doi: 10.17586/2226-1494-2023-23-5-955-966

# A new efficient adaptive rood pattern search motion estimation algorithm

**Sherin A. Shaker[1]✉, Ahmad Suki C. M. Arif[2], Yousef Fazea[3]**

[1,2] University Utara Malaysia, Sintok, 06000, Malaysia
[3] Marshall University, Huntington, 25755, United States of America

[1] Shereenshaker2004@yahoo.com✉, https://orcid.org/0000-0002-2501-7584
[2] suki1207@uum.edu.my, https://orcid.org/0009-0008-2724-1281
[3] fazeaalnades@marshall.edu, https://orcid.org/0000-0003-3544-2434

**Abstract**
Motion estimation plays a crucial role in video coding; the Adaptive Rood Pattern Search (ARPS) algorithm is a well known fast motion estimation algorithm. However, ARPS has certain limitations, such as the lack of an accurate starting motion vector, a fixed Zero Motion Prejudgment (ZMP) threshold unsuitable for fast motion video sequences, and the repetitive use of a Unit Rood Pattern (URP) resulting in increased computational complexity. To address these issues, this paper proposes a novel algorithm called Efficient Adaptive Rood Pattern Search (EARPS). EARPS overcomes these limitations by employing the Full Search algorithm to obtain optimal motion vectors for the first column in each frame, adopting a dynamic ZMP threshold that adapts to varying motion speeds in video sequences and utilizing URP only once to reduce computational overhead. The performance of the new proposed EARPS algorithm is evaluated and compared with that of ARPS algorithm using various video sequences with different motion speeds. The number of searching points and Peak Signal-to-Noise Ratio (PSNR) are used to quantify computing complexity and accuracy. The experimental findings show that EARPS surpasses ARPS in terms of computing complexity while retaining a decent degree of PSNR accuracy. The proposed EARPS motion estimation algorithm main contribution is to achieve high speed with reasonable accuracy, regardless of the type of motion speed in the video frames. The EARPS algorithm offers a substantial advancement over ARPS, delivering a more efficient motion estimation method with broader applicability in video processing. It represents a significant contribution to the development of effective motion estimation algorithms.

**Keywords**
motion estimation, computational complexity, ARPS, ZMP, PSNR

**For citation:** Shaker S.A., Arif A.S., Fazea Y. A new efficient adaptive rood pattern search motion estimation algorithm. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 5, pp. 955–966. doi: 10.17586/2226-1494-2023-23-5-955-966

УДК 004.627

# Новый эффективный адаптивный алгоритм шаблонного поиска для оценки движения

**Абделаал Шерин Шейкер[1]✉, Ахмад Суки Ч.М. Ариф[2], Юсеф Фазеа[3]**

[1,2] Университет Утра Малайзии, Синток, 06000, Малайзия
[3] Университет Маршалл, Хантингтон, 25755, Соединенные Штаты Америки

[1] Shereenshaker2004@yahoo.com✉, https://orcid.org/0000-0002-2501-7584
[2] suki1207@uum.edu.my, https://orcid.org/0009-0008-2724-1281
[3] fazeaalnades@marshall.edu, https://orcid.org/0000-0003-3544-2434

**Аннотация**
Оценка движения играет решающую роль при кодировании видео. Адаптивный алгоритм шаблонного поиска (Adaptive Rood Pattern Search, ARPS) является известным алгоритмом быстрой оценки движения. При этом ARPS имеет следующие ограничения: отсутствие точного начального вектора движения; фиксированный порог предварительного суждения о нулевом движении (Zero Motion Prejudgment, ZMP), неподходящий для

видеопоследовательностей с быстрым движением; повторяющееся использование шаблона единичного движения (Unit Rood Pattern, URP), что приводит к увеличению вычислительной сложности. Для решения данных ограничений предложен новый алгоритм под названием «Эффективный адаптивный алгоритм шаблонного поиска» (Efficient Adaptive Rood Pattern Search, EARPS). В основе EARPS лежит алгоритм полного поиска, который получает оптимальные векторы движения для первого столбца в каждом кадре, принимает динамический порог ZMP, который адаптируется к различным скоростям движения в видеопоследовательностях, и использует URP один раз для уменьшения вычислительных затрат. Выполнена оценка и сравнение производительности нового алгоритма EARPS и алгоритма ARPS с использованием различных видеопоследовательностей для различных скоростей движения. Количество точек поиска и пиковое отношение сигнал-шум (Peak Signal-to-Noise Ratio, PSNR) использованы для количественной оценки сложности и точности вычислений. Экспериментальные результаты показали, что EARPS превосходит ARPS с точки зрения вычислительной сложности, сохраняя при этом высокую степень точности PSNR. Основной вклад предложенного алгоритма оценки движения EARPS заключается в достижении высокой скорости с приемлемой точностью, независимо от скорости движения в видеокадрах. Алгоритм EARPS по сравнению с ARPS, обеспечил более эффективный метод оценки движения с более широкой применимостью в обработке видео. Полученный результат является значительным вкладом в разработку эффективных алгоритмов оценки движения.

**Ключевые слова**
оценка движения, вычислительная сложность, ARPS, ZMP и PSNR

**Ссылка для цитирования:** Шейкер А.Ш., Ариф Ахмад Суки Ч.М., Фазеа Ю. Новый эффективный адаптивный алгоритм шаблонного поиска для оценки движения // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 5. С. 955–966 (на англ. яз.). doi: 10.17586/2226-1494-2023-23-5-955-966

## Introduction

Motion Estimation (ME) is the most effective component of the video coding process because it enables efficient video data compression by anticipating the motion of objects within frames [1, 2]. Block Matching Motion Estimation is a popular method due to its simplicity and effectiveness; however, selecting an effective Block Matching Algorithm (BMA) necessitates a trade-off between computing complexity and precision [3, 4]. When selecting the optimal BMA, the trade-off between computational complexity and precision is crucial. While the Exhaustive Block Matching algorithm is the most accurate and straightforward BMA, it is also the most computationally demanding. Because of their predetermined search patterns, rapid algorithms, such as Three Step Search, New Three Step Search, Four Step Search, and Diamond Search are quicker but less accurate [5–8]. Researchers have developed several adaptive algorithms to solve the limitations of existing algorithms and produce more efficient and accurate techniques [9–11]. Among them, the well-known quick adaptive method is the Adaptive Rood Pattern Search (ARPS) [12–14]. However, ARPS has three critical weaknesses. Firstly, it lacks an accurate starting motion vector. Secondly, it uses a fixed Zero Motion Prejudgment (ZMP) [15–17] threshold value which is unsuitable for fast-motion video sequences. Lastly, it repetitively uses a (URP), leading to increased computational complexity and potentially inaccurate results in terms of coding speed or quality. This work introduces a revolutionary rapid Efficient Adaptive Rood Pattern Search (EARPS) method that delivers great speed and accuracy independent of motion speed in video frames. The results of the experiments reveal that the new EARPS strategy outperforms the ARPS algorithm in terms of processing cost while preserving a Peak Signal-to-Noise Ratio (PSNR) comparable to the Full Search (FS) algorithm. To begin with, the suggested EARPS method, it makes use of a Mean Absolute Frame Difference (MAFD) detector [18] to determine the sort of motion speed in each frame of the video sequence, whether slow or fast. When the kind of motion speed in the video frames is identified, the dynamic ZMP is enabled [19, 20]. The proposed algorithm next step should employ the appropriate threshold, whether slow or fast, to speed up calculations. The new proposed algorithm makes achievement with that point, which did not exist in the ARPS algorithm. Due to the lack of speed detectors, ARPS applied a constant ZMP threshold to all video sequences, regardless of the types of motion speeds present. In contrast to ARPS, which employs a predefined ZMP threshold value of 512 that is only appropriate for slow motion and does not take into account the kind of motion speed in the video sequence, the proposed EARPS algorithm utilizes a dynamic threshold value that is adjusted for both slow and rapid motion. The dynamic ZMP threshold in the proposed EARPS method accelerates calculations, and modifying its values greatly contributes to speeding up computations while maintaining acceptable accuracy PSNR [21], as it stops the searching process and declares the block to be a static block in an early stage in the algorithm saving many useless computations. Also, the matching criteria used is different; it is the Sum of Absolute Difference (SAD) in the ARPS algorithm, while in the EARPS algorithm, we use Mean Absolute Difference (MAD) which is more complex in computations but gives more precise and accurate results as matching criteria for blocks. Secondly, the proposed EARPS algorithm solves the problem for the accurate starting point that the ARPS algorithm was facing, the blocks in the leftmost column of each frame in ARPS are assigned a constant value of 2 pixels as a motion vector, this assumption harms the accuracy as it does not depend on actual calculations, it is only an assumption. Besides, it may be incorrect for most cases as it has no scientific proof, for example, why the motion vector is assumed to be two pixels and not any other value, EARPS algorithm uses the FS algorithm to calculate the motion vector for the leftmost column blocks. Due to its intensive search procedure, the FS algorithm gives the optimal motion vector to produce highly accurate results. So, the proposed EARPS algorithm benefits from an accurate starting point that is close to the global minimum. This precise starting point significantly enhances the accuracy and computational efficiency of the EARPS algorithm. Thirdly, while ARPS algorithm was using a URP in its refined local search stage many times unrestrictedly and without a limit to get the final motion vector, which led to useless computations, since the search area may be not within the global minimum. However, the proposed EARPS algorithm, due to the usage of FS algorithm (as already mentioned), gives an accurate starting point for the search that accordingly allows the URP to be used in the refined search stage only once, as the search process has been started in the correct location. That way the new proposed EARPS algorithm can speed its computations and get the accurate motion vector faster.

## Overview of the ARPS Algorithm

The most renowned rapid adaptive block matching technique is ARPS [22]. ARPS uses frame-block-coherent motion. Thus, if the macroblocks around it change, then the candidate block will also change. ARPS predicts the motion vector for the current macroblock using the macroblock to its left. The ARPS algorithm consists of initial and refined local search stages as shown in Fig. 1. ARPS uses the ZMP technique which identifies the immobile blocks and accordingly terminates the searching process early. The technique reduces computational complexity by identifying motionless blocks. ARPS algorithm employs ZMP to enhance search efficiency [23, 24]. ZMP speeds up computations, especially for slow-motion video sequences. The method reduces computing costs by omitting the exhaustive search for immobile blocks.

**Steps of ARPS Algorithm**

**Step 1:** Determine the matching error SAD between the current block and the corresponding block in the reference frame. When the matched error SAD value is compared to the ZMP threshold value $T = 512$, the outcome is: If the matching error is less than T value, the present block is deemed static, and the following search is skipped. If the block is the leftmost, ARPS assigns a value of 2, Step 2 used if the block is not on the left.

**Dynamic ZMP technique steps:**
1: **If** *(SAD < T)*
2: $MV = [0, 0]$
3: **Stop**.
4: **else** *in the event that the current block is the border block farthest to the left.*
5: **Γ=2,**
6: **else** Γ=*Max{│MVPredicted(x) │, {│MVPredicted(y)│}* *(coordinates or arms for the rood pattern)*
7: *Go to Step 2*

**Step 2:** In order to determine the size or magnitude of the Adaptive Rood Pattern (ARP) arm, the candidate block

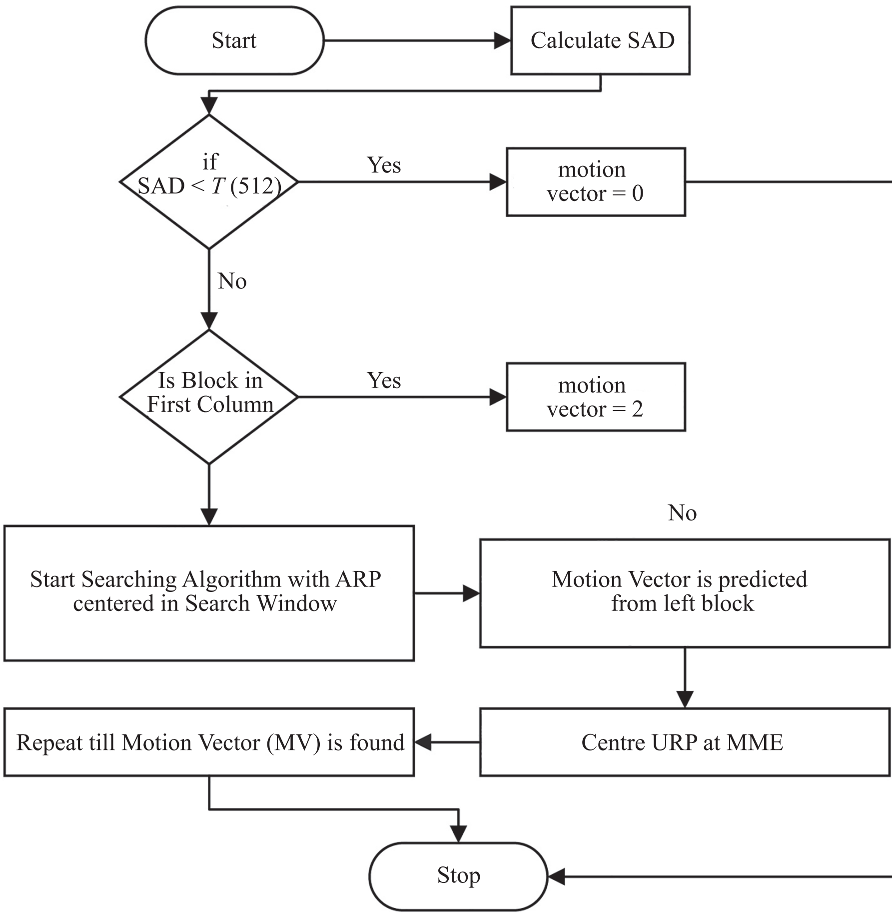


*Fig. 1.* Flowchart of ARPS algorithm

first checks for the projected motion vector from its left adjoining macroblock and then uses that motion vector by choosing the maximum of its coordinates. The four search points and the anticipated MV location are verified to determine the current Minimum Matching Error (MME) point which is aligned with the center of the ARP and the search window center point.

**Step 3:** phase three is the local search phase, which involves setting the center point of a URP at the MME point detected in Step 2 and checking its properties if the new MME point is not placed in the center of the current URP. This process is performed many times until the motion vector MV and the MME point are determined.

**Zero Motion Prejudgment in ARPS**

ZMP, which is tasked with spotting immobile blocks before the search phase, is a crucial part of the ARPS algorithm. When the candidate block receives a motion vector with the values (0, 0), indicating that there is no motion, the search is complete. It significantly reduces calculation. Using predetermined criteria, ZMP first determines the MME between the candidate block and the reference frame block [25, 26]. The matching error measure used by the ARPS algorithm is SAD, as shown in the following equation.

$$SAD = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|C_{ij} - R_{ij}|,$$

where, $n$ represents the macroblock size, $C_{ij}$ and $R_{ij}$ denote the pixels being compared in the current macroblock and reference macroblock, respectively. By calculating the SAD value, we can determine the similarity between the candidate block and the reference block. Static blocks are identified by comparing the MME (SAD) value to a fixed threshold known as the ZMP value, denoted as $T = 512$. If the SAD value is below this threshold, the block is considered stationary. The fixed ZMP threshold [27] provides a significant speed improvement without noticeably degrading the visual quality but is most effective for slow-motion video sequences. However, it poses a limitation for video sequences with extensive fast motion, since they cannot benefit substantially from this fixed threshold. Consequently, selecting an appropriate fixed threshold that suits all motion types in video sequences becomes challenging, representing the first weakness of the ARPS algorithm. To address this limitation, we propose using a threshold that can be adjusted based on the frame motion speed. For faster-motion videos, the threshold can be increased to obtain greater performance gains. In our recently proposed algorithm, we implement a dynamic ZMP threshold that adapts to the motion speed of both fast- and slow-motion videos. This approach affords greater adaptability and enhanced efficacy in a variety of motion scenarios.

**The Initial Search Step in the ARPS Algorithm**

ARP is adaptive. ARP is calculated dynamically for each macroblock depending on its predicted motion vector. Its size is determined by the motion vector of the left neighboring block. Fig. 2 shows a symmetrical adaptable rood design with four search points at the four vertices. ARP step size is the vertex-to-center distance. The four-

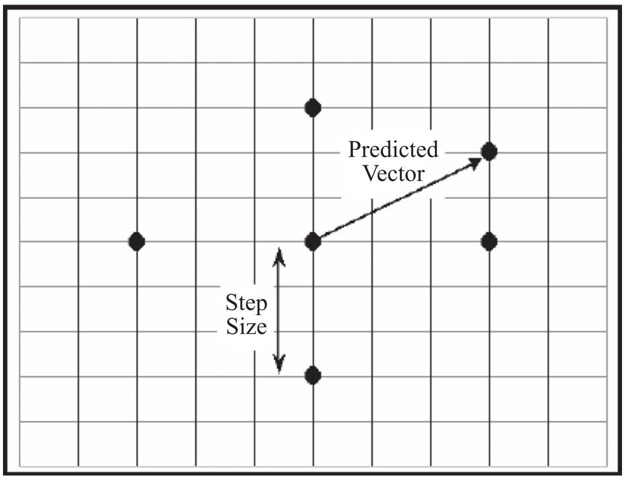


*Fig. 2.* ARPS Rood Pattern [5]

armed rood pattern vertices and the target motion vector are validated as searching points. The starting motion vector is crucial. Thus, the ARPS algorithm initial step has to choose a good local search starting point. The ARPS approach tried to find a starting point as close to the global minimum as possible, however, the algorithm had no left blocks to predict from for the leftmost boundary blocks in each frame. Accordingly, the ARP arm size is set to two pixels $\Gamma = 2$. That assumption harmed the accuracy as it does not depend on actual calculations, it is only an assumption. Besides, it may be incorrect in most cases as it has no scientific proof, for example, why the motion vector is assumed to be two pixels and not any other value. So, the ARPS algorithm has a second point of weakness which affects its performance considerably. That inaccurate starting point of the algorithm can cause the whole search process to be trapped in a local minimum problem, especially in the extended search path.

**The Refined Local Search Step in the ARPS Algorithm**

The ARPS algorithm then advances to the local search stage after choosing a starting point in the previous step. URP, which is a small, compact, and fixed-size search pattern, is applied repeatedly and randomly until MME becomes the center point of the search pattern. So, URP is applied until the ultimate motion vector is discovered. The frequent and unrestricted use of the URP by the ARPS algorithm, however, is a drawback. Using that tiny pattern search stage may gradually improve the ME. Although it may appear like this, and the approach is effective, it may need more time and computer power, which would lower the algorithm performance. The assumption of a unimodal error surface may be an issue with unlimited use of the URP. This presumption makes the unavoidable assumption that the search space has a single global minimum. Depending solely on the starting point, which may itself be inaccurate, repeatedly applying a small search pattern can restrict the algorithm ability to explore the entire search space and find the true global minimum. Consequently, this limitation can negatively impact the accuracy and reliability of the ME results of the ARPS algorithm [28, 29].

**Efficient Adaptive Rood Pattern Search algorithm — EARPS algorithm**

The proposed EARPS algorithm aims to address the limitations of the ARPS by introducing a fast and accurate block-based search technique that improves computational efficiency while ensuring image quality. EARPS improves upon the three weaknesses identified in the ARPS algorithm.

**Firstly: the problem of fixed ZMP threshold unsuitable for fast motion videos**

EARPS utilizes an adjustable or dynamic ZMP threshold. This threshold has two values: one suitable for frames with small motion and another for frames with fast motion. By adapting the threshold to the specific motion characteristics of the frames, EARPS achieves faster computations without compromising image quality. The dynamic ZMP threshold is assigned a lower value when slow motion is observed and a higher value when fast motion is observed. One common method for evaluating video frame discrepancies is the MAFD detector. This detector is used by EARPS to find instances of movement in videos. This average absolute pixel intensity difference between frames is provided by the MAFD detector. The equation below can be used to depict the MAFD calculation.

$$MAFD_n = \frac{1}{MN}\sum_{i}^{M-1}\sum_{j}^{N-1}|f_n(i,j) - f_{n-1}(i,j)|, \qquad (1)$$

where $M$ and $N$ are the width and height of the frames, $f_n(i, j)$ is the pixel intensity at position $(i, j)$ and $n$ is the frame number. At each frame transition, the MAFD, which is the first-order derivative of $f_n$, calculates the level of dissimilarity by conducting extensive experiments with various video samples. A threshold $T$ value of 14 is assigned to the MAFD detector. When the difference in pixel intensities between consecutive frames is below this threshold, indicating a small value, the motion is classified as slow. Conversely, if the difference exceeds the threshold, the motion is identified as fast. The determination of the motion type between successive frames using the MAFD detector plays a crucial role in the EARPS algorithm. It enables the algorithm to employ an adjustable ZMP threshold that varies depending on the motion speed behavior. Specifically, two predetermined threshold values are used: β for slow motion and ɣ for fast motion. The MAD minimal matching error calculated as shown in the equation below is compared to these thresholds. Static blocks have matching errors below the ZMP threshold.

$$totalMAD = \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|C_{ij} - R_{ij}|,$$

where $n$ is macroblock side, $C_{ij}$ and $R_{ij}$ are the pixels being compared in current macro block and reference macro block, respectively.

**Dynamic ZMP technique steps:**

1: **Check If** *(MAFD < 14),* **then.**
2: *declare the frame has slow motion.*
3: **Calculate** *MAD for every block in the frame.*
4: **If** *(MAFD < ZMP), threshold slow β Then*
5: ***MV* = 0**
6: **Stop** *the search.*
7: **else** *move to the modified ARPS algorithm and declare the frame has a fast motion.*
8: **Calculate** *MAD for every block in the frame.*
9: **If (***MAD < ZMP*) *threshold fast ɣ*
10: *Then MV =0, Stop the search*
11: **else** *move to the modified ARPS algorithm.*
12: *END*

**Secondly: the problem of the optimal starting point**

The EARPS algorithm effectively addresses the second critical weakness of the ARPS algorithm. It achieves this by utilizing the FS algorithm for the leftmost column blocks in each frame. The FS algorithm is an exhaustive search method that searches all pixels in the search window to determine the optimal motion vector. Due to its intensive search procedure, the FS algorithm produces highly accurate results. By applying the FS algorithm as the initial step in EARPS, the algorithm benefits from an accurate starting point that is close to the global minimum. This precise starting point significantly enhances the accuracy and computational efficiency of the proposed EARPS algorithm.

**Thirdly: the problem of unrestrictedly and repeatedly using of URP search pattern**

To overcome the third weakness, EARPS focuses the search on the nearest location to the global minimum error or the best matching point for each block. By doing so, the refined local search step using the URP is applied only for one iteration. This is because there is a high likelihood that a single iteration is sufficient to obtain the correct motion vector, thanks to the accurate searching locations determined during the initial search step. By reducing the number of iterations and computations, EARPS achieves improved computational complexity without sacrificing the accuracy of the coded video. Fig. 3 shows the steps of the proposed EARPS algorithm.

**EARPS algorithm**

1: **compute** MAFD between every two successive frames.
2: **If** *(MAFD < T), T the threshold value is set to a value =14* **then**
3: *Frames have slow motion.*
4: **else**
5: *Frames have fast motion.*
6: **Compute** *MAD at the center point of the search window MAD (0,0),*
7: **If** *(MAD <ZMP) TH Slow β or MAD<ZMPTH Fast ɣ* **then**
8: *Stop*
9: **else if** *this is the leftmost boundary block,* **then.**
10: *Search method = FS.*
11: **else** *Search method = modified ARPS*

**Contribution of the new proposed EARPS algorithm**

— The proposed EARPS method begins by determining the motion speed in each frame of the video series using a MAFD detector. After determining the motion speed in the video frames, the following stage of the

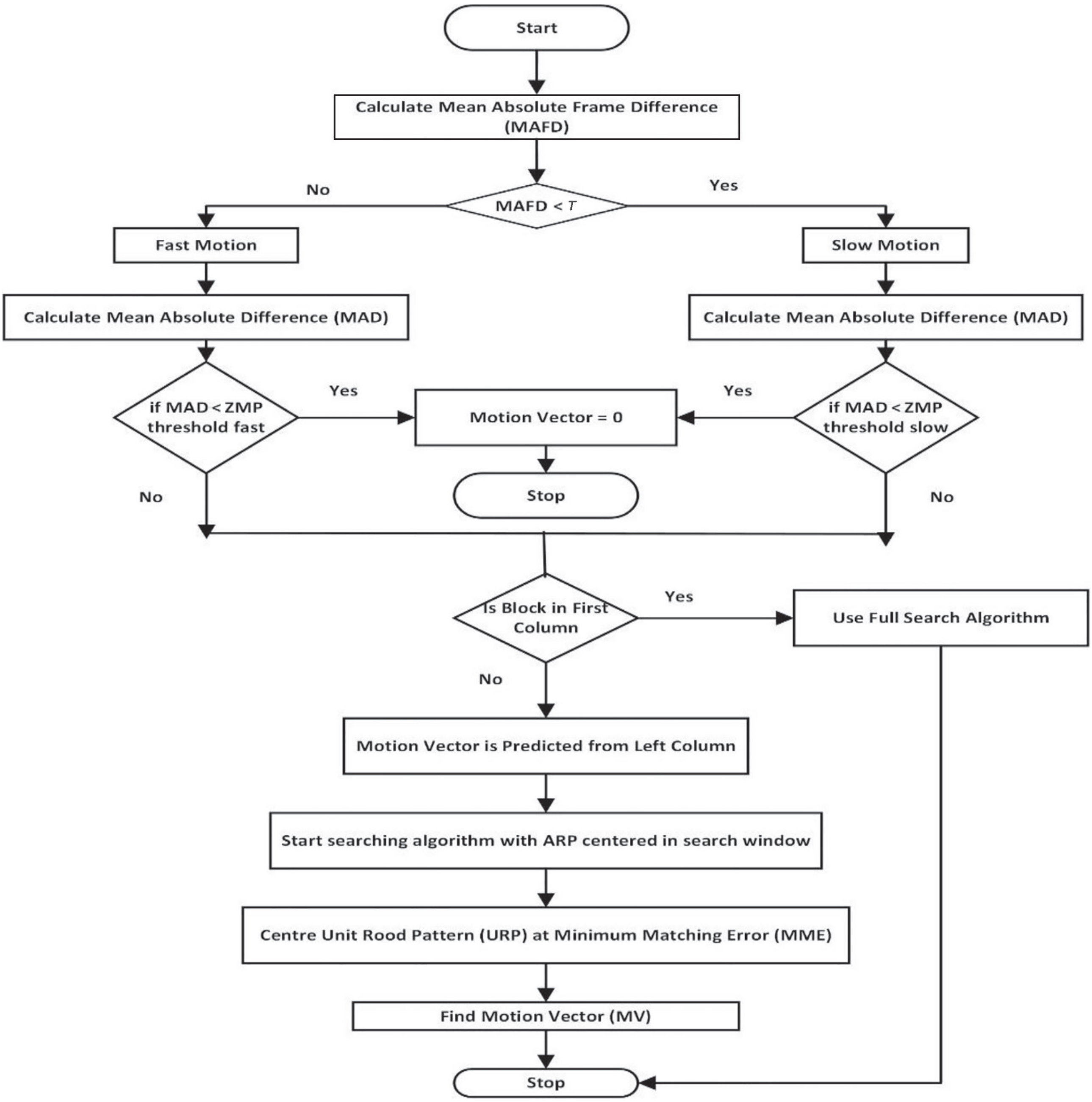


*Fig. 3*. Flowchart of EARPS algorithm

algorithm dynamic ZMP may employ the appropriate threshold, slow or fast, to speed up calculations. The new proposed algorithm makes achievement with that point, which did not exist in the ARPS algorithm. As ARPS did not use speed detectors, it used a fixed ZMP threshold for all video sequences, whatever the type of motion speed in those sequences.

— EARPS uses a dynamic threshold value that is adjusted for slow and fast motion while ARPS uses a predetermined ZMP threshold value of 512 which is suitable only for slow motion not caring for the type of motion in the video sequence. The dynamic ZMP threshold in the proposed EARPS algorithm speeds up the computations and the changing of its values helps so much in increasing speed while preserving reasonable accuracy, since it stops the searching process and declares the block to be static block in an early stage in the algorithm saving many useless computations.

— The matching criteria used are different; it is SAD in ARPS, while it is MAD in EARPS which is more complex in computations but gives more precise and accurate results as a matching criterion for blocks.

— The blocks in the leftmost column of each frame in ARPS are assigned a constant value of 2 pixels as a motion vector, as previously mentioned; that assumption harmed the accuracy, as it does not depend on actual calculations, it is only an assumption. Besides, it may be incorrect for most cases as it has no scientific proof, for example, why the motion vector is assumed to be two pixels and not any other value, while in EARPS the leftmost column blocks use the FS algorithm to calculate the motion vector. By utilizing the FS algorithm for the leftmost column blocks in each frame. The FS algorithm gives the optimal motion vector. Due to its intensive search procedure, the FS algorithm produces highly accurate results. EARPS algorithm benefits from an accurate starting point that is close to the global minimum. This precise starting point significantly enhances the accuracy and computational efficiency of the EARPS algorithm.

— The usage of the FS algorithm in the proposed EARPS algorithm (as mentioned), helps to give an accurate starting point for the search which accordingly allows the URP in the refined search stage to be used only once as the searching process was put in the correct location. In that way EARPS can speed its computations and get the accurate motion vector faster. While the URP is used unrestrictedly in the ARPS algorithm, which leads to useless computations

## Simulation Results

To achieve better results, ME algorithms compete to use fewer search positions and to add new techniques to deliver the best search results. FS algorithm evaluates $N \times N$ pixel block, within a search window with a range $w$ in both directions in the reference frame. The candidate block is compared to $N \times N$ pixel block for each of the $(2w + 1)$ 2 search places (including the current row and column of the reference frame). FS involves numerous calculations. If $w = 7$ pixels away from the current block locations that requires $15 \times 15 = 225$ search positions. Despite FS gives optimal results, yet modern microprocessors cannot do complete search with acceptable speed, particularly in real-time applications. Comparisons between the previous FS, ARPS algorithms and the proposed EARPS algorithm will be conducted using two important metrics to get the simulation results.

**The computational complexity of the algorithm**

To minimize computational complexity, all algorithms strive for fewer checking points. Prediction inaccuracy and computational complexity are used to assess fast BMA performance. By restricting checking sites, fast search pattern BMAs reduce computation. The computational complexity of the equation below may be directly compared by counting checking points.

$$\text{Average No. of Search Points (NSP)} = \frac{\text{Total Search No.}}{\text{Total No. of Blocks}}.$$

Calculation time is used to gauge the algorithm overhead, which includes storing and requesting blocks for matching, comparing blocks, etc. Speed of search is NSP. Equation below represents the speed ratio using searching points:

$$\text{Speed up ratio (NSP)} = \frac{\text{NSP (FS)}}{\text{NSP (algorithm)}}.$$

And speed-up ratio using computational time is shown as follows.

$$\text{Speed up ratio (Time)} = \frac{\text{Total Time (FS)}}{\text{Total Time (algorithm)}}.$$

**The accuracy of the encoded image**

PSNR difference is used for performance comparison. The difference in PSNR between any algorithm and FS algorithm reveals the accuracy of that algorithm. PSNR compares the predicted frame to the target frame to determine search accuracy. PSNR is calculated by the Motion Estimation Process to measure frame accuracy as follows

$$PSNR = 10\mathrm{Log}_{10}\left[\frac{(Peak\ to\ peak\ value\ of\ original\ data)^2}{MSE}\right].$$

The following criteria must be precisely specified during algorithm construction in order to achieve the search accuracy-complexity trade-off. We compared the suggested EARPS approach to ARPS in order to analyze the trade-off between computational complexity and PSNR.

**Verification of Computational Complexity for the Proposed EARPS Algorithm**

According to the first step of the proposed EARPS algorithm for the detection of the type of motion speed in every frame in the video sequence, experimentation with various values for the MAFD detector previously mentioned in equation (1), discovered that a value of 14 proved to be suitable as a threshold. This value had been previously used in well-known research related to scene changes. To validate its effectiveness, the value of 14 was applied to 20 different video sequences, and the results confirmed its suitability as a threshold. When the MAFD value calculated is below 14, it indicates slow motion, while values above 14 indicate fast motion. Therefore, the value of the threshold $T$ for MAFD = 14 can be confidently used as a threshold for detecting motion speed. So EARPS algorithm can accurately classify the motion in the video frames as either slow or fast. The MAFD detector was applied to a set of 14 videos, namely, "akiyo", "bridge close", "bridge far", "container", "hall", "mother-daughter", "missa", "news", "paris", "pencil", "waterfall", "ship", "silent", and "tempete", the absolute difference in pixel intensities between consecutive frames was consistently smaller than the threshold value of 14. Consequently, the 14 videos can be classified as slow-motion videos. On the other hand, when the MAFD detector was applied to another set of 6 videos, including "bus", "caltrain", "football", "foreman", "garden", and "stefan", the simulation results revealed that the absolute difference in pixel intensities between consecutive frames surpassed 14, As a result, these videos can be categorized as fast-motion videos. The simulation results are shown in Fig. 4, *a*, using the "akiyo" video sequence as an example of slow motion, and Fig. 4, *b*, the "bus" sequence is used as an example of a fast sequence.

We evaluated each strategy on different video sequences and calculated computational cost by averaging the number of searching positions in each frame. We put the new EARPS algorithm to the test on slow and fast motion video sequences. Several slow and fast speed video sequences were investigated. Fig. 5, *a* shows a comparison of computation points for the "silent" video sequence, representing slow motion. Similarly, Fig. 5, *b* compares computation points for the "bus" video sequence

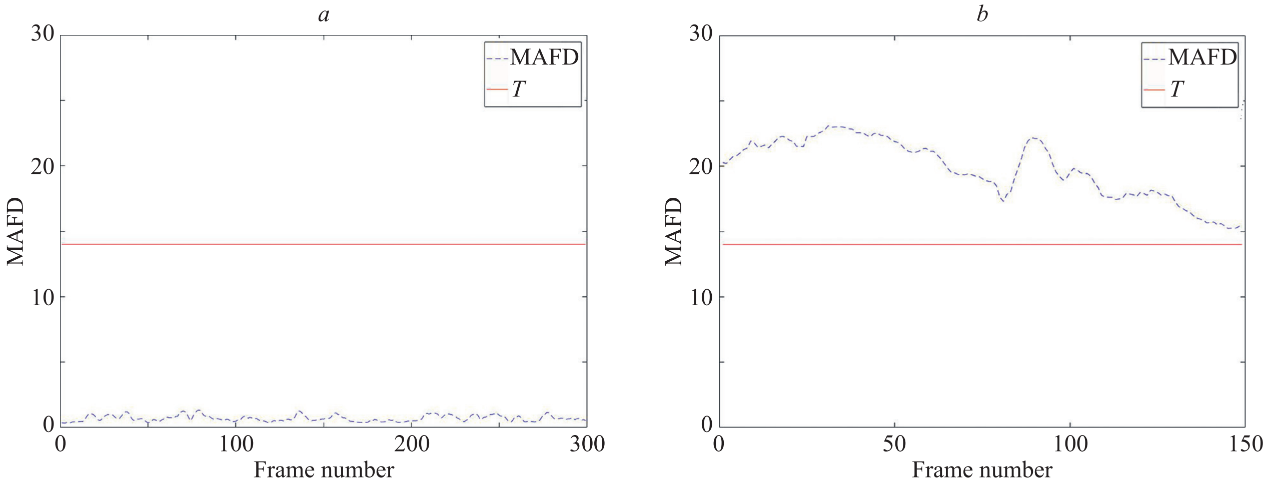


*Fig. 4*. Motion speed detection for sequences: “akiyo” (*a*) and “bus” (*b*)

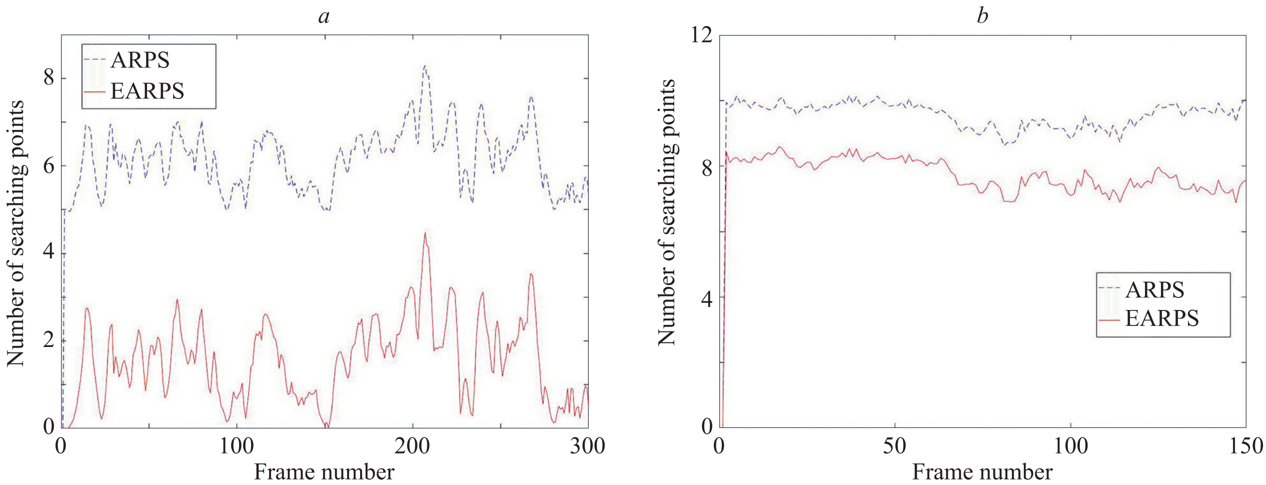


*Fig. 5*. Comparison of computations for ARPS and EARPS for: “silent” sequence (*a*); “bus” sequences (*b*)

representing fast motion. The proposed EARPS algorithm achieves the objective of minimizing computation points, as it consistently demonstrates the lowest number of points.

The simulation results in Fig. 6, *a* illustrate the impact of changing the ZMP threshold for slow motion, denoted as β. Applying the proposed algorithm to the “bridge close” video sequence as a sample of slow sequence shows that, when β is changed from 3 to 4, the number of searching points decreases resulting in faster computations, which aligns with the primary goal of the proposed algorithm.

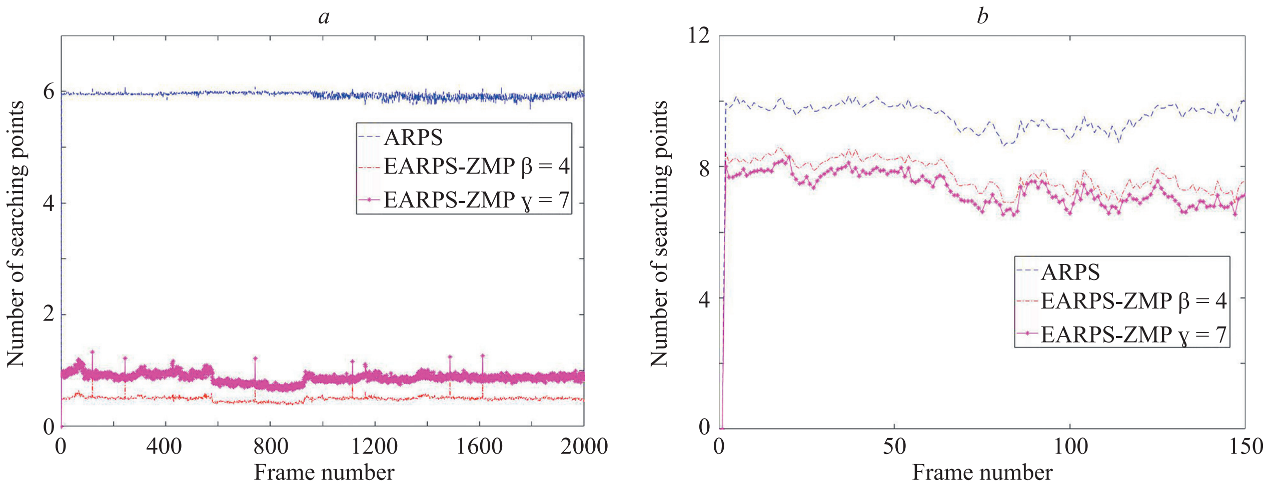


*Fig. 6*. Comparison of computations of ARPS and EARPS-ZMP algorithms: with β = 4 and 3 on the “bridge closing” video sequence (*a*); with ɣ = 4 and 7 on the “bus” video sequence (*b*)

Similarly, in Fig. 6, *b*, the ZMP threshold for fast motion, represented as ɣ, is varied between values 4 and 7. Applying the proposed algorithm to the "bus" video sequence as a sample of fast sequence shows that using ɣ equal to 7 yields better results compared to ɣ equal to 4. These simulation results demonstrate the influence of ZMP threshold values on the computational complexity of the algorithm.

Table 1 shows the average searching points per frame for each strategy on slow and fast-motion video sequences. The ZMP threshold was set by the EARPS algorithm at 3 and 4 for slow motion and 5 and 7 for fast motion. According to the table, FS has the highest average number of searching points, ARPS has less number, then the proposed EARPS has the lowest. These results demonstrate that the proposed EARPS approach has the best computational complexity, which was the algorithm main purpose. We discovered that increasing the slow motion ZMP threshold from 3 to 4 enhances EARPS performance. Calculations are faster when the average number of search places is reduced. The algorithm also enhances the search for fast-motion video sequences. Using the ZMP threshold to identify and delete static blocks early saves computational resources and simplifies the recommended approach.

## Verification of Accuracy and Video Quality for the Proposed EARPS Algorithm

PSNR is a metric used to assess video quality. The simulation results in (Fig. 7) show the "bus" video sequence which is a fast-motion video clip where EARPS produced PSNR values very near to the FS technique. Experiments show that the EARPS method outperforms the FS and ARPS algorithms in terms of calculation complexity while retaining an appropriate PSNR. For ME, the EARPS approach balanced computing complexity and PSNR. Techniques for estimating motion in video processing are improving. The EARPS algorithm increases the efficiency for video coding.

Table 2 compares PSNR for the three algorithms FS, ARPS, and EARPS when applied on various slow and fast-motion video sequences. The table shows that FS gives the highest PSNR, while ARPS and EARPS have comparable PSNR values. Additionally, the dynamic ZMP technique integrated with the proposed algorithm effectively maintains near-equivalent video quality to that

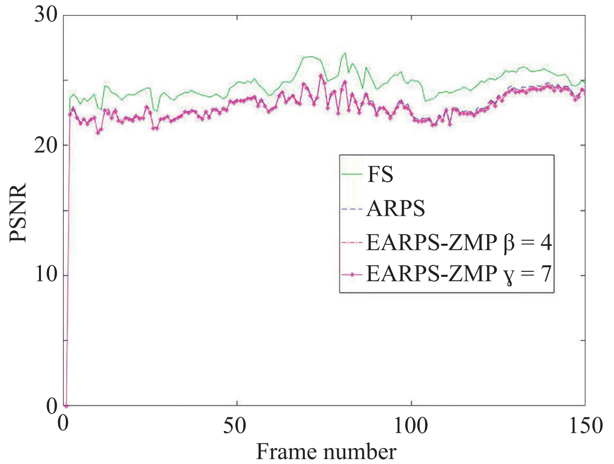


*Fig. 7.* PSNR Comparison of FS, ARPS, and EARPS algorithms on "bus" video sequence

*Table 1.* The average number of computations points per frame for different slow and fast video sequences in case β = 3, β = 4 and ɣ = 5, ɣ = 7

| Video Sequence | Type of motion speed | FS | ARPS | EARPS β = 3 Fast threshold ɣ = 5 | EARPS β= 4 Fast threshold ɣ = 7 |
|---|---|---|---|---|---|
| pencil | slow | 204.2828 | 5.2612 | 0.8883 | 0.4270 |
| bridge close | slow | 204.1807 | 5.9306 | 0.8695 | 0.4943 |
| bridge far | slow | 204.1835 | 5.9222 | 0.5055 | 0.0240 |
| container | slow | 203.6019 | 5.1312 | 0.3897 | 0.2202 |
| missa | slow | 204.2828 | 5.1772 | 0.1384 | 0.1047 |
| mother daughter | slow | 203.6419 | 6.4961 | 1.0714 | 0.8075 |
| news | slow | 203.6540 | 5.6543 | 0.8962 | 0.7431 |
| paris | slow | 204.0910 | 5.2994 | 0.8888 | 0.7537 |
| silent | slow | 203.4971 | 6.1313 | 1.8222 | 1.5591 |
| waterfall | slow | 203.4971 | 5.2428 | 3.2892 | 1.7439 |
| akiyo | slow | 203.6645 | 5.0330 | 0.2982 | 0.2154 |
| full | Fast | 204.2043 | 7.0789 | 4.8055 | 4.5304 |
| highway | Fast | 204.1807 | 8.1427 | 2.8505 | 2.1376 |
| caltrain | Fast | 204.2828 | 6.8714 | 6.2343 | 6.1345 |
| garden | Fast | 204.2888 | 8.6079 | 7.5052 | 7.4972 |
| foreman | Fast | 203.6019 | 8.8433 | 8.2325 | 8.1495 |
| bus | Fast | 202.9209 | 9.5001 | 8.3452 | 8.2351 |

*Table 2.* Comparison of PSNR values for FS, ARPS, EARPS Algorithms

| Video Sequence | FS | ARPS | EARPS Slow threshold β =3, Fast threshold ɣ = 5 | EARPS Slow threshold β =4, Fast threshold ɣ = 5 |
|---|---|---|---|---|
| container | 38.1402 | 38.0905 | 38.0883 | 38.0883 |
| hall | 34.6978 | 34.6335 | 34.6087 | 34.5842 |
| highway | 34.6529 | 33.7771 | 33.6693 | 33.5379 |
| waterfall | 34.4321 | 34.4310 | 34.4285 | 34.4050 |
| tempete | 26.5588 | 26.3933 | 26.3814 | 26.3686 |
| mother-daughter | 40.2563 | 40.1360 | 39.9699 | 39.7358 |
| news | 36.6172 | 36.3410 | 36.2431 | 36.1670 |
| paris | 33.5439 | 33.3533 | 33.3340 | 33.3019 |
| bridge-far | 38.4102 | 38.3896 | 38.3283 | 38.3217 |
| bridge-close | 35.5151 | 35.5148 | 35.5143 | 35.5142 |
| silent | 35.3268 | 34.9261 | 34.8755 | 34.8051 |
| missa | 42.5250 | 42.5197 | 42.5143 | 42.5105 |
| pencil | 39.3883 | 39.3698 | 39.2218 | 39.0969 |
| ship | 36.6025 | 36.6025 | 36.6005 | 36.6010 |
| meeting | 34.6662 | 34.6390 | 34.5957 | 34.5501 |
| stefan | 25.1846 | 24.8694 | 24.7779 | 24.7698 |
| coastguard | 30.3304 | 30.3119 | 30.3001 | 30.2491 |
| bus | 24.5111 | 22.9449 | 22.8960 | 22.8960 |
| foreman | 31.1071 | 30.8226 | 30.6819 | 30.5173 |
| garden | 25.0739 | 24.9228 | 24.8799 | 24.8799 |
| akiyo | 42.7995 | 42.7694 | 42.7253 | 42.6305 |
| football | 24.9109 | 24.4958 | 24.4666 | 24.4614 |
| caltrain | 22.5290 | 22.4767 | 22.4456 | 22.4434 |
| full | 31.2719 | 31.179 | 31.1364 | 31.1130 |

of the FS algorithm. The table shows the PSNR values when using ZMP threshold slow β = 3, ZMP threshold fast ɣ = 5. Simulation results show the PSNR values when changing the ZMP threshold value for the slow motion β to 4 while keeping ZMP threshold fast ɣ = 5. Despite changing the ZMP threshold slow motion from 3 to 4 has a good effect on the computational speed, but on the other side that causes degradation in PSNR values for some cases. For that reason, the selection of ZMP threshold values is so important and should be suitable for the applications used, as some applications need very high speed, whatever the quality is, only to be near to the optimal. In contrast, others care for the accuracy of the video quality, not caring for speed like medical applications. Therefore, the resulting PSNR values achieved when changing the ZMP threshold values for slow and fast motion, demonstrate that the selection of appropriate ZMP threshold values is crucial for balancing computational speed and video quality.

## Conclusion

In conclusion, the results of the studies reveal that the proposed EARPS algorithm outperforms both FS and ARPS in terms of computing complexity while maintaining an acceptable degree of PSNR accuracy. This result indicates the effectiveness of the proposed strategy in balancing computational complexity and PSNR. It also underlines the technique advantages. In summary, the proposed EARPS approach contributes to the development of more exact and economical motion estimating algorithms for video processing applications.

## References


1. Adapa V.P., Vuyyuru A. A survey on block matching algorithms for video coding. *International Journal of Electrical and Computer Engineering (IJECE)*, 2017. vol. 7, no. 1, pp. 216–224. https://doi.org/10.11591/ijece.v7i1.pp216-224
2. Manikandan L.C., Nair S.A.H., Sanal Kumar K.P., Selvakumar R.K. A study and analysis on block matching algorithms for motion estimation in video coding. *Cluster Computing*, 2019, vol. 22, no. S5, pp. 11773–11780. https://doi.org/10.1007/s10586-017-1478-z
3. Srinivas Rao K., Paramkusam A.V. Block matching algorithms for the estimation of motion in image sequences. *Pattern Recognition and Image Analysis*, 2022, vol. 32, no. 1, pp. 33–44. https://doi.org/10.1134/s1054661822010072
4. Agha S., Khan M., Jan F. Efficient fast motion estimation algorithm for real-time applications. *Journal of Real-Time Image Processing*, 2022, vol. 19, no. 2, pp. 403–413. https://doi.org/10.1007/s11554-021-01188-7
5. Sehairi K., Chouireb F., Meunier J. Comparative study of motion detection methods for video surveillance systems. *Journal of Electronic Imaging*, 2017, vol. 26, no. 2, pp. 023025. https://doi.org/10.1117/1.jei.26.2.023025
6. Pang S., Zhang X., Li H., Lu Y. Robust block-matching algorithm for motion estimation using an anti-interference similarity criterion and the bilateral optimization scheme. *Applied Optics*, 2021, vol. 60, no. 16, pp. 4746. https://doi.org/10.1364/ao.424238
7. Ravi P.R. Diamond search optimization-based technique for motion estimation in video compression. *International Journal of e-collaboration*, 2023, vol. 19, no. 3, pp. 1–4. https://doi.org/10.4018/ijec.316773
8. Jiao S., Wang Y., Luan L., Yu X. Research on fast motion estimation in H264 coding. *Proceedings of SPIE*, 2022, vol. 12083, pp. 120832T. https://doi.org/10.1117/12.2623419
9. Pan Z., Zhang R., Ku W., Wang Y. Adaptive pattern selection strategy for diamond search algorithm in fast motion estimation. *Multimedia Tools and Applications*, 2019, vol. 78, no. 2, pp. 2447–2464. https://doi.org/10.1007/s11042-018-6353-2
10. Amirpour H., Mousavinia A. A dynamic search pattern motion estimation algorithm using prioritized motion vectors. *Signal, Image and Video Processing*, 2016, vol. 10, no. 8, pp. 1393–1400. https://doi.org/10.1007/s11760-016-0906-5
11. Arnaudov P., Ogunfunmi T. Dynamically adaptive fast motion estimation algorithm for HD video. *Journal of Signal Processing Systems*, 2020, vol. 92, no. 10, pp. 1115–1131. https://doi.org/10.1007/s11265-020-01574-7
12. Vanshree V. A novel adaptive rood pattern search algorithm. *IOSR Journal of Electrical and Electronics Engineering*, 2013, vol. 4, no. 4, pp. 14–18. https://doi.org/10.9790/1676-0441418
13. Schinde T. Adaptive pixel-based direction oriented fast motion estimation for predictive coding. *Proc. of the 2022 Picture Coding Symposium (PCS)*, 2022, pp. 331–335. https://doi.org/10.1109/pcs56426.2022.10018044
14. Nie Y., Ma K.-K. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 2002, vol. 11, no. 12, pp. 1442–1449. https://doi.org/10.1109/tip.2002.806251
15. Madhuvappan C.A., Ramesh J. An efficient VLSI architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant depended on search algorithm in HEVC. *Wireless Personal Communications*, 2023, vol. 130, no. 4, pp. 2305–2325. https://doi.org/10.1007/s11277-023-10237-2
16. Shajin F.H., Rajesh P., Raja M.R. An efficient VLSI architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant-based search algorithm in HEVC. *Circuits, Systems, and Signal Processing*, 2022, vol. 41, no. 3, pp. 1751–1774. https://doi.org/10.1007/s00034-021-01850-2
17. Arora S.M., Rajpal N., Khanna K. A new approach with enhanced accuracy in ZMP for motion estimation in real time applications. *Journal of Real-Time Image Processing*, 2019, vol. 16, no. 4, pp. 989–1005. https://doi.org/10.1007/s11554-016-0593-z
18. Yi X., Ling N. Fast pixel-based video scene change detection. *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 3443–3446. https://doi.org/10.1109/iscas.2005.1465369
19. Mukherjee R., Biswas B., Vinod I.G. Half-way stop adaptive pattern search algorithm for motion estimation and dedicated VLSI


## Литература


1. Adapa V.P., Vuyyuru A. A survey on block matching algorithms for video coding // International Journal of Electrical and Computer Engineering (IJECE). 2017. V. 7. N 1. P. 216–224. https://doi.org/10.11591/ijece.v7i1.pp216-224
2. Manikandan L.C., Nair S.A.H., Sanal Kumar K.P., Selvakumar R.K. A study and analysis on block matching algorithms for motion estimation in video coding // Cluster Computing. 2019. V. 22. N S5. P. 11773–11780. https://doi.org/10.1007/s10586-017-1478-z
3. Srinivas Rao K., Paramkusam A.V. Block matching algorithms for the estimation of motion in image sequences // Pattern Recognition and Image Analysis. 2022. V. 32. N 1. P. 33–44. https://doi.org/10.1134/s1054661822010072
4. Agha S., Khan M., Jan F. Efficient fast motion estimation algorithm for real-time applications // Journal of Real-Time Image Processing. 2022. V. 19. N 2. P. 403–413. https://doi.org/10.1007/s11554-021-01188-7
5. Sehairi K., Chouireb F., Meunier J. Comparative study of motion detection methods for video surveillance systems // Journal of Electronic Imaging. 2017. V. 26. N 2. P. 023025. https://doi.org/10.1117/1.jei.26.2.023025
6. Pang S., Zhang X., Li H., Lu Y. Robust block-matching algorithm for motion estimation using an anti-interference similarity criterion and the bilateral optimization scheme // Applied Optics. 2021. V. 60. N 16. P. 4746. https://doi.org/10.1364/ao.424238
7. Ravi P.R. Diamond search optimization-based technique for motion estimation in video compression // International Journal of e-collaboration. 2023. V. 19. N 3. P. 1–4. https://doi.org/10.4018/ijec.316773
8. Jiao S., Wang Y., Luan L., Yu X. Research on fast motion estimation in H264 coding // Proceedings of SPIE. 2022. V. 12083. P. 120832T. https://doi.org/10.1117/12.2623419
9. Pan Z., Zhang R., Ku W., Wang Y. Adaptive pattern selection strategy for diamond search algorithm in fast motion estimation // Multimedia Tools and Applications. 2019. V. 78. N 2. P. 2447–2464. https://doi.org/10.1007/s11042-018-6353-2
10. Amirpour H., Mousavinia A. A dynamic search pattern motion estimation algorithm using prioritized motion vectors // Signal, Image and Video Processing. 2016. V. 10. N 8. P. 1393–1400. https://doi.org/10.1007/s11760-016-0906-5
11. Arnaudov P., Ogunfunmi T. Dynamically adaptive fast motion estimation algorithm for HD video // Journal of Signal Processing Systems. 2020. V. 92. N 10. P. 1115–1131. https://doi.org/10.1007/s11265-020-01574-7
12. Vanshree V. A novel adaptive rood pattern search algorithm // IOSR Journal of Electrical and Electronics Engineering. 2013. V. 4. N 4. P. 14–18. https://doi.org/10.9790/1676-0441418
13. Schinde T. Adaptive pixel-based direction oriented fast motion estimation for predictive coding // Proc. of the 2022 Picture Coding Symposium (PCS). 2022. P. 331–335. https://doi.org/10.1109/pcs56426.2022.10018044
14. Nie Y., Ma K.-K. Adaptive rood pattern search for fast block-matching motion estimation // IEEE Transactions on Image Processing. 2002. V. 11. N 12. P. 1442–1449. https://doi.org/10.1109/tip.2002.806251
15. Madhuvappan C.A., Ramesh J. An efficient VLSI architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant depended on search algorithm in HEVC // Wireless Personal Communications. 2023. V. 130. N 4. P. 2305–2325. https://doi.org/10.1007/s11277-023-10237-2
16. Shajin F.H., Rajesh P., Raja M.R. An efficient VLSI architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant-based search algorithm in HEVC // Circuits, Systems, and Signal Processing. 2022. V. 41. N 3. P. 1751–1774. https://doi.org/10.1007/s00034-021-01850-2
17. Arora S.M., Rajpal N., Khanna K. A new approach with enhanced accuracy in ZMP for motion estimation in real time applications // Journal of Real-Time Image Processing. 2019. V. 16. N 4. P. 989–1005. https://doi.org/10.1007/s11554-016-0593-z
18. Yi X., Ling N. Fast pixel-based video scene change detection // Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS). 2005. P. 3443–3446. https://doi.org/10.1109/iscas.2005.1465369
19. Mukherjee R., Biswas B., Vinod I.G. Half-way stop adaptive pattern search algorithm for motion estimation and dedicated VLSI

architecture. *Circuits, Systems, and Signal Processing*, 2023, vol. 42, no. 3, pp. 1617–1638. https://doi.org/10.1007/s00034-022-02182-5
20. Mishra A.K., Kohli N. Enhanced adaptive threshold algorithm with weighted search points for fast motion estimation. *International Journal of Information Technology*, 2023, vol. 15, no. 2, pp. 845–857. https://doi.org/10.1007/s41870-022-01067-9
21. Deepa M.T. PSNR based analysis of block matching algorithms for motion estimation. *International Journal of Scientific & Engineering Research*, 2013, vol. 4, no. 8, pp. 5518.
22. Mukkerjee R., Puniyani A. A fast adaptive algorithm for sub-pixel motion estimation. *Proc. of the IEEE Region 10 Symposium (TENSYMP)*, 2022, pp. 1–5. https://doi.org/10.1109/tensymp54529.2022.9864538
23. Arora S.M., Rajpal N., Purwar R. Dynamic pattern search algorithm with zero motion prejudgment for fast motion estimation. *Proc. of the Fifth International Conference on Advanced Computing & Communication Technologies*, 2015, pp. 138–142. https://doi.org/10.1109/acct.2015.133
24. Rippel O., Nair S., Lew C., Branson S., Anderson A., Bourdev L. Learned video compression. *Proc. of the IEEE/CVF International Conference On Computer Vision*, 2019, pp. 3454–3463. https://doi.org/10.1109/iccv.2019.00355
25. Kerfa D., Saidane A. An efficient algorithm for fast block matching motion estimation using an adaptive threshold scheme. *Proceedings of SPIE*, 2019, vol. 10996, pp. 109960C. https://doi.org/10.1117/12.2505732
26. Salih Y.A., George L.E. Improved hybrid block-based motion estimation for inter-frame coding. *Circuits, Systems, and Signal Processing*, 2021, vol. 40, no. 7, pp. 3500–3522. https://doi.org/10.1007/s00034-020-01637-x
27. Chatterjee S.K., Vittapu S.K., Kundu S. Prediction-biased diamond search algorithm: a new approach to reduce motion estimation complexity. *Microsystem Technologies*, 2021, vol. 27, no. 5, pp. 2027–2032. https://doi.org/10.1007/s00542-020-05167-z
28. Roberto E Souza M., Maia H.D.A., Pedrini H. Survey on digital video stabilization: concepts, methods, and challenges. *ACM Computing Surveys*, 2023, vol. 55, no. 3, pp. 1–37. https://doi.org/10.1145/3494525
29. Kidani Y., Unno K., Kawamura K., Watanabe H. Memory bandwidth constrained overlapped block motion compensation for video coding. *ITE Transactions on Media Technology and Applications*, 2023, vol. 11, no. 1, pp. 1–12. https://doi.org/10.3169/mta.11.1

architecture // Circuits, Systems, and Signal Processing. 2023. V. 42. N 3. P. 1617–1638. https://doi.org/10.1007/s00034-022-02182-5
20. Mishra A.K., Kohli N. Enhanced adaptive threshold algorithm with weighted search points for fast motion estimation // International Journal of Information Technology. 2023. V. 15. N 2. P. 845–857. https://doi.org/10.1007/s41870-022-01067-9
21. Deepa M.T. PSNR based analysis of block matching algorithms for motion estimation // International Journal of Scientific & Engineering Research. 2013. V. 4. N 8. P. 5518.
22. Mukkerjee R., Puniyani A. A fast adaptive algorithm for sub-pixel motion estimation // Proc. of the IEEE Region 10 Symposium (TENSYMP). 2022. P. 1–5. https://doi.org/10.1109/tensymp54529.2022.9864538
23. Arora S.M., Rajpal N., Purwar R. Dynamic pattern search algorithm with zero motion prejudgment for fast motion estimation // Proc. of the Fifth International Conference on Advanced Computing & Communication Technologies. 2015. P. 138–142. https://doi.org/10.1109/acct.2015.133
24. Rippel O., Nair S., Lew C., Branson S., Anderson A., Bourdev L. Learned video compression // Proc. of the IEEE/CVF International Conference On Computer Vision. 2019. P. 3454–3463. https://doi.org/10.1109/iccv.2019.00355
25. Kerfa D., Saidane A. An efficient algorithm for fast block matching motion estimation using an adaptive threshold scheme // Proceedings of SPIE. 2019. V. 10996. P. 109960C. https://doi.org/10.1117/12.2505732
26. Salih Y.A., George L.E. Improved hybrid block-based motion estimation for inter-frame coding // Circuits, Systems, and Signal Processing. 2021. V. 40. N 7. P. 3500–3522. https://doi.org/10.1007/s00034-020-01637-x
27. Chatterjee S.K., Vittapu S.K., Kundu S. Prediction-biased diamond search algorithm: a new approach to reduce motion estimation complexity // Microsystem Technologies. 2021. V. 27. N 5. P. 2027–2032. https://doi.org/10.1007/s00542-020-05167-z
28. Roberto E Souza M., Maia H.D.A., Pedrini H. Survey on digital video stabilization: concepts, methods, and challenges // ACM Computing Surveys. 2023. V. 55. N 3. P. 1–37. https://doi.org/10.1145/3494525
29. Kidani Y., Unno K., Kawamura K., Watanabe H. Memory bandwidth constrained overlapped block motion compensation for video coding // ITE Transactions on Media Technology and Applications. 2023. V. 11. N 1. P. 1–12. https://doi.org/10.3169/mta.11.1

## Authors

**Sherin A. Shaker** — Magister, PhD Student, University Utara Malaysia, Sintok, 06000, Malaysia, https://orcid.org/0000-0002-2501-7584, Shereenshaker2004@yahoo.com

**Ahmad Suki C. M. Arif** – PhD, Lecturer, University Utara Malaysia, Sintok, 06000, Malaysia, https://orcid.org/0009-0008-2724-1281, suki1207@uum.edu.my

**Yousef Fazea** — PhD, Assistant Professor, Marshall University, Huntington, 25755, United States of America, sc 56803894200, https://orcid.org/0000-0003-3544-2434, fazeaalnades@marshall.edu

## Авторы

**Шейкер Абделаал Шерин** — магистр, аспирант, Университет Утра Малайзии, Синток, 06000, Малайзия, https://orcid.org/0000-0002-2501-7584, Shereenshaker2004@yahoo.com

**Ариф Ахмад Суки Че Мохамед** — PhD, преподаватель, преподаватель, Университет Утра Малайзии, Синток, 06000, Малайзия, https://orcid.org/0009-0008-2724-1281, suki1207@uum.edu.my

**Фазеа Юсеф** — PhD, доцент, Университет Маршалл, Хантингтон, 25755, Соединенные Штаты Америки, sc 56803894200, https://orcid.org/0000-0003-3544-2434, fazeaalnades@marshall.edu

*Received 04.04.2023*
*Approved after reviewing 17.07.2023*
*Accepted 17.09.2023*

*Статья поступила в редакцию 04.04.2023*
*Одобрена после рецензирования 17.07.2023*
*Принята к печати 17.09.2023*