

Velocity and Direction-Based Motion Estimation in Video Compression to Reduce Computation and Increase Video Quality

Dadvar Hosseini Avashanagh¹  | Mehdi Nooshyar²  | Saeed Barghandan³  | Majid Ghandchi⁴ 

¹Department of Electrical Engineering, Islamic Azad University, Ahar Branch, Ahar, Iran.
dadvar.hosseini@iau.ir

²Department of Electrical and Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran.
nooshyar@uma.ac.ir

³Department of Electrical Engineering, Islamic Azad University, Ahar Branch, Ahar, Iran.
Saeed.Barghandan@iau.ac.ir

⁴Department of Electrical Engineering, Islamic Azad University, Ahar Branch, Ahar, Iran.
majid.ghandchi@iau.ac.ir

Correspondence

Mehdi Nooshyar, Associate Professor of Electrical and Computer Engineering, University of Mohaghegh, Ardabil, Iran.
Email: nooshyar@uma.ac.ir

Main Subjects:

Video Processing

Paper History:

Received: 29 August 2024

Revised: 11 October 2024

Accepted: 14 October 2024

Abstract

High-Efficiency Video Coding (HEVC) represents the latest advancement in video compression standards, offering significantly improved efficiency by supporting high-definition (HD) videos with double the compression capability of the H.264 standard. However, motion estimation within HEVC remains the most computationally intensive aspect of the encoding process, accounting for the majority of the required processing time. Although various methods have been developed to mitigate this complexity, and many have demonstrated practical effectiveness, the processing demands continue to pose challenges for real-time applications. This research introduces a fast sub-pixel motion estimation algorithm designed to minimize the number of search points during video encoding. The proposed approach leverages a mathematical model grounded in motion physics and image characteristics across consecutive frames, incorporating statistical insights into the movement of elements within the video sequence. By reducing the number of search points, the algorithm lowers computational complexity while maintaining or even enhancing relevant video quality metrics.

Keywords: Pixels, Motion estimation, Compression, Video coding, HEVC.

Highlights

- Reducing the number of search points in video motion estimation.
- Reducing the complexity of motion estimation calculations.
- Increasing the speed of motion estimation calculations.
- Increasing the quality of the final image and video: This increases in quality in terms of PSNR-HVS and PEVQ-MOS made the image better from the point of view of human vision.

Citation: D. Hosseini Avashanagh, M. Nooshyar, S. Barghandan, and M. Ghandchi, "Velocity and Direction-Based Motion Estimation in Video Compression to Reduce Computation and Increase Video Quality," *Journal of Southern Communication Engineering*, vol. 15, no. 57, pp. 28-44, 2025, doi:10.30495/jce.2025.1993480.1340 [in Persian].

COPYRIGHTS

©2025 by the authors. Published by the Islamic Azad University Bushehr Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0>



1. Introduction

High Efficiency Video Coding (HEVC) stands at the forefront of international video compression standards, delivering nearly double the compression efficiency of its predecessor, H.264/AVC, while supporting high-definition video formats. Nevertheless, Motion Estimation (ME) remains the most computationally demanding stage in HEVC encoding, accounting for up to 90% of the total encoding time. Conventional ME algorithms, including the Three-Step Search (TSS) and its derivatives, aim to reduce the search space but often face challenges in preserving accuracy, especially in high-motion environments.

To address these limitations, this study introduces the **Direction and Velocity-Based Fast Motion Estimation (DVBFME)** algorithm, inspired by principles of human visual perception. The proposed approach predicts macroblock displacement by leveraging both motion speed and direction, enabling efficient sub-pixel motion searches with fewer candidate points and lower computational complexity. This method not only accelerates the encoding process but also maintains competitive video quality, offering a practical solution for real-time applications.

2. Innovation and Contributions

The key contributions of this research include:

- **Novel ME Algorithm:** Development of an innovative ME algorithm that utilizes motion direction and velocity to select predictive search points.
- **Efficiency in Search Points:** Significant reduction in the number of search points during sub-pixel refinement without sacrificing accuracy.
- **Human Visual System Integration:** Incorporation of motion prediction techniques inspired by the human visual system to enhance search efficiency.
- **Computational Savings:** Demonstration of considerable computational savings while either maintaining or improving the Peak Signal-to-Noise Ratio (PSNR) in reconstructed video.

3. Materials and Methods

The DVBFME algorithm operates through a maximum of four stages, typically converging within three:

- **Stage 1:** The initial frame of each video sequence is encoded as an intra-frame reference.
- **Stage 2:** Motion vectors between the second frame and the reference are estimated using a modified Three-Step Search (TSS) with reduced search points. Stationary macroblocks are identified when the best match is located at the center of the search window. For these blocks, only minimal neighborhood searches are conducted in subsequent frames.
- **Stage 3:** For moving macroblocks, the magnitude (pixels/frame) and direction of displacement are calculated based on previous motion vectors. The search window for the next frame is centered accordingly, followed by sub-pixel refinement to improve accuracy.
- **Stage 4:** If satisfactory matching is not achieved, the algorithm reverts to Stage 2 to ensure robustness.

This approach leverages spatial and temporal motion consistency, particularly in sequences with predictable motion patterns (e.g., videoconferencing), where multiple macroblocks share similar motion characteristics. By transmitting group motion vectors instead of individual ones, the algorithm reduces bitrate while maintaining encoding efficiency.

4. Results and Discussion

The DVBFME algorithm was implemented in MATLAB 2022a and tested on standard 1080p FHD sequences, including Kimono, Trevor, Caltrain, Coastguard, Missa, and Surfside. All experiments were conducted on a workstation equipped with an Intel Core i7-3632QM CPU, 8 GB RAM, and Intel HD Graphics 4000 GPU. Compared to traditional TSS, the proposed method achieved:

PSNR Improvement: An increase in PSNR ranging from 0.2 dB to 1.5 dB across the test sequences.

Bitrate Reduction: More efficient transmission of motion vectors leading to reduced bitrate.

Complexity Reduction: A decrease in computational complexity by up to 35× in certain sequences, nearing real-time processing capabilities.

The performance improvements are attributed to the predictive search initialization, which reduces redundant search points while maintaining matching accuracy. The balance between encoder-side complexity and decoder simplicity makes this method suitable for low-bitrate, bandwidth-constrained applications.

5. Conclusion

This research introduces a motion estimation algorithm inspired by human visual perception that significantly reduces computational load while maintaining video quality in HEVC encoding. By predicting macroblock displacement through motion direction and velocity analysis, the **Direction and Velocity-Based Fast Motion Estimation (DVBFME)** method achieves notable efficiency in sub-pixel refinement. The approach leverages

spatial and temporal consistency in motion patterns to minimize redundant searches, optimizing both encoding speed and resource usage without compromising visual fidelity.

Declaration of Competing Interest: Authors do not have conflict of interest. The content of the paper is approved by the authors.

Author Contributions: **Dadvar Hosseini Avashanagh:** Simulation and software and writing of the article, **Mehdi Noshyar:** Simulation and software, **Saeed Burghandan:** Manuscript editing, **Majid Ghandchi:** Manuscript editing.

Open Access: Journal of Southern Communication Engineering is an open access journal. All papers are immediately available to read and reuse upon publication.

تخمین حرکت مبتنی بر سرعت و جهت در فشرده‌سازی ویدیو برای کاهش محاسبات و افزایش کیفیت ویدیو

داور حسینی اوشانق^۱ | مهدی نوشیار^۲ | سعید برغندان^۳ | مجید قندچی^۴

چکیده:

کدگذاری ویدیویی با کارایی بالا (HEVC) آخرین استاندارد کدگذاری ویدیویی است. این استاندارد با ارائه دو برابر راندمان فشرده‌سازی در مقایسه با استاندارد H.264، از فیلم‌های با وضوح بالا (HD) پشتیبانی می‌کند. محاسبات تخمین حرکت در HEVC پیچیده‌ترین قسمت از کدینگ ویدیو است که بیش‌ترین زمان کدینگ ویدیو را به این پردازش اختصاص می‌دهد. روش‌های زیادی برای کاهش زمان تخمین حرکت پیشنهاد شده که بسیاری از آن‌ها به‌صورت عملی مؤثر بوده است. با وجود به‌کارگیری الگوریتم‌های مؤثر در تخمین حرکت باز هم مقدار زمان پردازش نسبت به زمان آنی (real time) بسیار زیاد است. در این تحقیق یک الگوریتم تخمین حرکت زیر پیکسل سریع با تعداد نقاط جستجوی کمتر برای کدینگ ویدیو پیشنهاد شده است. روش پیشنهادی بر پایه مدلی از فیزیک حرکت و ویژگی‌های تصویر در قاب‌های متوالی است که از اطلاعات آماری حرکت عناصر قاب‌های دنباله‌ی ویدیویی استفاده کرده است. این الگوریتم با کاهش تعداد نقاط جستجو و بهبود نسبی پارامترهای کیفی ویدیو، پیچیدگی محاسبات را کاهش می‌دهد.

کلیدواژه‌ها: پیکسل، تخمین حرکت، فشرده‌سازی، کدینگ ویدیو، HEVC.

^۱گروه مهندسی برق، دانشگاه آزاد اسلامی، واحد اهر، اهر، ایران.

dadvar.hosseini@iau.ir

^۲گروه مهندسی برق و کامپیوتر، دانشگاه محقق اردبیلی، اردبیل، ایران.

nooshyar@uma.ac.ir

^۳گروه مهندسی برق، دانشگاه آزاد اسلامی، واحد اهر، اهر، ایران.

Saeed.Barghandan@iau.ac.ir

^۴گروه مهندسی برق، دانشگاه آزاد اسلامی، واحد اهر، اهر، ایران.

majid.ghandchi@iau.ac.ir

نویسنده مسئول:

^{*}مهدی نوشیار، دانشیار گروه مهندسی برق و کامپیوتر، دانشگاه محقق اردبیلی.

nooshyar@uma.ac.ir

موضوع اصلی:

پردازش ویدئو

تاریخچه مقاله:

تاریخ دریافت: ۸ شهریور ۱۴۰۳

تاریخ بازنگری: ۲۰ مهر ۱۴۰۳

تاریخ پذیرش: ۲۳ مهر ۱۴۰۳

تازه‌های تحقیق:

- کاهش تعداد نقاط جستجو در تخمین حرکت ویدیو.
- کاهش پیچیدگی محاسبات تخمین حرکت.
- افزایش سرعت محاسبات تخمین حرکت.
- افزایش کیفیت تصویر و ویدیو نهایی، که از نظر PSNR-HVS و PEVQ-MOS باعث بهبود از دیدگاه بینایی انسان گردید.

COPYRIGHTS

©2025 by the authors. Published by the Islamic Azad University Bushehr Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0>



۱-مقدمه

تخمین و جبران حرکت (ME/MC)^۱، قسمت اصلی در فشرده‌سازی ویدیو است که بهترین بلوک تطبیق‌یافته را در قاب‌های مرجع برای کاهش افزونگی زمانی بین قاب‌های متوالی پیدا می‌کند. جابجایی بین بهترین بلوک تطبیق‌یافته و بلوک فعلی، که به‌عنوان بردار حرکت (MV)^۲ شناخته می‌شود، توسط تخمین حرکت تولید می‌شود [۱].

تخمین حرکت، پیچیده‌ترین قسمت محاسباتی در سیستم‌های کدگذاری ویدیویی است و ۵۰ تا ۹۰ درصد پیچیدگی محاسباتی را به خود اختصاص می‌دهد [۲]. برای غلبه بر محاسبات سنگین ناشی از تعداد زیاد نقاط جستجو، الگوریتم‌های جستجوی سریع متعددی ارائه شده‌اند. به‌عنوان مثال، می‌توان به الگوریتم‌های تخمین و جبران حرکت مبتنی بر شبکه‌های عصبی (MEMC)^۳، (Ne) [۳]، جستجوی سه‌مرحله‌ای (TSS)^۴ [۴]، جستجوی سه‌مرحله‌ای جدید (NTSS)^۵ [۵]، جستجوی چهارمرحله‌ای (FSS)^۶ [۶]، جستجوی الماس (DS)^۷ [۷]، جستجوی شش‌ضلعی (HS)^۸ [۸]، جستجوی منطقه آزمایشی (TZS)^۹ [۹]، جستجوی الماس اصلاح‌شده (MDS)^{۱۰} [۱۰]، جستجوی الگوی ریشه تطبیقی (ARPS)^{۱۱} [۱۱]، جستجوی منطقه آزمایشی اصلاح‌شده (MTZS)^{۱۲} [۱۲] و جستجوی شبکه شش‌ضلعی (HGS)^{۱۳} [۱۳] اشاره کرد. تخمین حرکت پویا و تکامل (DMEE)^{۱۴} [۱۴] نیز از جمله این الگوریتم‌هاست. بسیاری از این روش‌ها با کاهش تعداد نقاط جستجو، عملکرد بهتری ارائه می‌دهند [۱۵].

کدگذارهای استاندارد ویدیو به‌طور گسترده در برنامه‌های مختلف موبایل و تلویزیون استفاده می‌شوند و اجرای کارآمد آن‌ها مسئله‌ای حیاتی برای موفقیت برنامه‌های کاربردی است. در این کدگذارها، تخمین حرکت، افزونگی‌های زمانی را بین قاب‌های متوالی یک توالی ویدیو حذف می‌کند و این عمل بیش‌ترین مقدار فشردگی را بر سیستم فشرده‌ساز اعمال می‌کند [۱۶]. برای کاهش پیچیدگی محاسبات و افزایش سرعت پردازش به‌منظور نزدیک شدن به زمان آئی، می‌توان از قابلیت‌های سیستم بینایی انسان بهره برد. بررسی‌های متعدد نشان داده‌اند که پردازش در یک "مسیر حرکت" حداقل در دو مرحله رخ می‌دهد. اولین مرحله، قشر بینایی اولیه (ناحیه V1) را تشکیل می‌دهد و اطلاعات از آنجا به ناحیه بصری زمانی میانی (MT یا V5) منتقل می‌شود. این انتقال به ناحیه MT کمک می‌کند تا تعدادی از یافته‌های روانی را برای تجسم مدل قبلی حرکت و توضیح درک سرعت و جهت در الگوهای چهارطرفه استفاده کند. این ناحیه از مدل فعلی برای اهداف آینده در یک سرعت تخمینی (درک شده) استفاده می‌کند و پاسخ‌های نورون‌های ناحیه MT را محاسبه می‌نماید [۱۷].

در دنیای واقعی، حرکت اجسام به‌گونه‌ای انجام می‌گیرد که حرکت را می‌توان دنبال کرد. هرچه سرعت حرکت نسبت به زمان بیشتر باشد، محاسبات سنگینی برای تخمین درست از جهت و مسیر حرکت لازم است. سیستم بینایی و محاسبات ذهنی انسان حرکت اشیاء را به‌گونه‌ای حساب می‌کند که با تخمین سرعت و جهت حرکت، تقریبی از مقصد نهایی آن را به‌دست می‌آورد. از این‌رو، با به‌دست آوردن سرعت و جهت حرکت عناصر در زمان فعلی، می‌توان موقعیت این عناصر را در زمان‌های آینده پیش‌گویی کرد.

سیستم بینایی انسان از تعداد زیادی قاب در مقیاس زمان بهره می‌برد، بنابراین فرایند بینایی انسان تقریباً پیوسته اتفاق می‌افتد. برعکس، دنباله‌های ویدیویی در زمان پیوسته نیستند و از تعداد قاب‌های محدودی در واحد زمان استفاده می‌کنند. همچنین ممکن است زاویه مشاهده موضوع در یک زمان بسیار کوتاه تغییر کند. با چشم‌پوشی از برخی محدودیت‌ها، می‌توان بر اساس سیستم بینایی انسان، الگوریتمی پیشنهاد کرد که با به‌دست آوردن میزان سرعت حرکت بلوک‌ها نسبت به پیکسل بر سطح

¹ motion estimation /motion compensation

² motion vector

³ Motion Estimation and Motion Compensation Driven Neural Network

⁴ Three-Step Search

⁵ new Three Step Search

⁶ four Step Search

⁷ Diamond Search

⁸ Hexagonal Search

⁹ Test Zone Search

¹⁰ Modified Diamond Search

¹¹ Adaptive Rood Pattern Search

¹² Modified Test Zone Search

¹³ Hexagonal Grid Search

¹⁴ Dynamic Motion Estimation and Evolution

(قاب)، تخمینی از موقعیت بلوک در قاب‌های آینده به دست آورد و بدین ترتیب تعداد نقاط جستجوی کمتری را برای به دست آوردن بهترین تطبیق (کم‌ترین خطا) انجام دهد.

الگوریتم پیشنهادی در چهار مرحله انجام می‌شود و در سه مرحله می‌تواند بهترین تطبیق را انجام دهد که عبارت است از ۱- محاسبه بردار حرکت ۲- ارزیابی صحت تطبیق انجام یافته ۳- تصمیم برای قاب‌های بعدی. اگر به هر دلیلی بهترین تطبیق صورت نگیرد مرحله چهارم اجرا می‌شود که شامل جستجوی نقاط ارزیابی نشده است. با اجرای این الگوریتم مشخص شد که روش پیشنهادی قادر است نتایج قابل قبولی را در بهبود سرعت تخمین حرکت و کیفیت تصاویر نهایی ارائه کند. در ادامه این تحقیق به ترتیب به بررسی الگوریتم TSS، معرفی روش پیشنهادی، نتایج حاصل شده از شبیه‌سازی الگوریتم پیشنهادی خواهیم پرداخت و در آخر عملکرد و چشم انداز توسعه آینده این الگوریتم را بیان خواهیم کرد.

۲- الگوریتم TSS

الگوریتم TSS یکی از روش‌های رایج در تخمین حرکت ویدیو است که برای کاهش زمان محاسباتی و افزایش کارایی در تحلیل ویدیوها طراحی شده است. این الگوریتم به‌ویژه در کدگذاری ویدیوها استفاده می‌شود و می‌تواند به‌طور مؤثری دقت تخمین حرکت را حفظ کند.

الگوریتم TSS شامل سه مرحله اصلی است. در مرحله اول، یک ماکروبلوک (یا بلوک) در قاب مرجع بررسی می‌شود. در این مرحله، الگوریتم یک جستجوی اولیه در یک ناحیه کوچک اطراف موقعیت پیش‌بینی شده بلوک انجام می‌دهد. این ناحیه معمولاً به‌صورت یک مربع یا مستطیل تعریف می‌شود و الگوریتم از یک نقطه مرکزی شروع کرده و نقاط اطراف آن را بررسی می‌کند تا بهترین انطباق را پیدا کند.

در مرحله دوم، پس از پیدا کردن بهترین موقعیت در مرحله اول، الگوریتم به ناحیه برنده شده متمرکز می‌شود و جستجو را با دقت بیشتری انجام می‌دهد. این مرحله شامل بررسی نقاط بیشتری در اطراف نقطه بهینه پیدا شده در مرحله اول است تا دقت تخمین حرکت افزایش یابد.

مرحله سوم به جستجو در ناحیه‌ای کوچک‌تر از ناحیه قبلی اختصاص دارد. در این مرحله، الگوریتم دوباره جستجو را انجام می‌دهد تا بهترین انطباق ممکن را پیدا کند و شامل بررسی نقاط مجاور و نزدیک به موقعیت بهینه است که در مراحل قبلی پیدا شده است.

از مزایای الگوریتم TSS می‌توان به کاهش زمان محاسبات و حفظ دقت تخمین حرکت اشاره کرد. با این حال، این الگوریتم حساسیت به حرکات سریع را دارد و عملکرد آن ممکن است به تنظیمات و پارامترهای خاصی بستگی داشته باشد که باید به‌دقت انتخاب شوند. به‌طور کلی، الگوریتم TSS به‌عنوان یک روش مؤثر برای تخمین حرکت در ویدیوها شناخته می‌شود و به خاطر سادگی و کارایی آن در بسیاری از سیستم‌های کدگذاری ویدیو مورد استفاده قرار می‌گیرد.

۳- الگوریتم پیشنهادی DVBFME¹

با توجه به حجم بالای اطلاعات قاب‌های متوالی در یک توالی ویدیویی امکان ارسال آن بدون تخمین حرکت تقریباً امکان‌پذیر نیست. الگوریتمی بر اساس ویژگی‌های سیستم بینایی انسان پیشنهاد می‌شود که جابجایی بلوک‌ها را به‌صورت هوشمند پیشگویی می‌کند و بردار حرکت دقیق‌تری را برای جبران حرکت تولید می‌کند. این روش باعث می‌شود پیچیدگی محاسبات کاهش یابد و سرعت محاسبات به زمان آنی نزدیک شود. در این تکنیک، با استفاده از تعداد نقاط جستجوی کمتر و کاهش نرخ بیت، درعین حال که موقعیت بلوک در قاب بعدی به‌درستی تخمین زده می‌شود، سرعت محاسبات تخمین حرکت افزایش یافته و PSNR بهبود می‌یابد. استاندارد فشرده‌سازی ویدیو به‌خاطر استفاده از نرخ بیت پایین، در حرکات سریع نیازمند پردازش زیادی است؛ به همین دلیل، حرکات سریع تأخیر زمانی زیادی را به سامانه اعمال می‌کند. الگوریتم پیشنهادی می‌تواند با کاهش زمان فشرده‌سازی و کاهش اطلاعات اضافی در فرایند تخمین حرکت، کیفیت نهایی ویدیو را بهبود بخشد.

¹ Direction and velocity-based Fractional motion estimation

برای به کارگیری الگوریتم پیشنهادی در روند تخمین حرکت پیشرفته، مختصات بلوک‌های تطبیق یافته نسبت به مختصات قاب مرجع و مقدار بردار حرکت محاسبه می‌شود. با استفاده از اطلاعات به دست آمده از بردارهای حرکت، سرعت و زاویه حرکت تخمین زده می‌شود. با به دست آوردن سرعت و جهت حرکت هر ماکروبلوک، در قاب بعدی، به میزان حرکت تخمین زده شده از موقعیت بلوک، موقعیت غیردقیقی از نقاط جستجوی منطقه‌ای بلوک مورد نظر پیش‌بینی می‌شود. در ناحیه‌ی به دست آمده مجموع تفاوت‌های مطلق (SAD¹) برای تخمین حرکت محک می‌خورد. اگر SAD قابل قبولی به دست آید، ادامه جستجو برای کسب حداکثر تطبیق بهتر به صورت 1/2 پیکسل و نهایتاً به صورت 1/4 پیکسل ادامه پیدا می‌کند تا بهترین تطبیق ممکن حاصل شود.

تخمین حرکت مبتنی بر الگوریتم پیشنهادی سعی دارد تا ماکرو بلوک (MB) جاری را با جستجوی بهترین ماکرو بلوک کاندید که کمترین مقدار SAD را ارائه می‌دهد، پیش‌بینی کند. مقدار SAD برای بلوک‌های با اندازه M×N در هر موقعیت (m,n) به صورت رابطه ۱ محاسبه می‌شود:

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |C(i, j) - R(m+i, n+j)| \quad (1)$$

در اینجا، C(i, j) یک بلوک از قاب جاری است و R(m+i, n+j) بلوک کاندید از پنجره جستجو (SW) قاب قبلاً کدگذاری شده است. مقدار (m, n) که نمایانگر موقعیت نسبی بین MB جاری و بهترین MB کاندید مطابقت یافته در SW است، به عنوان بردار حرکت (MV) شناخته می‌شود [۱۸].

روش پیشنهادی در سه مرحله می‌تواند بهترین تطبیق را انجام دهد تا بردار حرکت به دست آید که بلوک دیاگرام الگوریتم پیشنهادی در شکل ۱ قرار داده شده است.

مرحله ۱- اولین قاب هر دنباله‌ی ویدیویی به عنوان قاب مرجع به صورت یک تصویر واحد مانند JPEG. به روش درون قابی رمز می‌شود و در بلوک دیاگرام شکل ۱ در قسمت 1 Refrence frame نشان داده شده است.

مرحله ۲- برای محاسبه اختلاف دومین قاب هر دنباله‌ی ویدیویی با قاب مرجع رمز شده و به دست آوردن بردار حرکت، بلوک‌های قاب دوم نسبت به قاب مرجع به وسیله الگوریتم 1/4 پیکسل سه مرحله‌ای جستجو می‌شود. در بررسی مرحله دوم دو نتیجه حاصل می‌شود و در بلوک محاسبه بردار حرکت انجام می‌شود.

الف - ماکروبلوک از نظر مکانی نسبت به ماکروبلوک مرجع ثابت است.

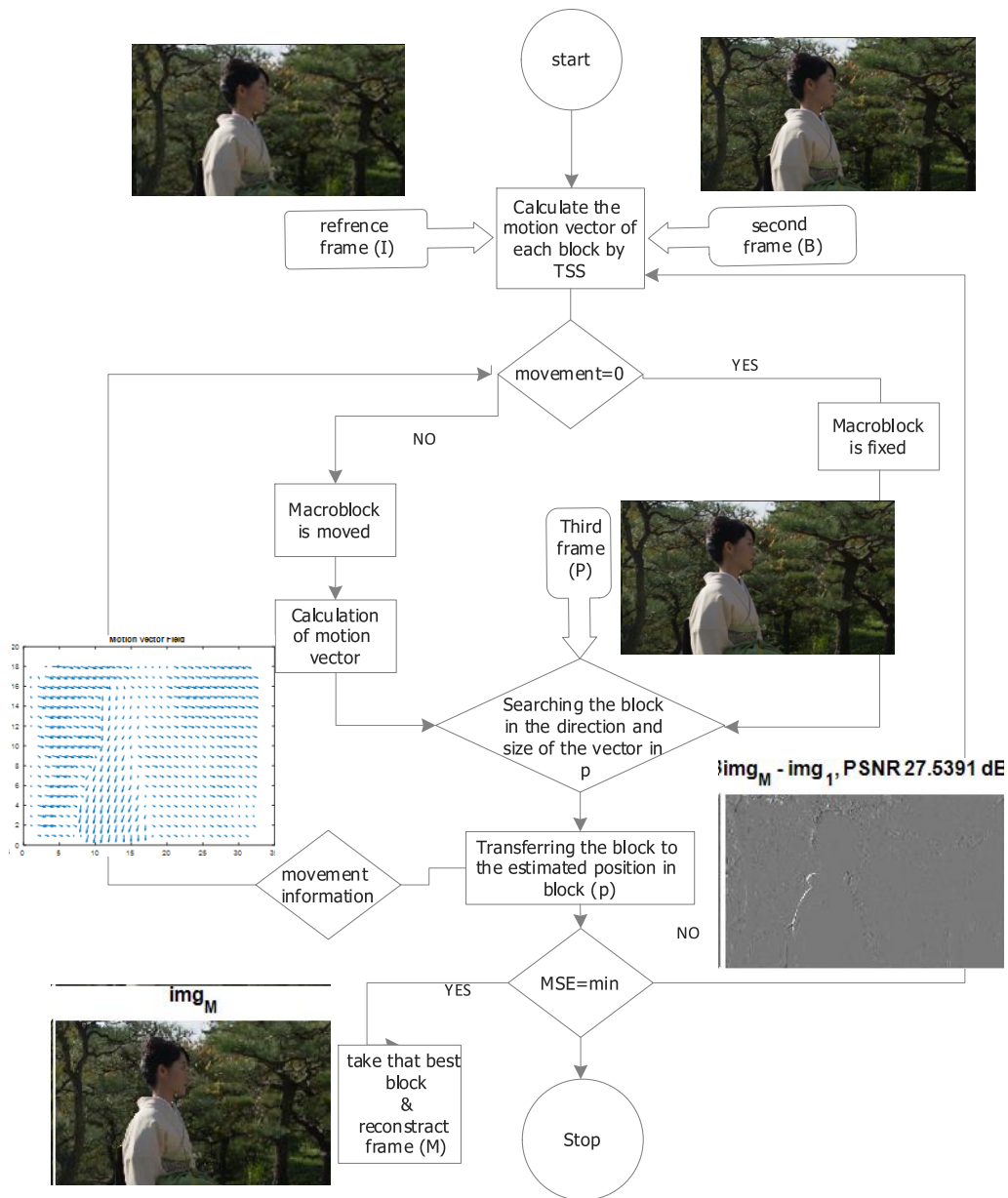
ب - ماکروبلوک از نظر مکانی نسبت به ماکروبلوک مرجع جابه‌جا شده است.

نتایج الف و ب در بلوک جابجایی ارزیابی می‌شود.

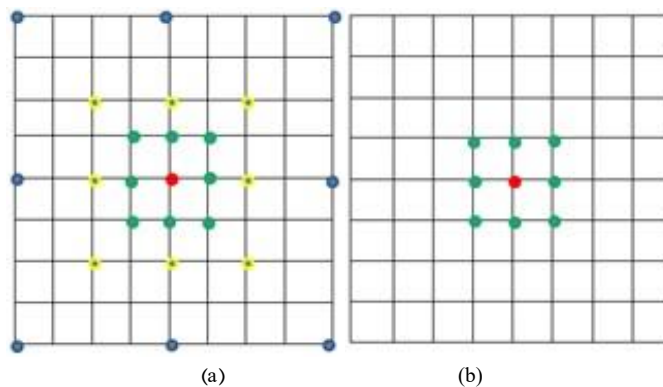
اگر در قاب دوم تطبیق بلوک به گونه‌ای انجام گیرد که مانند شکل ۲ بهترین تطبیق در مرکز ماکروبلوک اتفاق افتد، در این حالت هیچ اطلاعات حرکتی برای بلوک مورد نظر به دست نمی‌آید. در این حالت ماکروبلوک از منظر الگوریتم DVBFME، ثابت است و هیچ حرکتی را انجام نداده است (بلوک ثابت است) تصمیم‌گیری می‌شود. لذا در قاب بعدی برای جستجوی ماکروبلوک مورد نظر، اولین نقطه نامزد برای بررسی بهترین تطبیق، نقطه مرکز بلوک است. در این قاب برای تطبیق بهتر ابتدا ماکروبلوک فعلی در موقعیت ماکروبلوک قبلی بررسی می‌شود، اگر نتیجه قابل قبولی حاصل شد برای تطبیق بهتر این ماکروبلوک، به اندازه یک پیکسل همسایگی این منطقه را بررسی می‌کند. از بین این دو بررسی نتیجه بهتر را انتخاب می‌کند. ولی برای اطمینان بیشتر و جلوگیری از اوجاج احتمالی، همسایگی نقطه مورد نظر تا دقت ربع پیکسل دوباره جستجو می‌شود.

شکل ۲- a اگر با جستجو تا مرحله ربع پیکسل، بهترین تطبیق در مرکز ماکروبلوک باشد، نشان می‌دهد هنوز این ماکروبلوک ثابت است و هیچ تغییر مکانی در این ماکروبلوک رخ نداده است. در قاب بعدی برای همین ماکروبلوک مرحله دوم تکرار می‌شود. در غیر این صورت اگر کوچک‌ترین حرکتی حتی به اندازه ربع پیکسل اتفاق افتاده باشد (تطبیق خارج از مرکز ماکروبلوک به دست آید) دوباره اختلاف نقطه مرکزی ماکروبلوک با قاب قبلی محاسبه می‌شود در قسمت (بلوک متحرک است) ارزیابی می‌شود. برای تطبیق بهتر مرحله سوم اجرا می‌شود.

¹ Sum of Absolute Differences



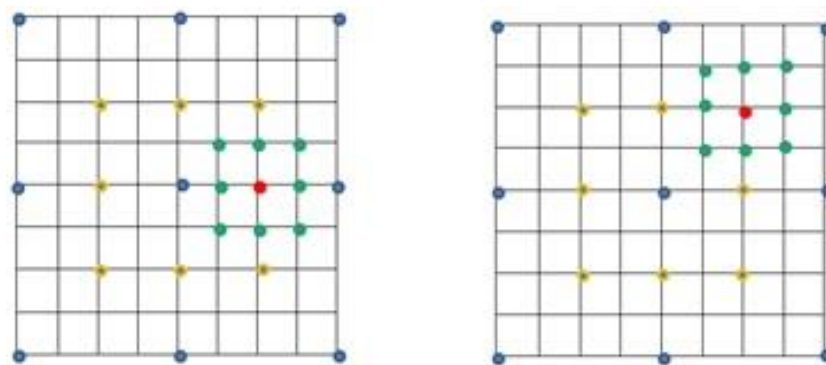
شکل ۱: بلوک دیاگرام الگوریتم پیشنهادی
Figure 1. Block diagram of the proposed algorithm



شکل ۲: (a) جستجوی 1/4 پیکسل کامل و تطبیق در مرکز ماکروبلوک. (b) جستجوی بهینه
Figure 2. (a) 1/4 full pixel search and matching in the center of the macroblock. (b) Optimal search

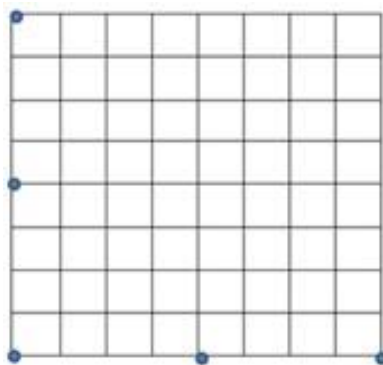
هدف از جستجوی با دقت $1/4$ در همسایگی نقطه مورد نظر این است که اگر حرکت ناچیزی توسط ماکروبلوک مورد بررسی اتفاق افتاده باشد، با بررسی و به دست آوردن بردار حرکت (بدون بررسی $1/4$ پیکسل) نمی توان محل واقعی ماکروبلوک را تشخیص داد. لذا با توقف الگوریتم جستجو و اعلام بردار حرکت، احتمال اعوجاج وجود دارد. با به دست آوردن پارامترهای تطبیق بلوک، نقطه‌ی با مشخصات تطبیق بهتر برای ترسیم بردار حرکت نهایی انتخاب می شود.

مرحله ۳- اگر بهترین تطبیق خارج از مرکز بلوک مورد آزمایش اتفاق بیفتد شکل ۳، الگوریتم پیشنهادی، بلوک را متحرک تشخیص می دهد، یعنی موقعیت ماکروبلوک نسبت به قاب مرجع تغییر یافته است و این الگوریتم سعی در تعیین موقعیت دقیق این بلوک در قاب آینده را دارد. به همین خاطر این الگوریتم مقدار جابه جایی بهترین نقطه تطبیق یافته نسبت به قاب قبلی را بر حسب پیکسل بر قاب به دست آورده جهت حرکت نیز محاسبه می کند.



شکل ۳. روند جستجوی $1/4$ پیکسل کامل و تطبیق در خارج از مرکز ماکروبلوک
Figure 3. $1/4$ full pixel search process and macroblock off-center matching

با به دست آوردن مقدار و جهت حرکت ماکروبلوک، برای بررسی قاب بعدی، نقطه برنده شده در قاب قبلی به مرکز پنجره جستجو منتقل می شود. برای دستیابی به نقطه‌ی درست، الگوریتم پیشنهادی با استفاده از اطلاعات حرکتی قاب قبلی شیوه جستجو را طوری تغییر می دهد تا بلوک پیش بینی شده را بدون جستجو در سایر نقاط، در نقطه‌ی مورد نظر امتحان کند و بهترین تطبیق را به دست آورد. این عمل از طریق احتمال وقوع فعلی بر اساس قاب قبلی و احتمال وقوع بعدی بر اساس قاب فعلی شکل می گیرد. با جمع اطلاعات کاملی از حرکت ماکروبلوک، برای جستجو در قاب‌های بعدی تصمیم گیری می شود.



شکل ۴: جستجوی $1/4$ پیکسل بدون استفاده از IME
Figure 4. $1/4$ pixel search without using IME

مرحله ۴- اگر در مرحله سوم با جستجوی $1/2$ پیکسل SAD قابل قبولی حاصل نشود به این معنی است که در قاب مورد بررسی الگوی حرکتی تغییر یافته است و یا الگوریتم در تشخیص تقریبی موقعیت بلوک، محاسبه درستی را انجام نداده است. اگر این عمل با هر عنوان ممکن اتفاق افتد، الگوریتم پیشنهادی برای جلوگیری از محاسبات اضافی، جستجو با دقت $1/2$ را متوقف کرده سپس مرحله دوم تکرار می شود (شکل ۴).

سعی بر این است که اطلاعات حرکتی هر بلوک مستقل از بلوک دیگر باشد. ولی در بعضی از دنباله‌های ویدیویی حرکت بلوک‌ها یکنواخت به نظر می‌رسند، مانند ویدیو کنفرانس‌ها که بیشتر پشت زمینه ثابت و سر و گردن گوینده متحرک است. چون اجزای سر به هم متصل است، لذا حرکت آن‌ها به غیر از چرخش سر، به یک‌سو، با یک سرعت انجام می‌گیرد. از این روش می‌توان در رمزگذاری ویدیو با نرخ بیت پایین بهره برد. یعنی اگر تغییرات مجموعه‌ای از ماکروبلوک‌ها نسبت به هم با یک سرعت و در یک جهت باشد می‌توان به جای ارسال تک تک اطلاعات ماکروبلوک‌ها اطلاعات حرکتی مجموعه ماکروبلوک‌ها را ارسال کرد. اگرچه ممکن است ظاهراً این روش پیچیدگی محاسبات را بالا ببرد ولی در سیستم‌های پخش عمومی سعی شده است پیچیدگی‌های بالا متوجه رمزکننده‌ها باشد نه رمزگشاها تا با کاهش پیچیدگی در گیرنده‌ها هزینه آن‌ها کاهش یابد. الگوریتم پیشنهادی این خواسته را با اعمال پیچیدگی‌ها در طرف فرستنده ارضا می‌کند. این عمل در مقابل کاهش نرخ بیت و افزایش PSNR عمل معقولی است.

تخمین حرکت و به دست آوردن بردار حرکت به طور قابل توجهی به کاهش نرخ بیت در فشرده‌سازی ویدئو کمک می‌کند، زیرا این فرآیندها به ما اجازه می‌دهند، به جای ذخیره‌سازی کامل هر قاب، تنها تغییرات بین قاب‌های متوالی را ثبت کنیم. با شناسایی جابه‌جایی‌ها در تصویر، می‌توانیم فقط اطلاعات مربوط به حرکت (بردارهای حرکت) را ذخیره کنیم و داده‌های تکراری را حذف نماییم. این رویکرد به ما این امکان را می‌دهد که از تکنیک‌های فشرده‌سازی مؤثرتری استفاده کنیم و حجم داده‌ها را کاهش دهیم، در حالی که کیفیت تصویر را در نرخ بیت ثابت حفظ می‌کنیم. در نتیجه، تخمین حرکت به ما کمک می‌کند با بهینه‌سازی داده‌ها و حذف اطلاعات غیرضروری، انتقال و ذخیره‌سازی ویدئو را کارآمدتر کنیم. البته مقدار کاهش نرخ بیت به مقدار اندازه بلوک‌های انتخابی وابسته است و انتخاب درست آن در کیفیت نهایی تصویر بازسازی شده مشهود است. در این تحقیق با شبیه‌سازی الگوریتم پیشنهادی روی توالی ویدیویی FHD 1080p مشخص شد که نرخ فشرده‌سازی آن بسیار مؤثر بوده است و مقدار کاهش نرخ بیت به صورت رابطه زیر محاسبه می‌شود.

Frame size for img1: 49766400.00 bits

Motion vector size: 18432.00 bits

$$\text{Bit rate reduction percentage} = \left(\frac{\text{Bitrate for frame}}{\text{Bitrate of MotionVector}} \right) \times 100 \quad (2)$$

مشخص شد در این بررسی میزان ۲۷۰۰۰۰ درصد کاهش نرخ بیت برای هر قاب اتفاق افتاده است.

۴- نتایج شبیه‌سازی

الگوریتم پیشنهادی بر روی یک ایستگاه کاری که سیستم عامل Windows 10 Pro را اجرا می‌کند، پیاده‌سازی شد. این ایستگاه کاری دارای پردازنده Intel(R) Core (TM) i7-3632QM با فرکانس ۲/۲۰ GHz و ۸/۰۰ GB RAM بوده همچنین واحد GPU آن Intel (R) HD Graphics 4000 است و از MATLAB 2022a برای پردازش توالی‌های ویدیویی استاندارد مختلف با حداکثر اندازه بلوک ۱۶×۱۶ و دقت ۱/۴ پیکسل استفاده شد. این شبیه‌سازی روی توالی‌های Coastguard, Caltrain, Kimono, trevor, Missa و Surfside که یک قاب بازسازی شده آن‌ها در شکل ۷a نشان داده شده است. جبران حرکت انجام یافته بر اساس مقدار بردار حرکت به دست آمده در شکل ۷b و بردار حرکت قاب مربوطه که با الگوریتم پیشنهادی محاسبه و ترسیم شده در شکل ۷c ارائه شده است. از نظر PSNR بین ۰/۲ db تا ۱/۵ db افزایش کیفیت حاصل شد. ولی نقطه قوت این الگوریتم کاهش محاسبات تخمین حرکت است که به طور متوسط در بعضی از توالی‌ها تا ۳۵ برابر محاسبات لازم کاهش یافته است و نتایج به دست آمده در جدول ۱ ارایه شده است. مقدار میانگین (کیفیت و زمان محاسبات) به صورت جداگانه برای هر توالی ویدیویی در جدول ۱ قرار داده شده است.

زمان محاسبات این الگوریتم بر حسب زمان برای هر قاب در شکل ۶ نشان می‌دهد. رنگ آبی مربوط به مقدار محاسبات بر حسب ثانیه برای الگوریتم پیشنهادی و رنگ قرمز مربوط به الگوریتم جستجوی TSS است.

در پردازش تصویر و ویدیو، PSNR یک معیار رایج برای ارزیابی کیفیت یک تصویر فشرده یا بازسازی شده است. این معیار نسبت بین حداکثر توان ممکن سیگنال و توان نویزی که بر نمایش آن تأثیر می‌گذارد را محاسبه می‌کند. در واقع، PSNR سطح نویز

یا اعوجاج در یک تصویر را پس از فشرده‌سازی با بازسازی نسبت به تصویر اصلی اندازه‌گیری می‌کند. مقدار بالای PSNR نشان‌دهنده کیفیت بالاتر تصویر است. معادله PSNR به صورت رابطه ۳ است:

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (3)$$

که در آن MAX نمایانگر حداکثر مقدار پیکسل تصویر و MSE به معنای خطای میانگین مربعات بین تصویر اصلی و تصویر فشرده شده است و به صورت رابطه ۴ بیان می‌شود [۱۹].

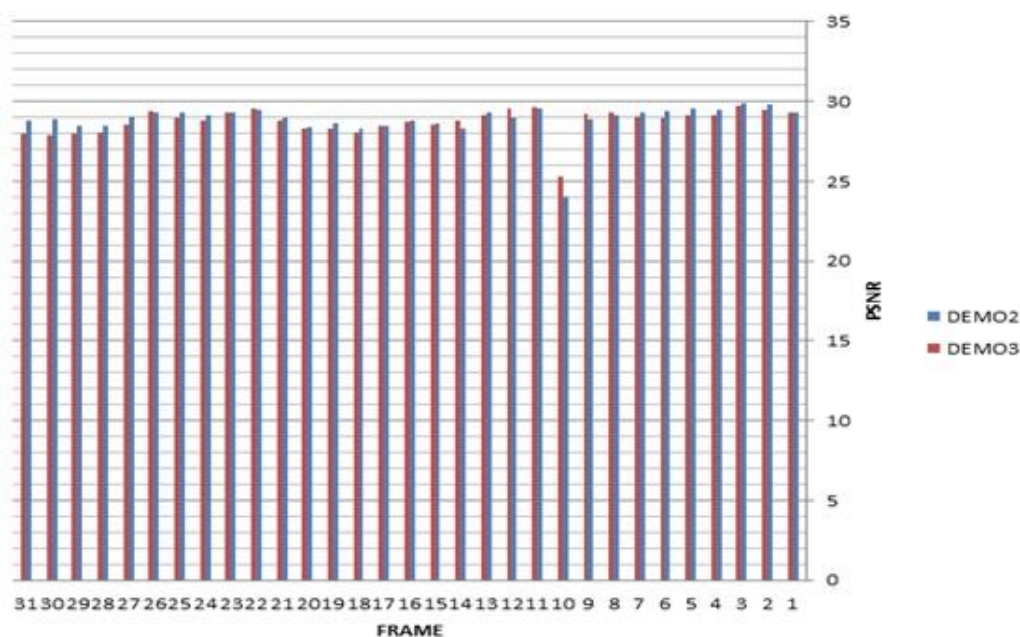
$$MSE = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \|R(x, y) - C(x, y)\|^2 \quad (4)$$

در رابطه ۳، m و n به‌عنوان تصاویر (قاب‌های فردی ویدیو) تعریف می‌شوند، که در آن بعد R(x, y) نمایانگر قاب مرجع و (x, y) C(x, y) نمایانگر قاب جبران حرکت شده است [۱۹].

نتیجه کیفی شبیه‌سازی (PSNR) این الگوریتم با رنگ آبی و الگوریتم TSS به رنگ قرمز در شکل ۵ نشان داده شده است. در اولین قاب به خاطر استفاده از یک الگوریتم یکسان مقدار PSNR یکسان است ولی بقیه قاب‌ها به جز مقدار کمی از قاب‌ها افزایش کیفیت را نشان می‌دهد.

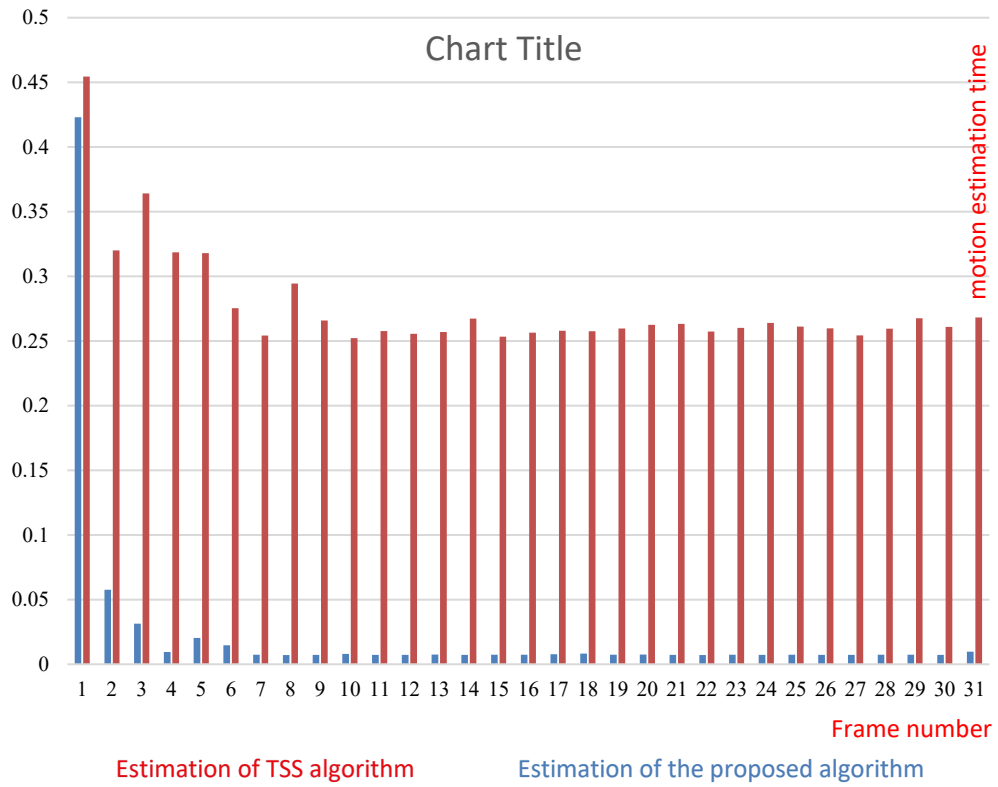
این الگوریتم محاسبات تخمین حرکت مبتنی بر بلوک را کاهش می‌دهد و اگر تطبیق بهتری در یک بلوک انجام نگیرد فقط همان بلوک با الگوریتم جایگزین بهتر، بهترین تطبیق را انجام خواهد داد.

۳۰ قاب از توالی استاندارد Coastguard با الگوریتم پیشنهادی شبیه‌سازی و با الگوریتم جستجوی سه‌مرحله‌ای مقایسه شد، و نتیجه کیفی آن در شکل ۵ به صورت نمودار نمایش داده شد.

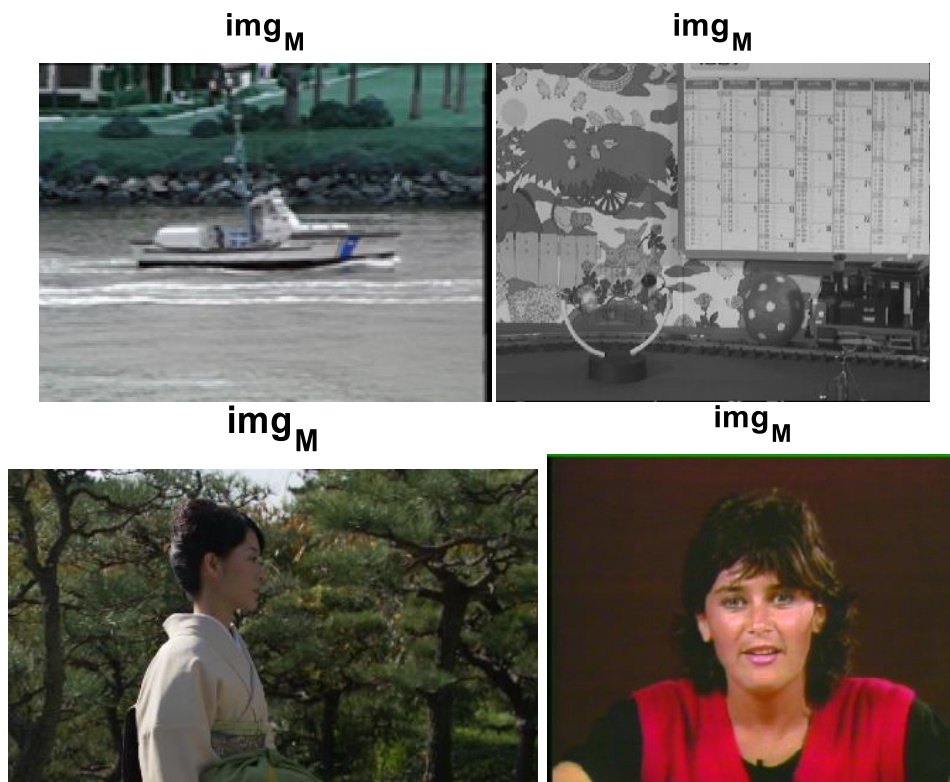


شکل ۵: مقایسه الگوریتم DVBFME و جستجوی TSS از لحاظ PSNR

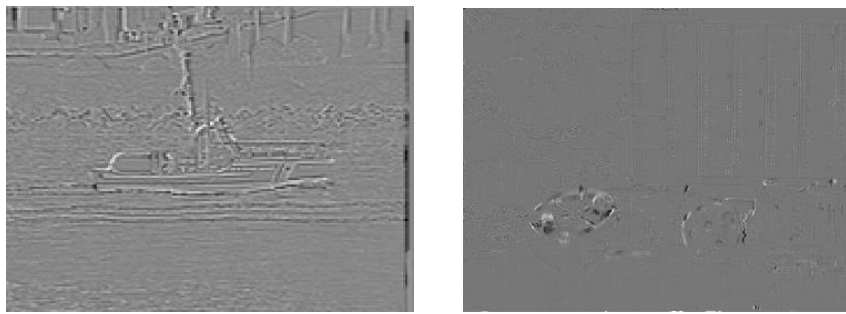
Figure 5. Comparison of the DVBFME algorithm and the TSS search in terms of PSNR



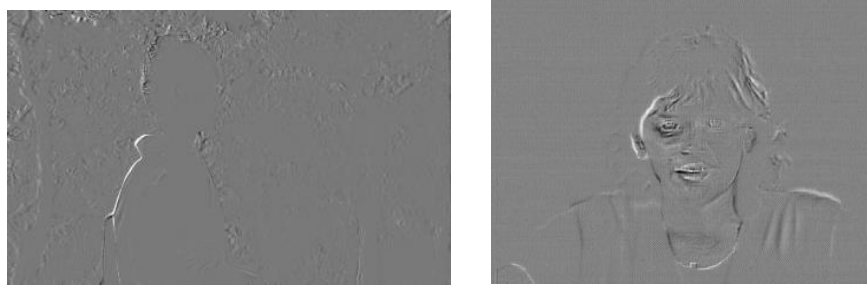
شکل ۶: مقایسه الگوریتم DVBFME و جستجوی TSS از لحاظ زمان پردازش
 Figure 6. Comparison of the DVBFME algorithm and the TSS search in terms of processing time



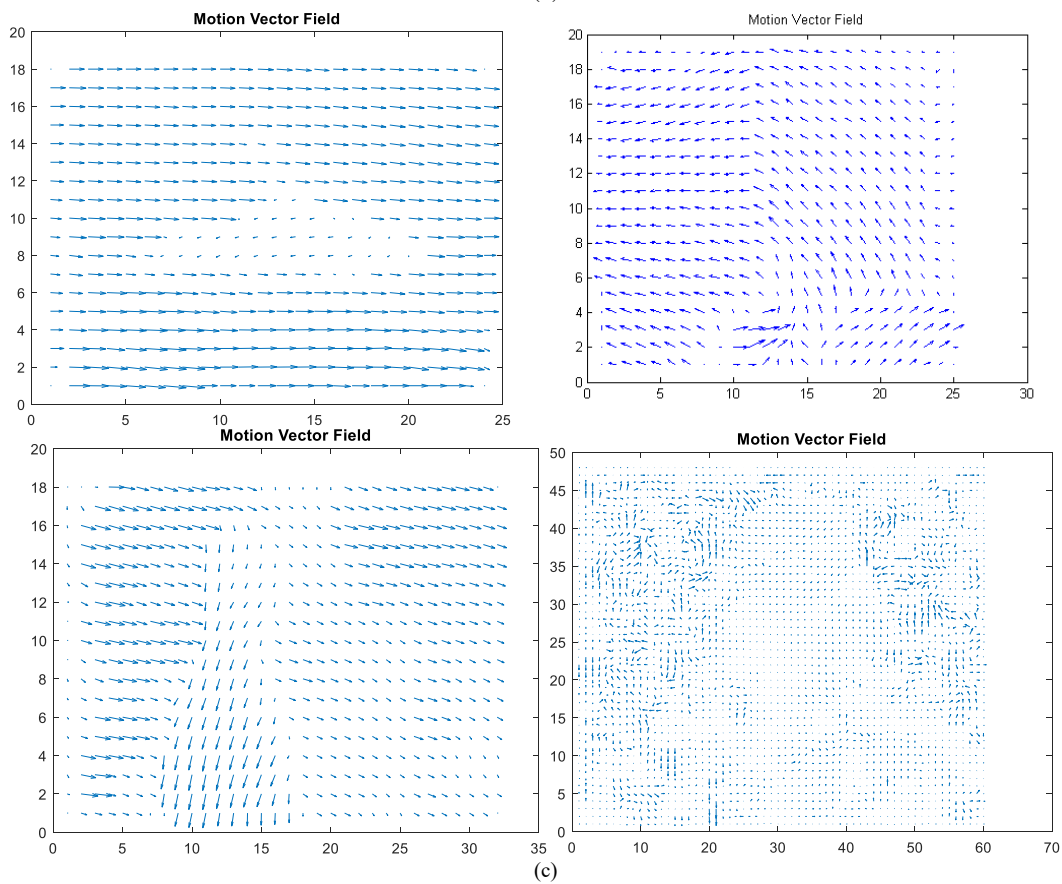
$img_M - img_2$, PSNR 29.4959 dB $img_M - img_1$, PSNR 30.3682 dB



$img_M - img_1$, PSNR 27.5391 dB $img_M - img_2$, PSNR 29.535 dB



(b)



(c)

شکل ۷: (a) قاب بازسازی شده، (b) بردارهای حرکت و (c) تصاویر جریان حرکت شده
Figure 7. (a) Reconstructed frame, (b) motion compensation images, and (c) motion vectors

همچنین برای بررسی دقیق‌تر، الگوریتم پیشنهادی با تعدادی از الگوریتم‌های ارایه شده در جدول ۲ مقایسه شد. با مشاهده نتایج به‌دست‌آمده و ارزیابی آن با سایر روش‌ها اختلاف معناداری را با آنها نشان می‌دهد.

جدول ۱: میانگین زمان تخمین و جبران حرکت هر توالی و پارامترهای کیفی

Table 1. Average estimation time and movement compensation of each sequence and qualitative parameters

| Motion estimation time (second) | Motion compensation time (second) | SSIM | MSE | PSNR | Algorithm | Sequence |
|---------------------------------|-----------------------------------|-------|-----------|--------|-----------|------------|
| 0.048000 | 0.0282222 | 0.977 | 0.000086 | 39.720 | DVBFME | Trevor |
| 0.4011935 | 0.0251258 | 0.932 | 0.00038 | 33.191 | TSS | |
| 0.0370593 | 0.05862187 | 0.946 | 0.00049 | 28.936 | DVBFME | Caltrain |
| 1.2880645 | 0.05770925 | 0.902 | 0.0011 | 29.539 | TSS | |
| 0.0167042 | 0.01873125 | 0.964 | 0.000615 | 31.892 | DVBFME | Coastguard |
| 0.2711806 | 0.02314193 | 0.920 | 0.00163 | 26.980 | TSS | |
| 0.0376906 | 0.05690312 | 0.994 | 0.000893 | 39.674 | DVBFME | Missia |
| 0.4939402 | 0.05845161 | 0.942 | 0.000828 | 32.765 | TSS | |
| 0.3695011 | 0.50926801 | 0.992 | 0.0000797 | 48.869 | DVBFME | Surfside |
| 3.150000 | 0.54545025 | 0.882 | 0.000786 | 30.700 | TSS | |

جدول ۲: متوسط SSIM و PSNR قاب‌های بازسازی شده

Table 2. Average SSIM and PSNR of reconstructed frames

| Algorithms | Sequences | | | | |
|------------|----------------|-----------------|------------|--------|-------|
| | BlowingBuBbles | BasketballDrill | Racehorses | kimono | |
| PSNR (dB) | DVBFME | 36.82 | 38.5 | 34.94 | 40.1 |
| | TSS | 19.9 | 25.5 | 18.6 | 37.1 |
| | VTM[20] | 34.03 | 35.4 | 34.7 | 35.8 |
| | HM[21] | 31.05 | 33.6 | 33.5 | 35.3 |
| | RLVC[22] | 30.69 | 30.8 | 31.2 | 34.2 |
| | M-LVC[23] | 30.81 | 30.7 | 30.8 | 34.6 |
| | DVC-PRO[24] | 30.89 | 31.1 | 31.01 | 34.9 |
| | DCVC[25] | 31.7 | 31.3 | 31.7 | 34.4 |
| | CANF-VC[26] | 33.1 | 32.7 | 32.8 | 35.01 |
| | TCMVC[27] | 33.6 | 32.9 | 33.5 | 35.2 |
| | HEM[28] | 34.25 | 33.55 | 34.3 | 35.82 |
| | [29] | 34.65 | 34.2 | 35.1 | 35.9 |
| | SSIM | DVBFME | 0.994 | 0.989 | 0.993 |
| TSS | | 0.718 | 0.749 | 0.95 | 0.861 |
| VTM | | 0.990 | 0.986 | 0.988 | 0.986 |
| HM | | 0.989 | 0.983 | 0.985 | 0.983 |
| RLVC | | 0.991 | 0.982 | 0.987 | 0.982 |
| M-LVC | | 0.987 | 0.977 | 0.98 | 0.977 |
| DVC-PRO | | 0.989 | 0.982 | 0.986 | 0.982 |
| DCVC | | 0.991 | 0.986 | 0.988 | 0.986 |
| CANF-VC | | 0.988 | 0.984 | 0.985 | 0.984 |
| TCMVC | | 0.992 | 0.987 | 0.989 | 0.987 |
| HEM | | 0.992 | 0.988 | 0.991 | 0.988 |
| [29] | | 0.993 | 0.989 | 0.991 | 0.989 |

الگوریتم پیشنهادی با تخمین دقیق مقدار جابجایی بلوکی انجام شده در قاب‌های متوالی، زمان تخمین حرکت را به مقدار قابل مقایسه‌ای کاهش می‌دهد. برای اثبات ادعای مطرح شده، الگوریتم پیشنهادی به یک توالی Full HD 1080p اعمال شده است و نتایج متوسط زمان تخمین حرکت انجام یافته را برای هر قاب از این توالی را در جدول ۳ ارائه کرده است

جدول ۳: متوسط زمان تخمین حرکت برای هر قاب 1080p

Table 3. Average motion estimation time for each 1080p frame

| Algorithms | Motion Estimation |
|------------|-------------------|
| DVBFME | 0.216 s |
| TSS | 8.42 s |
| TCMVC[27] | 0.81 s |
| HEM[28] | 0.75 s |
| [29] | 0.94 s |

۵- نتیجه‌گیری

ویدیوهای با رزولیشن بالا و تعداد قاب‌های بیشتر در ارتباطات ویدیویی نیاز به محاسبات بیشتری دارد این امر تأخیر قابل توجهی را بر سیستم کد کننده اعمال می‌کند، همچنین در بحث ویدیو کنفرانس‌ها که ارتباط به صورت زمان آنی از اهمیت بالایی برخوردار است ارزش تخمین حرکت سریع را دو چندان می‌کند. برای حل این مورد الگوریتم پیشنهادی مطرح شد و برای ارزیابی آن روی توالی ویدیویی استاندارد مختلف امتحان شد. نتایج نشان داد الگوریتم پیشنهادی زمان محاسبات را به طور چشمگیر کاهش داد و درعین حال باعث بهبود نسبی کیفیت ویدیو شد. لذا می‌توان با رعایت شرایط الگوریتم در کد کننده‌های ویدیویی در قسمت تخمین حرکت از آن بهره برد و با کاهش پیچیدگی محاسبات به تخمین حرکت آنی نزدیک شد. البته می‌توان از الگوی شتاب حرکتی بلوک‌ها در مقیاس پنجره جستجو برای تسریع این الگوریتم استفاده کرد ولی به خاطر لزوم تحقیقات زیاد برای به کارگیری موفق آن نتایج به دست آمده در این پژوهش مورد استفاده قرار نگرفته است.

مراجع

- [1] Y. Zhang, Ch. Zhang, R. Fan, S. Ma, Zh. Chen, and C. C. J. Kuo, "Recent advances on HEVC inter-frame coding: From optimization to implementation and beyond," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4321-4339, November 2020, doi: [10.1109/TCSVT.2019.2954474](https://doi.org/10.1109/TCSVT.2019.2954474).
- [2] S. Gogoi and R. Peesapati, "Design and implementation of an efficient multi-pattern motion estimation search algorithm for HEVC/H.265," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 319-328, November 2021, doi: [10.1109/TCE.2021.3126670](https://doi.org/10.1109/TCE.2021.3126670).
- [3] W. Bao, W. Sh. Lai, X. Zhang, and Zh. Gao, "MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 1-16, March 2021, doi: [10.48550/arXiv.1810.08768](https://doi.org/10.48550/arXiv.1810.08768).
- [4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *National Telecommunication Conference*, New Orleans, LA, 1981, pp. 531-535.
- [5] R. Li, B. Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-442, August 1994, doi: [10.1109/76.313138](https://doi.org/10.1109/76.313138).
- [6] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313-317, June 1996, doi: [10.1109/76.499840](https://doi.org/10.1109/76.499840).
- [7] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 369-377, August 1998, doi: [10.1109/76.709403](https://doi.org/10.1109/76.709403).
- [8] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349-355, May 2002, doi : [10.1109/TCSVT.2002.1003474](https://doi.org/10.1109/TCSVT.2002.1003474).
- [9] K. C. R. C. Varma and S. Mahapatra, "Complexity reduction of test zonal search for fast motion estimation in uni-prediction of high efficiency video coding," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 511-524, June 2021, doi: [10.1007/s11554-020-00983-y](https://doi.org/10.1007/s11554-020-00983-y).
- [10] O. Ndili and T. Ogunfunmi, "Fast algorithm and efficient architecture for integer and fractional motion estimation," *Journal of Signal Processing Systems*, vol. 75, no. 1, pp. 55-64, June 2014, doi: [10.1007/s11265-013-0793-8](https://doi.org/10.1007/s11265-013-0793-8).

- [11] Y. Nie and K. K. Ma, "Adaptive rood pattern search for fast blockmatching motion estimation," *IEEE Transaction on Image Processing*, vol. 11, no. 12, pp. 1442-1449, December 2002, doi: [10.1109/TIP.2002.806251](https://doi.org/10.1109/TIP.2002.806251).
- [12] N. Purnachand, L. N. Alves, and A. Navarro, "Fast Motion Estimation Algorithm for HEVC," *IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, Germany, 2012, pp. 34-37, doi: [10.1109/ICCE-Berlin.2012.6336494](https://doi.org/10.1109/ICCE-Berlin.2012.6336494).
- [13] K. Singh and S. R. Ahamed, "Computationally efficient motion estimation algorithm for HEVC," *Journal of Signal Processing Systems*, vol. 90, no. 12, pp. 1713-1727, December 2018, doi: [10.1007/s11265-017-1321-z](https://doi.org/10.1007/s11265-017-1321-z).
- [14] N. Kim and J. W. Kang, "Dynamic motion estimation and evolution video prediction network," *IEEE Transactions on Multimedia*, Vol. 23, PP. 3986-3998, November 2020, doi: [10.1109/TMM.2020.3035281](https://doi.org/10.1109/TMM.2020.3035281).
- [15] [S. Gogoi](https://orcid.org/0000-0001-9550-4411) and R. Peesapati, "A hybrid motion estimation search algorithm for HEVC/H.265," *IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, India, Rourkela, 2019, pp. 129-132, doi: [10.1109/iSES47678.2019.00037](https://doi.org/10.1109/iSES47678.2019.00037).
- [16] Y. S. Ho, "Advanced video coding for next-generation multimedia services," Published by InTech Janeza Trdine 9, Croatia. 2012, pp. 1-214.
- [17] E. P. Simoncelli and D. J. Heeger, "A model of neuronal responses in visual area MT," *Vision Research*, vol. 38, no. 5, pp. 743-761, March 1998, doi: [10.1016/S0042-6989\(97\)00183-1](https://doi.org/10.1016/S0042-6989(97)00183-1).
- [18] S. K. Chatterjee, S. K. Vittapu, and S. Kundu, "Prediction-biased diamond search algorithm: a new approach to reduce motion estimation complexity," *Microsystem Technologies*, vol. 27, pp. 2027-2032, January 2021, doi: [10.1007/s00542-020-05167-z](https://doi.org/10.1007/s00542-020-05167-z).
- [19] Sh. Agha, M. Khan, and F. Jan, "Efficient fast motion estimation algorithm for real-time applications," *Journal of Real-Time Image Processing*, vol. 19, pp. 403-413, January 2022, doi: [10.1007/s11554-021-01188-7](https://doi.org/10.1007/s11554-021-01188-7).
- [20] B. Bross, J. Chen, J. -R. Ohm, G. J. Sullivan, and Y. -K. Wang, "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC)," in *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463-1493, Sept. 2021, doi: [10.1109/JPROC.2020.3043399](https://doi.org/10.1109/JPROC.2020.3043399).
- [21] G. J. Sullivan, J. -R. Ohm, W. -J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: [10.1109/TCSVT.2012.2221191](https://doi.org/10.1109/TCSVT.2012.2221191).
- [22] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, "Learning for Video Compression With Recurrent Auto-Encoder and Recurrent Probability Model," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 388-401, Feb. 2021, doi: [10.1109/JSTSP.2020.3043590](https://doi.org/10.1109/JSTSP.2020.3043590).
- [23] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: Multiple Frames Prediction for Learned Video Compression," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 3543-3551, doi: [10.1109/CVPR42600.2020.00360](https://doi.org/10.1109/CVPR42600.2020.00360).
- [24] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An End-to-End Learning Framework for Video Compression," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3292-3308, 1 Oct. 2021, doi: [10.1109/TPAMI.2020.2988453](https://doi.org/10.1109/TPAMI.2020.2988453).
- [25] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 18114-18125.
- [26] Y.-H. Ho, C.-P. Chang, P.-Y. Chen, A. Gnutti, and W.-H. Peng, "CANFVC: Conditional augmented normalizing flows for video compression," in *Proc. 17th Eur. Conf. Comput. Vis. Tel Aviv*, Israel: Springer, 2022, pp. 207-223.

- [27] X. Sheng, J. Li, B. Li, L. Li, D. Liu, and Y. Lu, "Temporal Context Mining for Learned Video Compression," in *IEEE Transactions on Multimedia*, vol. 25, pp. 7311-7322, 2023, doi: [10.1109/TMM.2022.3220421](https://doi.org/10.1109/TMM.2022.3220421).
- [28] J. Li, B. Li and Y. Lu, "Hybrid spatial-temporal entropy modelling for neural video compression," in *Proc. 30th ACM Int. Conf. Multimedia*, pp. 1503-1511, 2022, doi: [10.1145/3503161.3547845](https://doi.org/10.1145/3503161.3547845).
- [29] X. Sheng, L. Li, D. Liu, and H. Li, "Spatial Decomposition and Temporal Fusion Based Inter Prediction for Learned Video Compression," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 7, pp. 6460-6473, July 2024, doi: [10.1109/TCSVT.2024.3360248](https://doi.org/10.1109/TCSVT.2024.3360248).