a technology framework for selective vectorization that gives developers theoretical foundations, efficiency taxonomies, and practical decision criteria to help them decide which algorithmic stages would benefit from SIMD instructions and which should use traditional parallelization. This methodical approach is especially useful for software development teams that work with computationally intensive biometric algorithms where performance is very important but there aren't enough resources, time, or specialized knowledge to make improvements.

Analysis of recent researches. Deep neural networks continue to advance biometric identification capabilities through innovative architectural designs. Hybrid approaches that combine convolutional neural networks with conventional algorithms demonstrate superior accuracy when processing low-quality fingerprints [1, 2]. Recent investigations explore several promising directions: integration of fingerprint domain knowledge into neural architectures [3, 4], application of transformer models for feature extraction [5], and development of unified networks that perform recognition tasks directly without intermediate processing stages [6]. These studies collectively establish that hybrid methodologies integrating domain expertise with machine learning outperform purely data-driven approaches in handling noise, distortions, and fingerprint variations.

Parallel processing architectures have been extensively studied for fingerprint matching acceleration. Research on high-performance computing systems reveals fundamental scalability limitations, with performance plateaus emerging at specific thread counts regardless of parallelization strategy [7]. GPU-accelerated implementations achieve substantial speedups compared to CPU-based solutions, but their effectiveness depends critically on memory access pattern optimization and efficient data transfer between host and device [8]. These findings emphasize that architectural considerations, including network topology and data movement patterns, constitute primary determinants of scalable implementation success rather than mere computational throughput.

SIMD vectorization presents complex optimization challenges at the microarchitectural level. Studies show that intensive use of vector instructions can trigger processor frequency management mechanisms, creating performance effects that extend beyond the computational operations themselves [9–11]. These microarchitectural interactions introduce substantial complexity into optimization strategies for biometric algorithms. Nevertheless, modern vectorization techniques demonstrate significant potential: through careful combination of SIMD instructions, data prefetching, and pipeline optimization, general-purpose processors can approach ASIC-level performance, substantially narrowing the gap between specialized and commodity hardware [12].

Ukrainian research makes important contributions to both biometric systems and high-performance computing methodologies. Studies address biometric identification algorithms including pattern recognition and fingerprint matching technologies [13]. Research on SIMD vectorization for x86-compatible processors provides practical frameworks for instruction-level parallelism optimization [14]. Multilevel parallel computing models demonstrate effective integration of MIMD and SIMD paradigms [15], while formal methodologies establish rigorous approaches to loop parallelization on GPU accelerators [16]. Comparative analyses of software versus hardware acceleration in digital signal processing offer valuable insights for hardware-software co-design optimization strategies [17].

Literature analysis identifies critical gaps in current understanding of biometric system optimization. Existing research predominantly addresses individual component optimization without examining systemic interactions. The integration of neural network approaches with traditional algorithms, particularly considering microarchitectural optimization constraints, remains insufficiently explored. Current methodologies lack comprehensive frameworks for analyzing processor component interactions during algorithm development. Performance models typically focus on discrete operations without accounting for microarchitectural resource contention or interaction effects between optimization strategies under varying processor states. Most critically, the literature provides insufficient guidance for practitioners: systematic technology frameworks that enable software developers to make informed, context-specific decisions about selective optimization strategy application across different algorithmic stages are notably absent. This creates a significant methodological gap between theoretical understanding of optimization techniques and practical requirements of high-performance biometric system development.

The objective of the article is to create and test a systematic software development technology for high-performance biometric comparators that uses selective SIMD vectorization principles. This technology fills the methodological gap in choosing optimization strategies by giving developers theoretical foundations, decision criteria, and practical advice on when and where to use vectorization instead of traditional parallelization. The goal of this research is to solve the following problems:

1. Creation of a theoretical framework that integrates parallel computation models and vectorization efficiency taxonomies relevant to biometric algorithms exhibiting diverse levels of data interdependency;

2. Assessment of the proposed theoretical models' relevance to multi-stage biometric sample identification algorithms via operational intensity analysis and scalability evaluation;

3. Creation of working hypotheses concerning the most effective optimization strategies for various algorithmic stages, grounded in theoretical predictions and efficiency benchmarks;

4. Thorough experimental validation of all computational models across diverse thread counts and dataset dimensions to confirm theoretical predictions and uncover unforeseen performance attributes;

5. Examination of acquired results encompassing the profiling of computational stages, exploration of microarchitectural impacts, and the development of technology framework suggestions for effective software development processes.

Presentation of main material. For modern biometric systems, rapid large data processing is crucial, requiring improved matching algorithms. One approach is vectorization using SIMD instructions (SSE, AVX, AVX-512). However, applying this technique in complex biometric algorithms presents difficulties related to processor architecture, requiring theoretical analysis investigation.

The theoretical foundation begins with the Roofline model, which defines general computational system performance limitations. According to this model, overall system performance can be described as follows:

$$\pi = \min(\pi_{\max}, M_{\max} \cdot I), \quad (1)$$

where π is overall system performance;

$\pi_{\max}$ is peak processor performance;

$M_{\max}$ – is peak memory bandwidth.

I is operational intensity, representing the ratio of executed operations to transferred data volume.

According to this model, computational systems can be divided into memory-bound and compute-bound according to the following metric:

$$\Delta I = \frac{\pi_{\max}}{I}, \quad (2)$$

where ΔI is the intensity limit point.

According to formula (2), computational models satisfying the inequality $I < \Delta I$ can be classified as memory-bound, while all others are compute-bound. Biometric systems characteristically show uneven distribution where some identification stages belong to the first category while others belong to the second. Determining overall intensity (and consequently, system performance) under these conditions is generally a complex task.

For evaluating parallel and vectorized system performance, Amdahl's law and its improvements, such as the Universal Scalability Law (USL), can be used. The Universal Scalability Law describes relative throughput as follows:

$$C(N) = \frac{N}{1 + \alpha \cdot (N-1) + \beta \cdot N \cdot (N-1)}, \quad (3)$$

where $C(N)$ is relative throughput;

N is the number of processors engaged;

α is the contention coefficient reflecting shared resource waiting ($0 \le \alpha \le 1$);

β is the coherency coefficient reflecting synchronization overhead ($\beta \ge 0$).

For vectorized systems, Amdahl's law can be applied in the following form:

$$\sigma_v = \frac{1}{1 - \rho + \frac{\rho}{\omega}}, \quad (4)$$

where σ_v is vectorization acceleration;

ρ is the vectorizable algorithm portion ($0 \le \rho \le 1$);

ω is SIMD register width (depends on vectorization type).

Besides vectorization acceleration, efficiency must be evaluated as shown below:

$$\epsilon_v = \frac{c_v}{\omega}, \quad (5)$$

where ϵ_v is vectorization efficiency;

c_v is the number of activated vectorization channels.

Overall system performance can then be estimated as follows:

$$\pi_{\text{result}} = \min(C(N), \sigma_v, \pi), \quad (6)$$

where π_{result} is the final system productivity assessment.

Depending on parameters, the following taxonomy is proposed for evaluating vectorization applicability:

1. High feasibility: vectorization application will likely positively affect system productivity. Evaluation criterion: $\epsilon_v > 0.9$;

2. Medium feasibility: vectorization application will likely neither benefit nor harm system performance. Evaluation criterion: $0.5 < \epsilon_v < 0.9$;

3. Low feasibility: vectorization application will likely adversely affect system performance. Evaluation criterion: $\epsilon_v < 0.5$.

The suggested technology for software development that allows for selective SIMD vectorization has a systematic workflow that helps developers choose the best optimization strategy. This technology framework has the following steps:

1. The first step is to break down the algorithm and sort the stages. Developers figure out the different computational stages in the biometric algorithm and look at their traits, such as how the data depends on each other, how hard the operations are, and how memory is accessed. Formulas (4) and (5) set the vectorization efficiency standards that each stage is compared to in order to see if it would work well with SIMD optimization;

2. Stage two uses theoretical models to guess how well things will work. For every identified stage, operational intensity is computed using the Roofline model to distinguish between memory-bound and compute-bound classifications. To guess how well an algorithm will scale in parallel, we look at its characteristics and use them to guess the Universal Scalability Law parameters. To find out if a stage is highly, moderately, or not feasible, the efficiency of vectorization is calculated;

3. Stage three is about making specific choices about instruments. Developers make smart decisions about how to optimize each algorithmic stage based on what they think will happen. If a stage can be vectorized easily, it gets SIMD implementation with the right instruction sets, like SSE or AVX. Stages with low feasibility keep the usual thread-based parallelization without vectorization. To clear up any doubts, stages with medium feasibility need prototyping and benchmarking;

4. Stage four includes putting something into action and checking to see if it works. The chosen optimization strategies are put into action with an eye toward microarchitectural factors like memory alignment, cache use, and data structure design. Comprehensive profiling across varying thread counts and dataset sizes validates theoretical predictions and identifies unexpected interactions between different optimization approaches.

The proposed technology works with a fingerprint comparison algorithm based on minutiae features. The algorithm is four-stage and includes the following steps:

1. Feature point extraction within centroid squares;

2. Calculation of local Euclidean structure metrics within extracted squares;

3. Local structure matching: two squares are compared from the reference point perspective and all others. Distances and angles in corresponding squares are analyzed. The main goal is to identify similar Euclidean structures as the foundation for the final stage;

4. Global matching conduct. Based on found similar structures, relative displacement calculation of the first image relative to the second is performed. After displacement execution, the entire point set is matched against each other. Based on match quantity, a similarity metric between images is determined.

Using the suggested technology framework on this algorithm gives clear suggestions for how to improve it. Stages one and two show that the data in them is not very dependent on each other because operations are done on separate minutiae points within centroid squares. This means that vectorization is likely to work well according to the established efficiency criteria. The calculations for the local Euclidean structure involve doing the same math over and over on coordinate data with memory access patterns that are easy to predict. This makes them great candidates for SIMD optimization. On the other hand, stages three and four show a high level of algorithmic complexity of order n squared m squared, with a lot of conditional branching based on similarity thresholds and complex data dependencies through nested iterations. The matching operations need to be synchronized often and access dictionary structures in an irregular way, which shows that vectorization is not very likely to work. This analysis results in the technology-driven decision to implement selective vectorization solely for stages one and two, while preserving traditional parallelization for stages three and four.

Three computational implementations were used to validate the technology: a pure parallelization model, a full SIMD vectorization model, and a selective hybrid model. The code example below shows how to use the parallel model, which is the starting point for the comparison. This implementation shows the four-step algorithmic structure that includes extracting minutiae, calculating metrics, comparing them locally, and comparing them globally. The vectorized model has the same algorithmic structure as the original, but in stages one and two, it uses SIMD primitives instead of scalar operations. The hybrid model uses parallelization for stages three and four and selective vectorization for stages one and two. This is based on theoretical predictions about how feasible vectorization is. Complete implementations of all three computational models are available in the referenced repository. The parallel version presented here demonstrates the intricate calculations and data dependencies that favor selective vectorization over full vectorization strategies. The minutiae comparison process exhibits extremely strong data interdependencies, reflected primarily in the $O(n^2m^2)$ algorithmic complexity, where n and m denote the number of minutiae points on the two fingerprint images being compared. These dependencies create computational bottlenecks that benefit from selective rather than uniform vectorization. Full implementation details and source code are provided in [18].

Experimental methodology. Experiments were conducted on a system with an Intel Core i9 processor supporting SSE4.2, AVX, and AVX2 instructions. Operating system: Windows, runtime environment: NET 8.0. The FVC 2000 (DB1_B) database contains 80 fingerprint images with 4,421 minutiae. Each measurement was repeated at least 5 times to ensure statistical reliability. Pre-measurement warm-up was performed to stabilize JIT compiler and thread pool states. Planned experiments included: execution time measurement relative to thread count used, execution time measurement relative to input dataset size, and algorithm execution stage profiling.

Experimental results. All experimental results were recorded using the auxiliary Stopwatch class. Table 1 reflects execution time measurement results relative to thread count used (on a dataset subset containing 32 images out of 80, containing 1,396 minutiae).

Table 1

Biometric comparator execution time depending on thread count

Thread count	Parallel version (ms)	Vectorized version (ms)	Hybrid version (ms)	Vectorized speedup	Hybrid speedup
1	32354.81	48202.66	30811.38	0.67	1.05
2	10005.52	11558.76	7463.21	0.87	1.34
3	7442.14	8640.63	4896.15	0.86	1.52
4	6703.53	7861.66	3906.52	0.85	1.72
5	5954.90	7479.67	3150.62	0.80	1.89
6	6207.16	8377.23	3378.49	0.74	1.84
7	6204.50	8404.29	3249.11	0.74	1.91
8	6365.02	8749.76	3130.85	0.73	2.03
10	6168.42	8714.61	2873.36	0.71	2.15
12	6364.33	9222.22	2890.54	0.69	2.20
16	6719.81	10342.11	2908.35	0.65	2.31
20	6818.92	10429.01	2807.38	0.65	2.43
22	6978.51	10717.22	2792.20	0.65	2.50
24	7410.61	11385.68	2911.62	0.65	2.55
32	7601.49	11860.72	2856.30	0.64	2.66

During the experiment, the following general data were obtained: optimal thread count parameters (5 for vectorized and parallel models, 22 for hybrid), and scalability coefficients (relative to single thread): 5.43 for parallel version, 6.44 for SIMD version, and 11.03 for hybrid version.

Experiment 1 results are illustrated on figures 1 and 2.

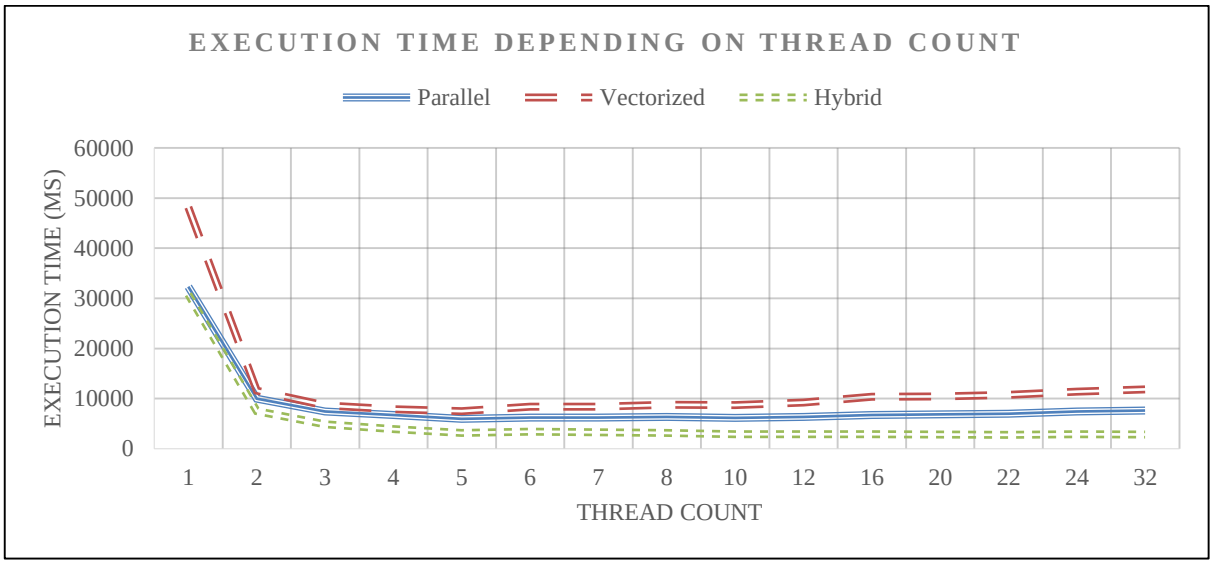

Fig. 1. Graph of execution time depending on thread count

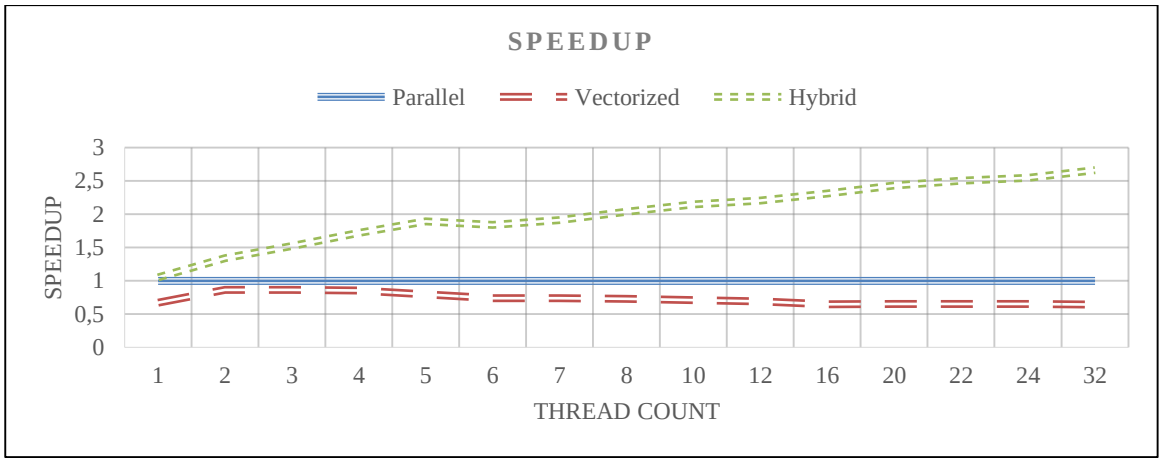

Fig. 2. Speedup graph

Tables 2 and 3 show the results of measuring the execution time relative to the size of the input dataset (fixed number of threads – maximum (32 units)).

Table 2

Biometric comparator execution time depending on dataset size

Dataset size	Parallel version (ms)	Vectorized version (ms)	Hybrid version (ms)	Vectorized speedup	Hybrid speedup
8	41.55	44.97	26.16	0.92	1.59
16	181.70	207.93	81.13	0.87	2.24
24	2932.56	3859.72	1009.70	0.76	2.90
32	7735.33	12140.93	2868.27	0.64	2.70
40	20388.46	39034.85	8537.78	0.52	2.39
48	26316.74	47202.80	10429.11	0.56	2.52
56	58307.50	95497.23	21141.34	0.61	2.76
64	85297.40	139615.75	29972.23	0.61	2.85
72	98843.78	146797.38	51935.87	0.67	1.90
80	205992.38	205114.97	60634.67	1.00	3.40

Table 3

Relative marginal increase in execution time and data volume depending on the dataset size

Dataset size	Parallel increase	Vectorized increase	Hybrid increase	Data increase
8	1	1	1	1
16	4.37	4.62	3.10	2.17
24	16.14	18.56	12.45	1.73
32	2.64	3.15	2.84	1.55
40	2.64	3.22	2.98	1.40
48	1.29	1.21	1.22	1.22
56	2.22	2.03	2.03	1.25
64	1.46	1.46	1.42	1.19
72	1.16	1.05	1.73	1.08
80	2.08	1.40	1.17	1.15

The results of Experiment 2 are illustrated in figures 3–5 for clarity.

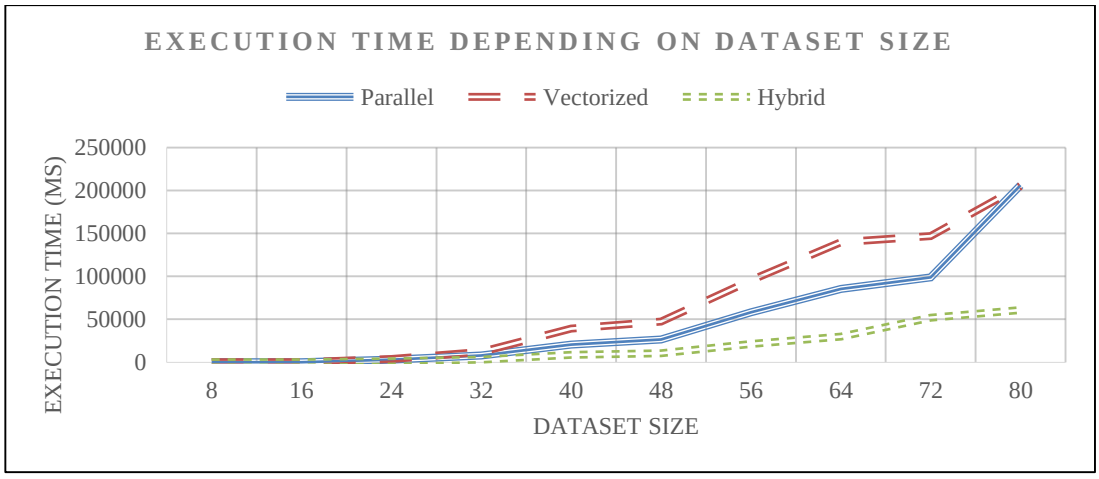

Fig. 3. Graph of execution time dependence on dataset size

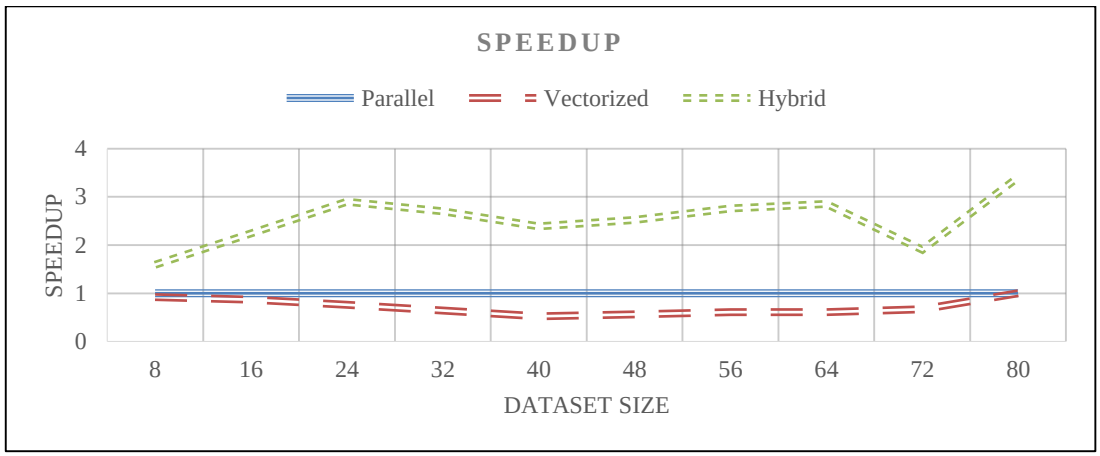

Fig. 4. Graph of acceleration of models relative to the parallel model of computing organization

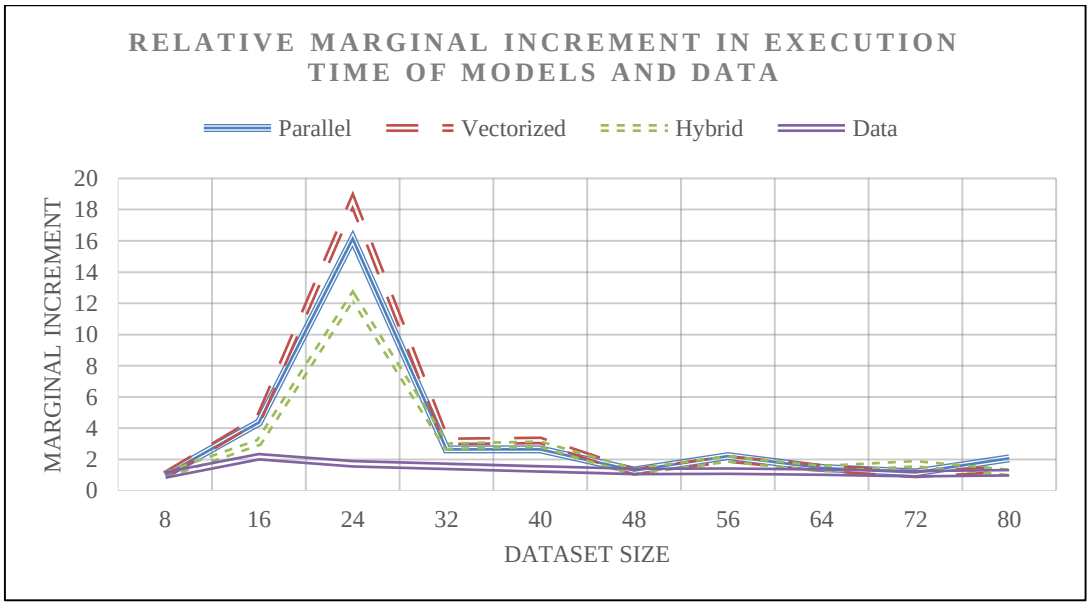

Fig. 5. Graph of the relative marginal increment in execution time of models and data

Tables 4–5 reflect algorithm execution stage profiling results (fixed thread count at maximum 32 units, all dataset data participate in identification).

Table 4

Biometric comparator stage execution times

Stage	Parallel version (ms)	Vectorized version (ms)	Hybrid version (ms)	Vectorized speedup	Hybrid speedup
Central minutiae selection	6.25 ± 7.54	0.41 ± 0.07	0.47 ± 0.04	15.38	13.33
Metric calculation	13.66 ± 14.69	6.12 ± 1.53	5.63 ± 0.37	2.23	2.43
Local comparison	21272.66 ± 352.41	23595.14 ± 718.51	16162 ± 54.69	0.90	1.32
Global comparison	113509.01 ± 4725.28	160956.03 ± 9556.09	45583 ± 824.14	0.71	2.49

Table 5

The share of time spent computing the algorithm stages from the total execution time of the model

Stage	Parallel (%)	Vectorized (%)	Hybrid (%)
Central minutiae selection	4.64×10^{-5}	2.22×10^{-6}	7.61×10^{-6}
Metric calculation	1.01×10^{-4}	3.31×10^{-5}	9.11×10^{-5}
Local comparison	15.8	12.8	26.2
Global comparison	84.2	87.2	73.8

The results of Experiment 3 are illustrated in figure 6 for clarity.

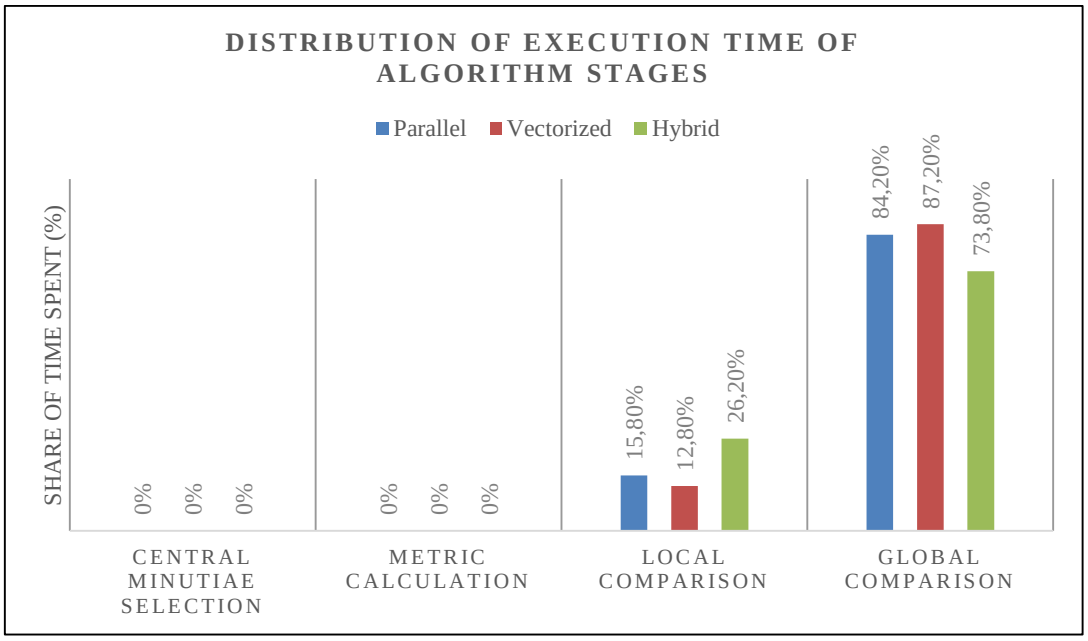

Fig. 6. Diagram of the distribution of execution time of the algorithm execution stages

Discussion and analysis. Conducted experiments illustrated ambiguous results. All experiments confirmed the working hypothesis regarding the inappropriateness of applying SIMD vectorization for local and global matching stages. However, despite this, at full load (80 images) algorithm results became equal. This might be related to excessive thread count used in calculations, requiring additional experiments. The consistently good hybrid version results compared to parallel version were unexpected. The hybrid version demonstrates stable acceleration regardless of thread count, input data volume, and other factors. Most interesting is that the most computationally expensive stages (according to profiling) are identical in parallel and hybrid versions. Additional verification confirmed no computational discrepancies, excluding design errors as possible causes. This phenomenon requires further investigation.

Various possible causes for this system behavior were examined, including memory structure alignment, JIT compilation results, thread pool special states, AVX instruction effects on processor physical parameters, cache memory states, and others. None of these hypotheses found confirmation. Profiling of parallel and hybrid systems was conducted, and internal states of data storage objects (particularly dictionaries and minutiae model classes) were studied in detail. The following hypotheses can be formulated:

1. Local matching acceleration (1.32 times according to table 3 results) can likely be explained by vectorization effects during cache dictionary data writing. This is confirmed by additional metrics such as chi-square coefficient ratios (4.48), variance index ratios (2.16), and basic statistical metrics of hash chain lengths. These results emphasize that vectorization writes data more uniformly to dictionaries, reducing overhead with each dictionary access, which would be barely noticeable with uniform or single access but becomes extremely noticeable with uneven access patterns like stage 3;

2. Global matching acceleration (2.49 times) can be explained by internal compiler operation. When working with SIMD vectorization, we applied object field access (coordinates). This likely led to their memory alignment, ensuring better access. Additionally, there may be affects similar to point 1 when accessing data from local matching result lists, though this effect appears weakly influential.

Research results are not final and require further experiments to confirm or refute formed hypotheses. Investigation of computational device parameter effects, compiler versions, and other factors is necessary, along with more thorough vectorization tool analysis to form stable conclusions regarding vectorization application benefits as demonstrated in this study. Moreover, additional research is needed regarding this effect's applicability in other biometric systems or dactyloscopic identifiers not working with fingerprint minutiae.

Conclusions and future research perspectives. This research described software development technology for high-performance biometric comparison systems. Thorough analysis of theoretical research was conducted, and parallel, vectorized, and hybrid systems were proposed for solving this task. Experiments revealed that, as theoretical models predicted, full vectorization does not provide significant performance gains. However, the hybrid model unexpectedly demonstrated significantly better results. Additional measurements allowed formation of working hypotheses explaining the discovered effect. Possible problems were identified and the necessity for further research was formulated to expand the developed software development technology's applicability to other biometric systems. The findings of the research are not conclusive and necessitate additional inquiry to corroborate and enhance the proposed technological framework. Based on the current findings and their limitations, several important directions for future research have been identified.

The initial research direction entails a systematic examination of microarchitectural factors affecting selective vectorization performance. The present study discovered unanticipated performance enhancements in the hybrid model, seemingly resulting from secondary optimization effects such as memory alignment and cache utilization improvements. Nonetheless, these hypotheses necessitate stringent experimental validation via controlled studies that isolate specific microarchitectural variables. Future studies ought to investigate the responses of various processor architectures, such as AMD Ryzen, ARM-based systems, and the latest Intel generations, to selective vectorization strategies. Examining the effects of cache hierarchy by systematically varying dataset sizes in relation to L1, L2, and L3 cache capacities would elucidate the impact of cache utilization on hybrid model performance. An examination of memory alignment effects via intentionally misaligned data structures would either confirm or disprove the alignment hypothesis. Analyzing compiler optimization by comparing different compiler versions and optimization flags would show how much performance improvements depend on certain compilation methods.

The second research direction concentrates on broadening the technological framework to encompass various biometric modalities beyond minutiae-based fingerprint recognition. The current validation utilized a specific algorithmic methodology characterized by significant data interdependency during matching phases and minimal interdependency during feature extraction phases. Subsequent research ought to examine the applicability of selective vectorization principles to various biometric identification techniques, such as iris recognition systems, facial recognition algorithms, voice authentication processing, and palm print analysis. Each modality has its own set of computational traits, such as different levels of data parallelism, memory access patterns, and algorithmic complexity. Using the proposed technology framework in a systematic way across many biometric modalities would show where it can be used and what changes need to be made for different types of computations.

The third research direction focuses on scalability in extreme computational environments and distributed architectures. The current experiments employed datasets comprising eighty fingerprint images with several thousand minutiae points. In large-scale deployment scenarios, databases with millions of biometric templates need distributed computing architectures. Future research should examine the scalability of selective vectorization technology in massively parallel systems, encompassing GPU acceleration, multi-node cluster computing, and heterogeneous computing environments that integrate CPUs with specialized accelerators. Research into load balancing techniques for hybrid vectorized and non-vectorized stages in distributed systems would tackle real-world deployment issues. An examination of communication overhead in distributed selective vectorization implementations would ascertain the ideal granularity for workload distribution.

The fourth area of research looks at how to combine selective vectorization with other optimization methods in biometric systems. The present study concentrated on the interplay between parallelization and SIMD vectorization; however, contemporary high-performance systems utilize multiple optimization layers concurrently. Future research ought to examine the interplay between selective vectorization and algorithmic optimizations, including spatial indexing structures for search space reduction, multi-resolution processing strategies for the prompt dismissal of non-matching candidates, and adaptive precision techniques that modify computational accuracy in accordance with matching confidence levels. An examination of integrated optimization strategies would ascertain whether selective vectorization advantages synergize with alternative performance enhancement methodologies or if optimization interactions yield diminishing returns. Research ought to investigate the correlation between selective vectorization and memory hierarchy optimization, encompassing prefetching strategies, data layout transformations, and blocking techniques to enhance temporal locality.

The fifth research direction entails the formalization and quantification of the technology decision framework. The present study suggests efficiency thresholds and qualitative criteria for the applicability of vectorization, grounded in operational intensity and data dependency attributes. Subsequent research ought to formulate more robust quantitative models that accurately forecast vectorization performance enhancements based on algorithmic and data attributes. Research must ascertain empirical correlations among algorithmic attributes, including loop structure complexity, memory access stride patterns, branch prediction efficacy, and the practical vectorization speedup attained. Creating validated performance models would help software developers choose the best optimization strategy with more confidence during the planning stages. Research should also look into how well selective vectorization strategies work with different types of input, such as changes in the quality of biometric samples, the size of the database, and the patterns of query load. This will make sure that performance stays the same in production deployments. These research directions collectively address the limitations of the current study while enhancing the applicability and rigor of the proposed software development technology for high-performance biometric systems. A systematic examination of these avenues would elevate selective vectorization from a recognized technique for particular situations to a holistic methodology suitable for various computational environments in biometric identification and possibly other performance-sensitive fields necessitating analogous optimization strategy determinations.

References:

1. Deshpande, U.U., Malemath, V.S., Patil, S.M. and Chaugule, S.V. (2020), «CNNAI: A convolution neural network-based latent fingerprint matching using the combination of nearest neighbor arrangement indexing», *Frontiers in Robotics and AI*, Vol. 7, doi: 10.3389/frobt.2020.00113.
2. Grosz, S.A. and Jain, A.K. (2024), «AFR-Net: Attention-driven fingerprint recognition network», *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 6, No. 1, pp. 30–42, [Online], available at: arXiv: 2211.13897
3. Minaee, S., Azimi, E. and Abdolrashidi, A. (2019), «FingerNet: Pushing the limits of fingerprint recognition using convolutional neural network», *arXiv preprint*, [Online], available at: arXiv:1907.12956
4. Nguyen, D.-L., Cao, K. and Jain, A.K. (2018), «Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge», *Proceedings of the International Conference on Biometrics (ICB)*, Gold Coast, Australia, [Online], available at: https://github.com/luannd/MinutiaeNet
5. Tandon, S. and Namboodiri, A. (2022), «Transformer based fingerprint feature extraction», *Proceedings of the 26th International Conference on Pattern Recognition (ICPR)*, Montreal, Canada, pp. 870–876, [Online], available at: arXiv: 2209.03846
6. Tang, Y., Gao, F., Feng, J. and Liu, Y. (2017), «FingerNet: An unified deep network for fingerprint minutiae extraction», *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*, Denver, USA, doi: 10.1109/BTAS.2017.8272688.
7. Fillinger, A., Diduch, L., Hamchi, I. and Stanford, V.M. (2011), «Experiments in parallel fingerprint matching: Architectural implications for large scale fingerprint matching evaluation systems», *NIST Interagency/Internal Report (NISTIR)*, doi: 10.6028/NIST.IR.7754.
8. Gutierrez, P.D., Lastra, M., Herrera, F. and Benitez, J.M. (2014), «A high performance fingerprint matching system for large databases based on GPU», *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 1, pp. 62–71, doi: 10.1109/TIFS.2013.2291969.
9. Guermouche, A. and Orgerie, A.-C. (2022), «Thermal design power and vectorized instructions behavior», *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 2, doi: 10.1002/cpe.6261.
10. Jakobs, T., Naumann, B. and Rünger, G. (2020), «Performance and energy consumption of the SIMD Gram–Schmidt process for vector orthogonalization», *The Journal of Supercomputing*, Vol. 76, pp. 1999–2021, doi: 10.1007/s11227-019-02839-0.
11. Schöne, R., Ilsche, T., Bielert, M. et al. (2019), «Energy efficiency features of the Intel Skylake-SP processor and their impact on performance», *Proceedings of the IEEE High Performance Computing & Simulation (HPCS)*, Dublin, Ireland, pp. 399–406, doi: 10.1109/HPCS48598.2019.9188239.
12. Zhang, N., Fu, S. and Franchetti, F. (2024), «Towards closing the performance gap for cryptographic kernels between CPUs and specialized hardware», *Proceedings of the 58th IEEE/ACM International Symposium on Microarchitecture (MICRO '25)*, doi: 10.1145/3725843.3756120.

13. Koval, L.H., Zlepko, S.M., Novitskyi, H.M. and Krekoten, Ye.H. (2019), «Methods and technologies of biometric identification by results of literary sources», *Scientific Notes of Taurida National V.I. Vernadsky University*. Series. *Technical Sciences*, Vol. 30 (69), Part 1, No. 2, pp. 104–112.
14. Yanko, A., Martynenko, A. and But, O. (2021), «Methods of using SIMD instructions on x86-compatible processors of older generation», *Control, Navigation and Communication Systems. Collection of Scientific Papers*, No. 4, pp. 44–51, doi: 10.26906/SUNZ.2021.4.044.
15. Popov, O.V., Rudych, O.V. and Chistyakov, O.V. (2018), «Multilevel model of parallel computing for linear algebra problems», *CEUR Workshop Proceedings*, Vol. 2139, pp. 83–92, [Online], available at: https://ceur-ws.org/Vol-2139/83-92.pdf
16. Doroshenko, A.Yu. and Beketov, O.G. (2017), «Method of parallelization of loops for grid calculation problems on GPU accelerators», *Problems in Programming*, No. 1, pp. 59–66, doi: 10.15407/pp2017.01.059.
17. Shkil, O., Filippenko, O., Rakhlis, D. et al. (2024), «Analysis of the implementation efficiency of digital signal processing systems on the technological platform SoC ZYNQ 7000», *Radioelectronic and Computer Systems*, No. 4, pp. 91–98, doi: 10.32620/reks.2024.4.14.
18. Pohuliaiev, Y. (2025), «Software implementation of biometry comparison with selective SIMD usage (Version 1.0.0)», *Kharkiv National University of Radio Electronics*, doi: 10.5281/zenodo.17571115.

Погуляєв Юрій Сергійович – аспірант Харківського національного університету радіоелектроніки.
https://orcid.org/0009-0005-5883-1661.
Наукові інтереси:
– комп'ютерний зір;
– математичне моделювання;
– біометричні системи.
E-mail: yurii.pohuliaiev@nure.ua.

Смеляков Кирило Сергійович – доктор технічних наук, професор, завідувач кафедри ПІ Харківського національного університету радіоелектроніки.
https://orcid.org/0000-0001-9938-5489
Наукові інтереси:
– штучний інтелект;
– машинне навчання;
– комп'ютерний зір;
– математичне моделювання.
E-mail: kyrylo.smelyakov@nure.ua.

Погуляєв Ю.С., Смеляков К.С.

Технологія розробки програмного забезпечення для високопродуктивних компараторів біометричних даних з використанням вибіркової векторизації SIMD

Це дослідження представляє систематичну методологію розробки програмного забезпечення для високопродуктивних компараторів біометричних даних, засновану на принципах вибіркової векторизації SIMD. Технологія вирішує важливу проблему вибору найкращої стратегії оптимізації, створюючи теоретичну основу, яка вказує, коли й де використовувати векторизацію замість звичайної паралелізації на різних етапах алгоритму. Технологія поєднує модель Roofline для обчислювальних обмежень та Універсальний закон масштабованості для тестування паралельних систем, щоб надати розробникам таксономії ефективності та критерії прийняття рішення щодо використання векторизації в біометричних системах, які обробляють мільйони зразків. Для валідації технології використовувався чотириетапний алгоритм порівняння відбитків пальців на основі мінуцій, що виконувався за допомогою різних обчислювальних методів: традиційне паралелізування на основі потоків, векторизація SIMD з використанням інструкцій SSE та AVX, а також гібридні методології, що інтегрують обидва методи. В експериментальній валідації використовувалася архітектура Intel Core i9 разом з базою даних FVC 2000, що містить 80 зображень відбитків пальців та 4421 точку мінуцій, з використанням суворих статистичних методологій протягом кількох ітерацій. Алгоритм містить вилучення дрібних частинок з квадратів центроїдів, обчислення локальних евклідових структур, аналіз геометричних відстаней і кутів, а також глобальне зіставлення з розрахунками зміщення. Результати показали неочікувані результати, які суперечать поширеним уявленням про оптимізацію. Гібридна реалізація продемонструвала стабільне покращення продуктивності у всіх випадках, зі збільшенням швидкості у 2,66 раза з 32 потоками та покращенням у 3,40 раза з повними наборами даних. Це було очікувано, оскільки повна векторизація SIMD працювала не так добре, як традиційна паралелізація, через високу алгоритмічну взаємозалежність. Аналіз профілювання показав, що паралельна та гібридна версії мали однакові обчислювальні етапи. Це означає, що підвищення продуктивності відбулося завдяки вторинним ефектам оптимізації, таким як краще вирівнювання пам'яті, краще використання кешу завдяки рівномірному розподілу словника та оптимізація, індукована компілятором, під час доступу до поля об'єкта. Дослідження створює технологічну основу для вибіркової векторизації в біометричних додатках. Це надає розробникам програмного забезпечення систематичний спосіб створення високопродуктивних систем ідентифікації, які можна використовувати в реальних ситуаціях, де їх потрібно використовувати у великих масштабах.

Ключові слова: біометрична ідентифікація; векторизація; SIMD; паралелізація; продуктивність; AVX; SSE; технологія розробки ПЗ.

The article was sent to the editorial board on 25.09.2025.