remote sensing

MDPI

Article

Lossy and Lossless Video Frame Compression: A Novel Approach for High-Temporal Video Data Analytics

Zayneb Ahmed [1], Abir Jaafar Hussain [2], Wasiq Khan [2], Thar Baker [2], Haya Al-Askar [3,*], Janet Lunn [2], Raghad Al-Shabandar [4], Dhiya Al-Jumeily [2] and Panos Liatsis [5]

[1] Department of Mathematics, University of Baghdad, Baghdad, Iraq; Zainabaa_math@csw.uobaghdad.edu.iq

[2] Computer Science Department, Liverpool John Moores University, Byrom Street, Liverpool L33AF, UK; a.hussain@ljmu.ac.uk (A.J.H.); w.khan@ljmu.ac.uk (W.K.); t.baker@ljmu.ac.uk (T.B.); J.lunn@ljmu.ac.uk (J.L.); d.aljumeily@ljmu.ac.uk (D.A.-J.)

[3] Computer Science Department, College of Engineering and Computer Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

[4] Artificial Intelligence Team, Datactics, Belfast BT1 3LG, UK; raghad.al-shabandar@datactics.com

[5] Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, UAE; panos.liatsis@ku.ac.ae

* Correspondence: h.alaskar@psau.edu.sa

Received: 5 February 2020; Accepted: 11 March 2020; Published: 20 March 2020

Abstract: The smart city concept has attracted high research attention in recent years within diverse application domains, such as crime suspect identification, border security, transportation, aerospace, and so on. Specific focus has been on increased automation using data driven approaches, while leveraging remote sensing and real-time streaming of heterogenous data from various resources, including unmanned aerial vehicles, surveillance cameras, and low-earth-orbit satellites. One of the core challenges in exploitation of such high temporal data streams, specifically videos, is the trade-off between the quality of video streaming and limited transmission bandwidth. An optimal compromise is needed between video quality and subsequently, recognition and understanding and efficient processing of large amounts of video data. This research proposes a novel unified approach to lossy and lossless video frame compression, which is beneficial for the autonomous processing and enhanced representation of high-resolution video data in various domains. The proposed fast block matching motion estimation technique, namely mean predictive block matching, is based on the principle that general motion in any video frame is usually coherent. This coherent nature of the video frames dictates a high probability of a macroblock having the same direction of motion as the macroblocks surrounding it. The technique employs the partial distortion elimination algorithm to condense the exploration time, where partial summation of the matching distortion between the current macroblock and its contender ones will be used, when the matching distortion surpasses the current lowest error. Experimental results demonstrate the superiority of the proposed approach over state-of-the-art techniques, including the four step search, three step search, diamond search, and new three step search.

Keywords: remote sensing; IOT; smart city; block-matching algorithm; macroblocks; video compression; motion estimation

1. Introduction

Over 60% of the world's population lives in urban areas, which indicates the exigency of smart city developments across the globe, to overcome planning, social, and sustainability challenges [1,2] in urban areas. With recent advancements in the Internet of Things (loT), cyber-physical domains, and cutting-edge wireless communication technologies, diverse applications have been enabled within smart cities. Approximately 50 billion devices will be connected through the internet by 2020 [3], which gives an indication of visual IoT significance as a need for future smart cities across the globe. Visual sensors embedded within the devices have been used for various applications, including mobile surveillance [4], health monitoring [5], and security applications [6]. Likewise, world leading automotive industrial stakeholders anticipate the rise of driverless vehicles [7] based on adaptive technologies in the context of dynamic environments. All of these innovations and future advancements involve high temporal streaming of heterogeneous data (and in particular, video), which require the use of automated methods in terms of efficient storage, representation, and processing.

Many countries across the globe (Australia for instance) have been rapidly focusing on the expansion of closed-circuit television networks to monitor incidents and anti-social behavior in public places. Moreover, smart technology produces improved outcomes, e.g., in China, where authorities utilize facial recognition in automated teller machines in order to verify account owners in an attempt to crack down on money laundering in Macau [8]. Since the initial installation of these ATMs in 2017, the Monetary Authority of Macao confirmed that the success rate of stopping illegitimate cash withdrawals has reached 95%, which is a major success. Another application of video surveillance and facial recognition cameras was introduced in Australian casinos to catch cheaters and thieves [9]. Sydney's Star Casino invested AUD $10 million in a system that uses this technology to match faces with known criminals stored in a database, in an attempt to deter foul play. Such systems may be very useful in relevant application domains, such as delivering a better standard of service to regular customers.

Despite increases in demand and the significance of intelligent video-based monitoring and decision-making systems, there is a number of challenges which need to be addressed in terms of the reliability of data-driven and machine intelligence technologies [10]. These interconnected devices generate very high temporal resolution datasets, including large amounts of video data, which need to be processed by efficient video processing and data analytics techniques to produce automated, reliable, and robust outcomes. A special case, for instance, recently reported by the British Broadcasting Corporation (BBC), is existing face matching tools, deployed by the UK police, which are reported as staggeringly inaccurate [11]. Big Brother Watch, a campaign group, investigated the technology, and reported that it produces extremely high numbers of false positives, thus identifying innocent people as suspects. Likewise, the 'INDEPENDENT' newspaper [12] and 'WIRED'magazine [13] reported 98% of the Metropolitan and South Wales Police facial recognition technology misidentify suspects. These statistics indicate a major gap in existing technology, which needs to be investigated to deal with challenges associated with the autonomous processing of high temporal video data for intelligent decision making in smart city applications.

Recent developments enabled visual IoT devices to be utilized in unmanned aerial (UAVs) and ground vehicles (UGVs) [14,15] in remote sensing applications, i.e., capturing visual data in specific context, which is typically not possible for humans. In contrast to satellites, UAVs have the ability to capture heterogeneous high-resolution data, while flying at low altitudes. Low altitude UAVs, which are well-known for remote sensing data transmission in low bandwidth, include DJI Phantom, Parrot, and DJI Mavic Pro [16]. Moreover, some of the world leading industries, such as Amazon and Alibaba, utilize UAVs for the optimal delivery of orders. Recently, [17] proposed UAVs as a useful delivery platform in logistics. The size of the market for UAVs in remote sensing as a core technology, will grow to more than US $11.2 billion by 2020 [2]. High stream remote sensing data, including UAVs, needs to be transmitted to the target data storage/processing destination in real time, using limited transmission bandwidth, which is a major challenge [18]. Reliable and time efficient video data compression will significantly contribute towards minimizing the time-space resource needs, while reducing redundant information, specifically from high temporal resolution remote sensing data, and maintaining the quality of the representation, at the same time. This will ultimately improve the accuracy and reliability of machine intelligence based video data representation models, and subsequently impact on automated monitoring, surveillance, and decision making applications.

Video Data Compression: A challenge for the aforementioned smart city innovations and data driven technologies is to achieve acceptable video quality through the compression of digital video images retrieved through remote sensing devices and other video sources, whilst still maintaining a good level of video quality. There are many video compression algorithms proposed in the literature, including those targeting mobile devices, e.g., smart phones and tablets [19,20]. Current compression techniques aim at minimizing information redundancy in video sequences [21,22]. An example of such techniques is motion compensated (MC) predictive coding. This coding approach reduces temporal redundancy in video sequences, which, on average, is responsible for 50%–80% of the video encoding intricacy [23–26]. Prevailing international video coding standards, for example H.26x and the Moving Picture Experts Group (MPEG) series [27–33], use motion compensated predictive coding. In the MC technique, current frames are locally modeled. In this case, MC utilizes reference frames to estimate the current frame, and afterwards finds the residuals between the actual and predicted frames. This is known as the residual prediction error (RPE) [23,24,34–36].

Fan et al., [10] indicated that there are three main approaches for the reduction of external bandwidth for video encoding, including motion estimation, data reuse, and frame recompression. Motion estimation (ME) is the approximation of the motion of moving objects, which needs to be estimated prior to performing motion compensation predictive coding [23,24,37,38]. It should be emphasized that an essential part of MC predictive coding is the block matching algorithm (BMA). In this approach, video frames are split into a number of non-overlapping macroblocks (MBs). In the current frame, the target MB is searched against a number of possible macroblocks in a predefined search window in the reference frame so as to locate the best matching MB. Displacement vectors are determined as the spatial differences between the two matching MBs, which determine the movement of the MB from one location to another in the reference frame [39,40]. The difference between two macroblocks can be expressed through various block distortion measures (BDMs), e.g., the sum of absolute differences (SAD), the mean square error (MSE), and the mean absolute difference (MAD) [41]. For a maximum displacement of n pixels/frame, there are $(2n + 1)^2$ locations to be considered for the best match of the current MB. This method is called full search (FS) algorithm, and results in the best match with the highest spatial resolution. The main issue with the use of FS is excessive computational times, which led to the development of computationally efficient search approaches.

A number of fast block matching algorithms are proposed in the literature to address the limitations of the FS algorithm [42]. Work in this field began as early as the 1980s, and the rapid progress in the field resulted in some of these algorithms being implemented in a number of video coding standards [3,27–29]. Variable block sizes and numerous reference frames are implemented in the latest video coding standards, which resulted in considerable computational requirements. Therefore, motion estimation has become problematic for several video applications. For example, in mobile video applications, video coding for real time transmission is important. This further reinforces that this is an immensely active area of research. It should be noted that the majority of fast block matching algorithms are practically implemented to fixed block search ME and then applied to variable block search ME [43]. Typically, the performance of fast BMA is compared against the FS algorithm, by measuring reductions in the residual prediction error and computational times.

As with other video and image compression techniques, fast block matching algorithms can be categorized into lossless and lossy. Lossy BMAs can achieve higher compression ratios and faster implementation times than FS by sacrificing the quality of the compressed video, while lossless BMAs have the specific requirement of preserving video quality at the expense of typically lower compression ratios.

The main contributions of this research are as follows:

1. A novel technique based on the mean predictive block value is proposed to manage computational complexity in lossless and lossy BMAs.

2. It is shown that the proposed method offers high resolution predicted frames for low, medium, and high motion activity videos.
3. For lossless prediction, the proposed algorithm speeds up the search process and efficiently reduces the computational complexity. In this case, the performance of the proposed technique is evaluated using the mean value of two motion vectors for the above and left previous neighboring macroblocks to determine the new search window. Therefore, there is a high probability that the new search window will contain the global minimum, using the partial distortion elimination (PDE) algorithm.
4. For lossy block matching, previous spatially neighboring macroblocks are utilized to determine the initial search pattern step size. Seven positions are examined in the first step and five positions later. To speed up the search process, the PDE algorithm is applied.

The reminder of this paper is organized as follows. Section 2 provides an overview of fast block matching algorithms. The proposed framework, termed mean predictive block matching (MPBM) algorithms for lossless and lossy compression, is illustrated in Section 3. In Section 6, the simulation results of the proposed methods are presented. The conclusions of this work and avenues for further research are described in Section 5.

2. Fast Block Matching Algorithms

A variety of fast block matching algorithms have been developed with the aim of improving upon the computational complexity of FS. Some of these algorithms are utilized in video coding standards [23–29].

There exist a variety of lossy and lossless BM algorithms. Lossy BMAs can be classified into fixed set of search patterns, predictive search, subsampled pixels on matching error computation, hierarchical or multiresolution search, and bit-width reduction. Lossless BMAs include the successive elimination algorithm (SEA) and the partial distortion elimination (PDE) algorithm. A brief overview and recent developments in fast BM algorithms are presented in the following sub-sections.

2.1. Fixed Set of Search Patterns

The most widely used category in lossy block matching algorithms is the fixed set of search patterns or reduction in search positions. These algorithms reduce the search time and complexity by selecting a reduced set of candidate search patterns, instead of searching all possible MBs. By and large, they work on the premise that error reduces monotonically when the search location passes closer to the best-match position. Thus, such algorithms start their search according to a certain pre-defined uniform pattern at locations coarsely ranged near the search window. Then, the search is reiterated around the location of the minimum error with a smaller step size. Each search pattern has a specific shape (i.e., rectangular, hexagonal, cross, diamond, etc.) [24,44,45]. Examples of well-known fixed sets of search pattern algorithms include three step search (TSS) [46], new three step search (NTSS) [47], four step search (4SS) [48], diamond search (DS) [49], and simple and efficient search (SESTSS) [50].

This group of methods has received a great deal of attention due to their fast MB detection rates. However, there is a considerable loss in visual quality, particularly when the actual motion does not match the pattern, and thus, these algorithms may get stuck in local minima.

2.2. Predictive Search

Predictive search is a lossy block-matching algorithm that utilizes the connection among neighboring and current MBs, where information in temporal or spatial neighboring MBs, or both, is exploited. The algorithm attains the predicted motion vector (MV) by choosing one of the earlier-coded neighboring MVs, e.g., left, top, top right, their median, or the MV of the encoded MB in the previous frame. The MV predictor provides an initial prediction of the current MV, hence reducing the search points, and associated computations, by anticipating that the target macroblock will likely belong to the area of the neighboring MVs. This method has memory requirements for storage of the neighboring MVs. It has been widely used in the latest video coding standards [4,10,14], and in various algorithms, e.g., adaptive rood pattern search (ARPS) [51], unsymmetrical multi-hexagon search (UMHexagonS) [52], and others [53].

2.3. Partial Distortion Elimination Algorithm

The partial distortion elimination (PDE) algorithm has been widely utilized to reduce computational times and is applied in the context of full search techniques, such as H.263 [54] and H.264 [55]. It uses the halfway-stop method in the block distortion measure computation, by stopping the search when the partial entirety of matching distortion between current and candidate MBs exceeds the current minimum distortion.

To accelerate the process, PDE relies on both fast searching and fast matching. The former depends on how quickly the global minimum is detected in a given search area, which was used in Telenor's H.263 codec [45], while the later speeds up the calculation process by determining the matching error on a candidate MB, by reducing the average number of rows considered per MB. Similarly, Kim et al., [55–57] suggested a variety of matching scans that rely on the association between the spatial intricacy of the reference MB and the block-matching error, which is founded on the concept of representative pixels.

3. Mean Predictive Block Matching Algorithms for Lossless (MPBMLS) and Lossy (MPBMLY) Compression

In this section, the proposed mean predictive block-matching algorithm is introduced. In the lossless block-matching algorithm (MPBMLS), the purpose of the method is to decrease the computational time needed to detect the matching macroblock of the FS, while maintaining the resolution of the predicted frames, similarly to full search. This is performed by using two predictors, i.e., the motion vectors of the two previous neighboring MBs, specifically, the top (MV_A) and the left (MV_L) as shown in Figure 1, and further illustrated in Algorithm 1.

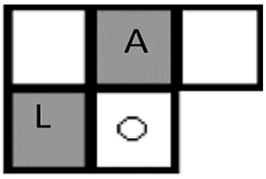

Figure 1. Position of the two predictive macroblocks, where A represents the location of motion vector A (MV_A), while L represents the location of motion vector L (MV_L).

Algorithm 1 Neighboring Macroblocks.

```
1: Let MB represents the current macroblock, while MBSet = {S|S ∈ video frames}
2: ForEach MB in MBSet
3:     Find A and L where
4:         - A & L ∈ MBSet
5:         - A is the top motion vector ($MV_A$)
6:         - L is the left motion vector ($MV_L$)
7:     Compute the SI where
8:         - SI is the size of MB = N × N (number of pixels)
9: End Loop
```

The aim of using two neighboring motion vectors for the prediction is to obtain the global matching MB faster than when using a single previous neighbor. Furthermore, the selection of these predictors will avoid unnecessary computations that derive from choosing three previous neighboring MBs knowing that at each location (i, j), N × N pixels are computed, where N × N is the macroblock size. The neighbors may move in different directions; therefore, these MVs are used to determine the new search window depending on the mean of its components. The search range of the new search window will be the mean of x-components and y-components, respectively, as follows:

$$\begin{aligned} x &= round\left(abs\left(\frac{MV_A(x)+MV_L(x)}{2}\right)\right) \\ y &= round\left(abs\left(\frac{MV_A(y)+MV_L(y)}{2}\right)\right) \end{aligned} \tag{1}$$

We have used the round and the absolute operations to get positive step size values for the search window in both x and y directions. The current MBs are searched in the reference image using first the search range of $\pm x$ in the x-axis and $\pm y$ in the y-axis instead of using the fixed search range of $\pm p$ for both of them as shown in Figure 2 and Algorithm 2.

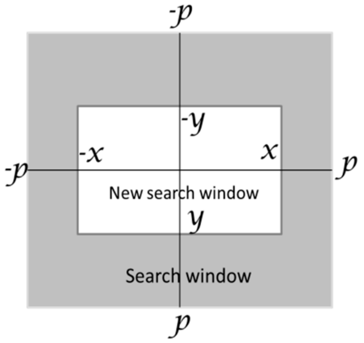

Figure 2. Default search window vs proposed search window.

Algorithm 2 Finding the new search window

```
Let MB represents cthe urrent macroblock in MBSet
ForEach MB in MBSet:
    Find x, y as:
        $x = round\left(abs\left(\frac{MVA(x)+MVL(x)}{2}\right)\right)$
        $y = round\left(abs\left(\frac{MVA(y)+MVL(y)}{2}\right)\right)$
        Such that x, y ≤ $p$, where $p$ is the search window size for the Full Search algorithm
    Find NW representing the set of all points at the corners of the new search window rectangle as:
        $NW = \{(x,\ y),\ (x, -y),\ (-x,\ y),\ (-x, -y)\}$
End Loop
```

Since there is high correlation between neighboring MBs, there is high likelihood that the global matching MB will be inside the new search window. Hence applying the PDE algorithm will speed up the search process. The search will stop when the error between the matching MB calculated from the search window range and the current MB is less than a threshold value, resulting in the remainder of the original search window to not being required to be checked. Otherwise, search will extend to the remainder of the default search window. The threshold is computed as N, i.e., the number of pixels of the MB. Figure 3 shows the block diagram of the proposed lossless MPBMLS algorithm.

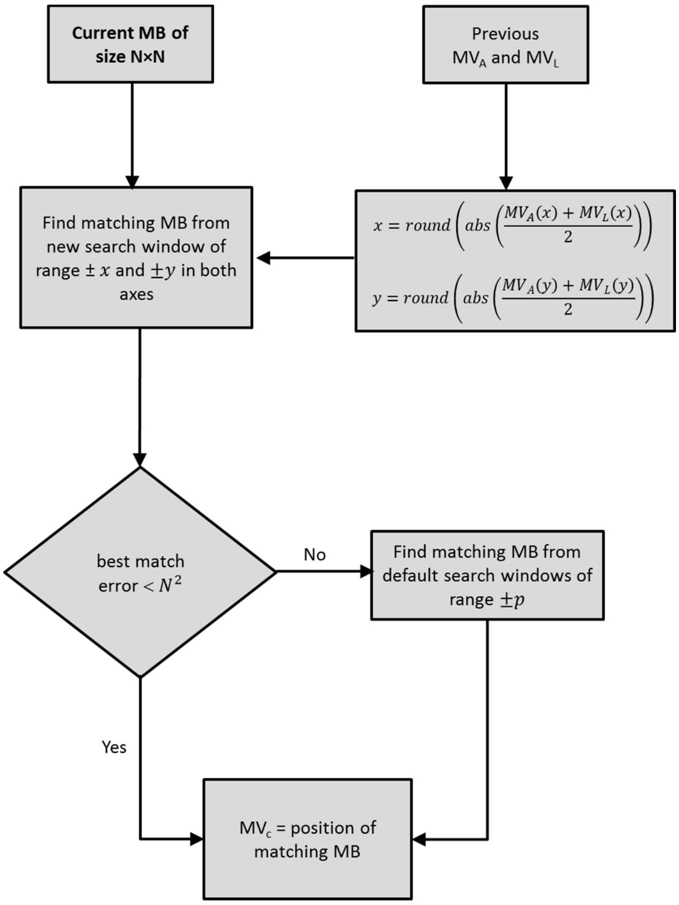
Current MB of size N×N
Previous MV_A and MV_L
Find matching MB from new search window of range $\pm x$ and $\pm y$ in both axes
$x = round\left(abs\left(\frac{MV_A(x) + MV_L(x)}{2}\right)\right)$
$y = round\left(abs\left(\frac{MV_A(y) + MV_L(y)}{2}\right)\right)$
best match error $< N^2$
No
Find matching MB from default search windows of range $\pm p$
Yes
MV_c = position of matching MB

Figure 3. Schematic of the novel mean predictive block matching algorithms for lossless compression (MPBMLS) in calculating the motion vector of the current MB.

For the lossy block matching algorithm (MPBMLY), the proposed method improve the search matching blocks by combining three types of fast block matching algorithms, i.e., predictive search, fixed set of search patterns, and partial distortion elimination.

Predictive search utilizes the motion information of the two previous, left and top, spatial neighboring MBs, as shown in Algorithm 3, in order to form an initial estimate of the current MV. As shown in the MPBMLS algorithm, using these predictors will determine the global matching MB faster than using only one previous neighbor, while reducing unnecessary computations. The maximum of the mean x and y components for the two predictor MVs is used to determine the step size.

The fixed set of search patterns method, as in the adaptive rood pattern searching technique (ARPS) [51], uses two categories of fixed patterns, the small search pattern (SSP) and the large search pattern (LSP), respectively. Moreover, the first step search includes the MVs of the two previous neighboring MBs with the LSP. The step size is used to determine the LSP position in the first step. Therefore, seven positions are examined in this step. To avoid unnecessary computations, this technique uses a pre-selected threshold value to assess the error between the matching and the current macroblocks, which is established from the first step. When the error is less than the threshold, the SSP is not required, resulting in reduced computation requirements.

Algorithm 3 Proposed MPBM technique

```
Step 1
Let $s = \sum SADcenter$ (i.e., midpoint of the current search) and $Th$ a pre-defined threshold value.
IF $s < Th$:
    No motion can be found
    Process is completed.
END IF

Step 2
IF MB is in top left corner:
    Search 5 LPS points
    ELSE:
        - $MV_A$ and $MV_L$ will be added to the search
        - Use $MV_A$ and $MV_L$ for predicting the step size as:

            $L_x = \text{round}\left(\text{abs}\left(\frac{MV_A(x) + MV_L(x)}{2}\right)\right)$

            $L_y = \text{round}\left(\text{abs}\left(\frac{MV_A(y) + MV_L(y)}{2}\right)\right)$

where step size = max{ Lx, Ly}.
        - Matching MB is explored within the LSP search values on the boundary of the step size
{(±step size, 0), (0, ±step size), (0,0)}
    Set vectors {($MV_A$), ($MV_L$)}, as illustrated in Figure 2.
END IFELSE

Step 3
    Matching MB is then explored within the LSP search values on the boundary of the step size
{(±step size, 0), (0, ±step size), (0,0)} and the set vectors {($MV_A$), ($MV_L$)}, as illustrated in Figure 2
The PDE algorithm is used to stop the partial sum matching distortion calculation between the
current macroblock and candidate macroblock as soon as the matching distortion exceeds the
current minimum distortion, resulting in the remaining computations to be avoided, hence,
speeding up the search.

Step 4
Let Er represent the error of the matching MB in step 3.
IF $Er < Th$:
The process is terminated and the matching MB provides the motion vector.
ELSE
- Location of the matching MB in Step 3 is used as the center of the search window
- SSP defined from the four points, i.e., {(±1, 0), (0, ±1)}, will be examined.
End IFELSE
IF matching MB stays in the center of the search window
-Computation is completed
ELSE
- Go to Step 1
- The matching center provides the parameters of the motion vector.
```

Partial distortion elimination is applied to improve computation times. The algorithm will stop the calculation of the partial sum of the current distortion value of the matching distortion between the current macroblock and the candidate macroblock once the matching distortion surpasses the current minimum distortion. Since the initial search depends on two neighboring MBs, the first step search window has a high probability to contain the globally optimal MB and hence computation times should be reduced.

The SADcenter value represents the absolute difference between the current MB and its respective value at the same location in the reference frame. MV_A and MV_L represent the motion vectors of the top and left macroblocks. Algorithm 3 provides an overview of the main steps involved in the proposed mean predictive block matching technique. Figure 4 shows the search pattern for the proposed lossy algorithm, while Figure 5 illustrates the block diagram of the proposed block matching algorithm, MPBMLY.

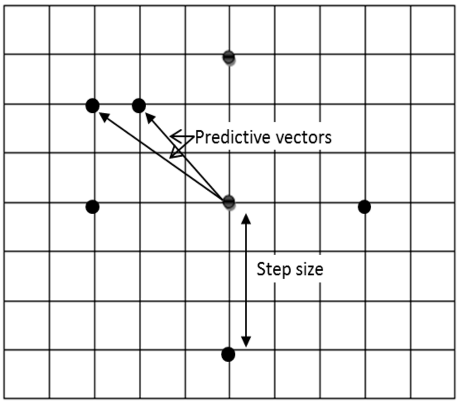

Figure 4. Mean predictive block matching algorithms for the MPBMLY algorithm, including the large search pattern (LSP) and two predictive vectors. Solid circle points (•) correspond to the search points.

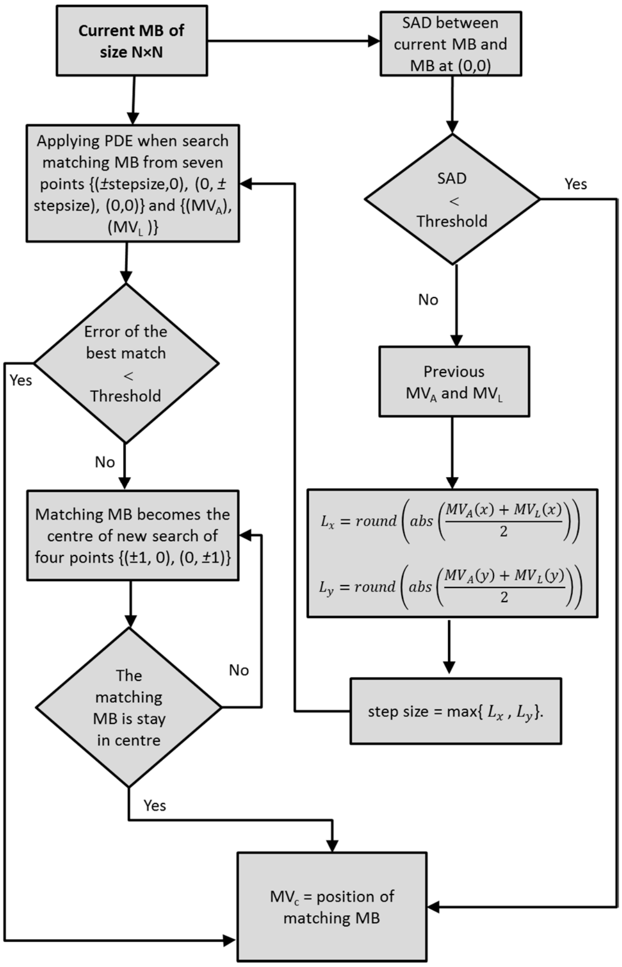
Current MB of size N×N
SAD between current MB and MB at (0,0)
Applying PDE when search matching MB from seven points {(±stepsize,0), (0, ± stepsize), (0,0)} and {(MV$_A$), (MV$_L$)}
SAD < Threshold
Yes
No
Error of the best match < Threshold
Yes
No
Previous MV$_A$ and MV$_L$
$L_x = round\left(abs\left(\frac{MV_A(x) + MV_L(x)}{2}\right)\right)$
$L_y = round\left(abs\left(\frac{MV_A(y) + MV_L(y)}{2}\right)\right)$
Matching MB becomes the centre of new search of four points {(±1, 0), (0, ±1)}
The matching MB is stay in centre
No
Yes
step size = max{ L_x , L_y}.
MV$_c$ = position of matching MB

Figure 5. Block diagram of the proposed MPBMLY algorithm.

4. Simulation Results

The performance of the proposed algorithms is evaluated using the criteria of the matching MB search speed and efficiency in maintaining the residual prediction error between the current frame and its prediction, at the same level as the full search technique. The results are benchmarked with state-of-the-art fast block matching algorithms, including diamond search (DS) [49], new three-step search (NTSS) [47], four step search (4SS) [48], simple and efficient search TSS (SESTSS) [50], and adaptive rood pattern search (ARPS) [51].

The simulations were performed using MATLAB running on Intel I CoreIi3 CPU M330@2.13 GHz processor. The experimental results of the proposed techniques are conducted on the luminance components of 50 frames of six popular video sequences. Three of them are CIF format (common intermediate format) video sequences (i.e., 352 × 288 pixels, 30fps). These are the "News", "Stefan", and "Coastguard" sequences. The remaining three are QCIF format (quarter common intermediate format) video sequences (i.e., 176 × 144 pixels, 30 fps), namely, the "Claire", "Akiyo", "Carphone" sequences.

The selected video sequences have various motion activities. "Akiyo" and "Claire" have low motion activity. "News" and "Carphone" have medium motion activity, while "Coastguard" and "Stefan" have high motion activity. The size of each MB was set to 16 × 16 for all video sequences. To avoid unreasonable results, which may arise due to high correlation between successive frames, the proposed and benchmarking algorithms used the two-steps backward frames as reference frames, which means that if the current frame is I then the reference frame is I-2.

Four performance measures are used to evaluate the performance of the proposed technique. Two of the measures are used to estimate the search speed of the algorithms, i.e., the time required for processing and the average number of search points required to obtain the motion vectors, respectively. The remaining two quality measures are used to evaluate the performance of the proposed algorithms in detecting the predicted frames, i.e., MSE and peak signal-to-noise ratio (PSNR), respectively.

The mean squared error is given by:

$$\mathrm{MSE} = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \left(\mathrm{f}(\mathrm{i},\mathrm{j}) - \hat{\mathrm{f}}(\mathrm{i},\mathrm{j}) \right)^2 \tag{2}$$

where M and N are the horizontal and vertical dimensions of the frame, respectively, and $f(i,j)$ and $\hat{f}(i,j)$ are the pixel values at location (i,j) of the original and estimated frames, respectively.

The peak signal-to-noise ratio is given by:

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{(\mathrm{f}_{\max})^2}{\mathrm{MSE}} \right) \tag{3}$$

where $\mathrm{f}_{\max}$ represents the highest possible pixel value. In our simulations, a value of 255 was used for an image resolution of 8 bits. The MSE and PSNR between the original and the compensated frames are measured by calculating the MSE and PSNR for each frame with their predicted frames, separately, and then calculating their arithmetic means.

A. Lossless Predictive Mean Block Matching Algorithm

Simulations were performed to test the performance of the proposed lossless predictive block matching algorithm. The SAD metric is used as the block distortion measure and determined as follows:

$$SAD = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| C(i,j) - \mathrm{R}\left(i + v_x, j + v_y\right) \right| \tag{4}$$

where $C(i,j)$ is the pixel value of the current MB C of dimension N×N at position (i,j) and $\mathrm{R}\left(i + v_x, j + v_y\right)$ is the pixel value of the reference frame of the candidate macroblock R with a displacement of $\left(v_x, v_y\right)$.

The proposed lossless algorithm is benchmarked with the full search (FS) and the partial distortion elimination (PDE) techniques. The results of the computational complexity were determined using: (1) the average number of search points required to get each motion vector, and (2) the computational times for each algorithm. Table 1 shows the average number of search points, while Table 2 shows the processing times. The resolutions of the predicted frames using the MSE and PSNR are shown in Tables 3 and 4, respectively.

Table 1. Mean value of search points per MB of size 16 × 16. CIF and QCIF correspond to the common intermediate format and quarter common intermediate format, respectively.

Sequence	Format	FS	PDE	MPBMLS
Claire	QCIF	184.56	184.6	48.98
Akiyo	QCIF	184.56	184.6	46.2
Carphone	QCIF	184.56	184.6	170.2
News	CIF	204.28	204.3	121.6
Stefan	CIF	204.28	204.3	204.3
Coastguard	CIF	204.28	204.3	204.3

Table 2. Computation time (in second) required to handle 50 frames.

Sequence	Format	FS	PDE	MPBMLS
Claire	QCIF	0.351	0.18	0.06
Akiyo	QCIF	0.334	0.11	0.01
Carphone	QCIF	0.336	0.18	0.15
News	CIF	1.492	0.65	0.38
Stefan	CIF	1.464	1.09	0.88
Coastguard	CIF	1.485	1.19	1.03

Table 3. Mean MSE with 50 frames.

Sequence	Format	FS	PDE	MPBMLS
Claire	QCIF	9.287	9.287	9.29
Akiyo	QCIF	9.399	9.399	9.399
Carphone	QCIF	56.44	56.44	56.44
News	CIF	30.33	30.33	30.33
Stefan	CIF	556.1	556.1	556.1
Coastguard	CIF	158.3	158.3	158.3

Table 4. Mean PSNR with 50 frames.

Sequence	Format	FS	PDE	MPBMLS
Claire	QCIF	38.94	38.94	38.94
Akiyo	QCIF	39.61	39.61	39.61
Carphone	QCIF	30.82	30.82	30.81
News	CIF	33.48	33.48	33.47
Stefan	CIF	22.16	22.16	22.16
Coastguard	CIF	26.19	26.19	26.19

It should be noted that the experimental results indicate that the proposed lossless technique reduces the search time of macroblock matching, while keeping the resolution of the predicted frames exactly the same as the ones predicted using full search. Furthermore, the performance of the proposed algorithm is more effective when the video sequences have lower motion activity and vice versa. This is due to using the two previous neighbors to predict the dimension of the new search window, which has a high probability to contain the global matching MB and subsequently, ignoring the remaining search points. For high motion activity video sequences, including "Stefan" and "Coastguard", the number of search points in the proposed technique is exactly the same as in FS and PDE with enhancements in processing times, which means that the proposed algorithm uses fast searching to detect the global minimum in the new search window. Figure 6, Figure 7, and Figure 8 show the frame by frame comparison of the average number of search points per MB, PSNR performance, and MSE for 23 frames of "Claire", "Carphone", and "Stefan" sequences, respectively.

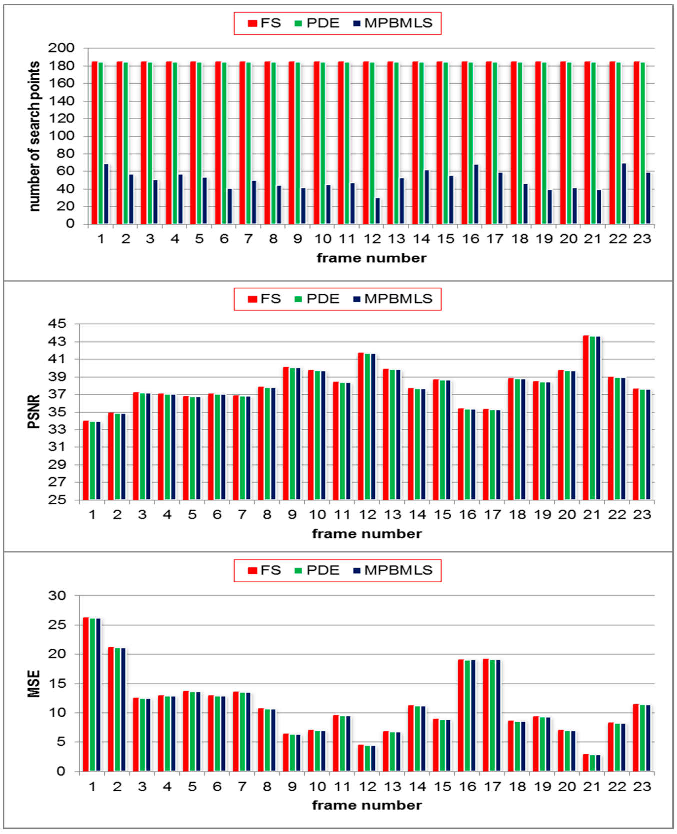

Figure 6. Average number of search points per MB, peak signal-to-noise ratio (PSNR) performance, and mean square error (MSE) of MPBMLS, full search (FS), and partial distortion elimination (PDE) for the "Claire" video sequence of 23 frames.

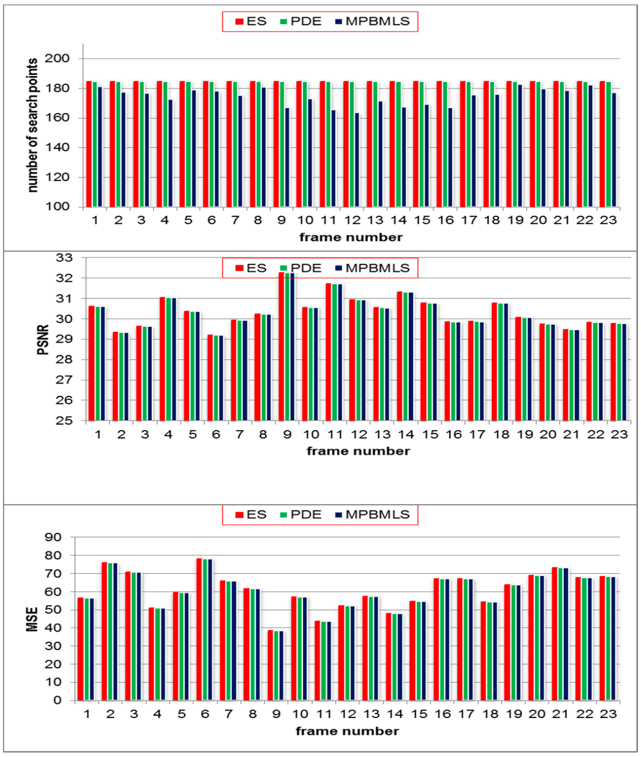

Figure 7. Average number of search points per MB, PSNR performance, and MSE of MPBMLS, FS, and PDE in the "Carphone" video sequence over 23 frames.

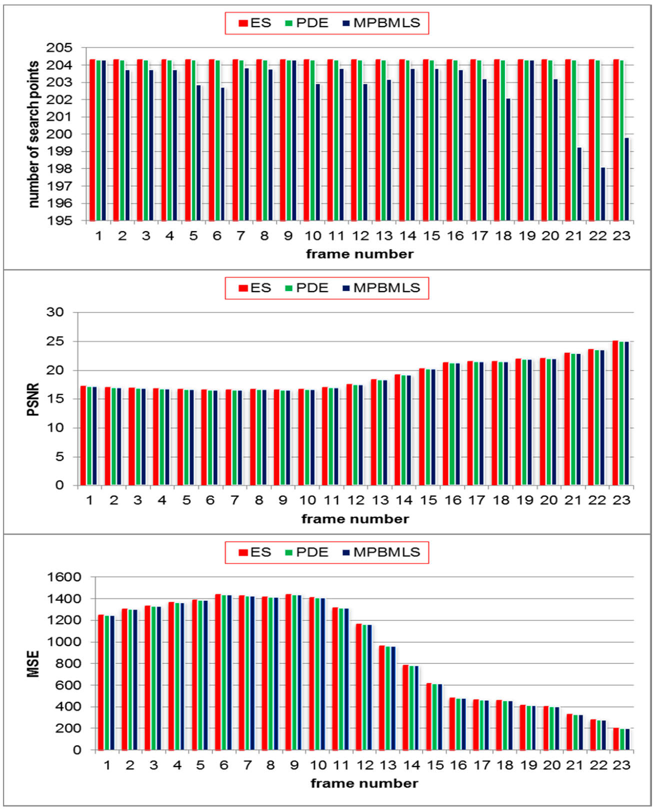

Figure 8. Average number of search points per MB, PSNR performance, and MSE of MPBMLS, FS, and PDE in the "Stefan" video sequence over 23 frames.

B. Lossy Predictive Mean Block Matching Algorithm

For lossy compression, the performance of the proposed technique is evaluated by benchmarking the results with state-of-the-art fast block matching algorithms. The SAD metric (Equation (4)) and the MAD metric (Equation (5)) were used as the block distortion measures. The MAD error metric is defined as as follows:

$$MAD = \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} \left| C(i,j) - R\left(i + v_x, j + v_y\right) \right| \tag{5}$$

where $C(i,j)$ is the pixel value of the current MB C of dimension N×N at position (i,j) and $R\left(i + v_x, j + v_y\right)$ is the pixel value of the reference frame of the candidate macroblock R with a displacement of $\left(v_x, v_y\right)$.

The computational complexity was measured using: (1) the average number of search points required to obtain each motion vector, as shown in Table 5 and, (2) the processing times of each algorithm, as shown in Table 6. The resolutions of the predicted frames using the mean of MSE and mean PSNR are shown in Tables 7 and 8, respectively.

Table 5. Mean number of search points per MB of size 16 × 16. FS, DS, NTSS, 4SS, SESTS, and ARPS correspond to full search, diamond search, new three-step search, four step search, simple and efficient search TSS, and adaptive rood pattern search, respectively.

Sequence	FS	DS	NTSS	4SS	SESTSS	ARPS	MPBMLY
Claire	184.6	11.63	15.09	14.77	16.13	5.191	2.128
Akiyo	184.6	11.46	14.76	14.67	16.2	4.958	1.938
Carphone	184.6	13.76	17.71	16.12	15.73	7.74	7.06
News	204.3	13.1	17.07	16.38	16.92	6.058	3.889
Stefan	204.3	17.69	22.56	19.05	16.11	9.641	9.619
Coastguard	204.3	19.08	27.26	19.91	16.52	9.474	8.952

Table 6. Time needed to process 50 frames (second).

Sequence	FS	DS	NTSS	4SS	SESTSS	ARPS	MPBMLY
Claire	0.351	0.037	0.031	0.031	0.037	0.025	0.015
Akiyo	0.354	0.036	0.031	0.031	0.037	0.023	0.006
Carphone	0.338	0.039	0.036	0.032	0.035	0.031	0.033
News	1.539	0.161	0.142	0.136	0.151	0.112	0.079
Stefan	1.537	0.267	0.232	0.174	0.15	0.158	0.139
Coastguard	1.551	0.263	0.235	0.178	0.15	0.152	0.14

Table 7. Mean MSE for 50 frames.

Sequence	FS	DS	NTSS	4SS	SESTSS	ARPS	MPBMLY
Claire	9.287	9.287	9.287	9.355	9.458	9.289	9.292
Akiyo	9.399	9.399	9.399	9.399	9.408	9.399	9.399
Carphone	56.44	58.16	57.56	62.12	69.62	60.02	59.08
News	27.29	29.41	28.2	29.6	31.22	29.81	28.64
Stefan	556.1	661.4	607.2	651.5	714.5	608	594
Coastguard	158.3	167.4	164.3	166	182.5	164.1	161.6

Table 8. Mean PSNR for 50 frames.

Sequence	FS	DS	NTSS	4SS	SESTSS	ARPS	MPBMLY
Claire	38.94	38.94	38.94	38.92	38.89	38.94	38.94
Akiyo	39.61	39.61	39.61	39.61	39.61	39.61	39.61
Carphone	30.82	30.69	30.7	30.4	30.1	30.58	30.6
News	33.77	33.45	33.63	33.42	33.19	33.39	33.56
Stefan	22.16	21.49	21.81	21.51	21.04	21.82	21.93
Coastguard	26.19	25.98	26.05	26.02	25.6	26.05	26.11

The simulation results indicate that the proposed algorithm (MPBM) outperformed the state-of-the-art methods in terms of computational complexity. Moreover, it is shown that it also preserves or reduces the error between the current and compensated frames. For low motion activity videos, the resolution of the predicted frame is close to the ones predicted by the full search algorithm with enhancements in computational complexity. For the medium and high motion activity videos, improvements in computational complexity and resolution of the predicted frames are reasonable in comparison with the other fast block matching algorithms. Moreover, it should be noted that the ratio between the PSNR and the computational time of the proposed algorithm gives the best results in comparison to the remaining benchmarking algorithms.

Figure 9 gives the frame by frame comparison for the average number of search points per MB, PSNR performance, and MSE of the proposed algorithm in comparison to the state-of-the-art algorithms for 23 frames of the “Carphone”, and “Stefan” video sequences.

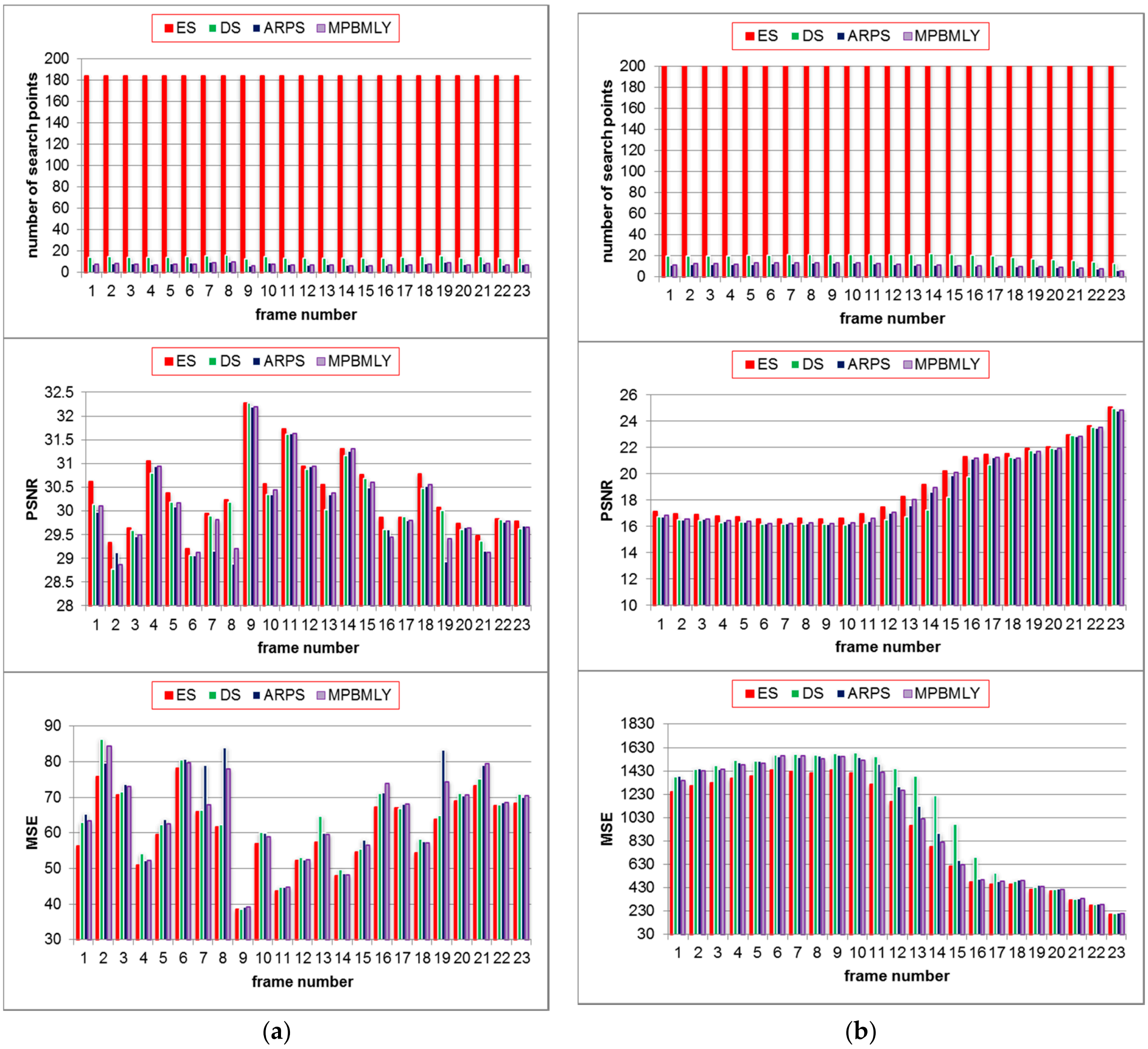

Figure 9. Average number of search points per MB (Top), PSNR performance (Middle), and MSE (Bottom) for MPBMLY and state-of-the-art fast block matching. (**a**) Average for the "Carphone" video sequence over 23 frames; (**b**) average for Stefan" video sequence over 23 frames.

5. Discussion

This manuscript presented a number of research contributions related to fast block matching algorithms in video motion estimation. Specifically, methods were proposed to reduce the computational complexity of lossless and lossy block matching algorithms, which is crucial for the effective processing of remote sensing data [58]. The proposed algorithm will enable faster yet quality transmission of video data, specifically, acquired by UAVs and surveillance cameras for remote monitoring and intelligent decision making in the context of multidisciplinary smart city applications

The extensive simulation results presented in this contribution indicated that the proposed techniques take advantage of the fact that motion in any video frame is usually coherent, hence dictating a high probability of a macroblock having the same direction of motion as the macroblocks surrounding it. Two previous neighboring MBs (above and left) have been utilized to find the first step of the search process. The aim of using these neighboring MBs is to improve the process of finding the global matching MB and to reduce unnecessary computations by selecting three previous neighboring MBs. To speed up the initial calculations, the proposed algorithms utilize mean value of motion vectors of these neighboring macroblocks.

The use of MVs predictors led to enhancing the probability of finding the global minimum in the first search. Therefore, we used partial distortion elimination algorithm to enhance and reduce processing times. For lossless block matching algorithm, the performance of the proposed algorithm is assessed using the initial calculation to determine the new search window. The new search window will contain the global minimum; hence, applying the PDE algorithm improves the search process. Moreover, investigating the remaining macroblocks of the search windows is not be needed when the error of the matching macroblock from this search is smaller than the previous determined minimum error value. The proposed technique makes use of three types of fast block matching algorithms: predictive search, fixed set of search patterns, and partial distortion elimination to further improve quality and reduce computational requirements. As discussed in [59], one of the major challenges associated with the high temporal satellite or UAVs video data processing is the trade-off between the limited transmission bandwidth and high rate data streaming. An optimal video compression algorithm is essential in reducing the transmission load and simultaneously maintaining video quality, which is important in enhancing the quality of the data representation model and the reliability of intelligent monitoring and decision-making systems. A typical example of such a scenario is the previously described video surveillance-based identification of crime suspects (in the context of the smart city system) [12,13,60] producing staggeringly inaccurate and dramatically high false positive rates (around 98%). One of the main reasons for such a failure is most probably the poor quality of transmitted data needed by face matching algorithms.

In our experiments, we utilized a number of video sequence types and the simulation results indicated that the proposed techniques demonstrate improved results in comparison to the state-of-the-art lossless and lossy block matching algorithms. These improvements were measured in terms of processing times for lossless block matching algorithm. While for the lossy block matching algorithm, the average number of search points required per macroblock, and the residual prediction errors in comparison to the standard fixed set of the search pattern of block matching algorithms, were used, and demonstrated significant improvements.

6. Conclusions

In this work, a novel block matching video compression technique is proposed for lossless and lossy compression. The simulation results indicate that the proposed algorithm, when used as a lossless block matching algorithm, reduces the search times in macroblock matching, while preserving the resolution of the predicted frames. Moreover, the simulation results for the lossy block matching algorithm show improvements in computational complexity, and enhanced resolution, when compared with the benchmarked algorithms which is vital for maintaining the high quality representations of remote sensing data. The proposed algorithm can be utilized in interdisciplinary innovative technologies, specifically related to future smart city applications, where high-resolution remote sensing data, specifically temporal video data, is needed to be transmitted and processed automatically for reliable representation, analysis and intelligent decision making. In future work, we aim to explore the use of the proposed algorithm in autonomous outdoor mobility assistance system for visually impaired people, based on live streaming data from multiple sensors and video devices, and compare its performance with state-of-the-art techniques.

Author Contributions: Conceptualization, A.J.H.; Formal analysis, W.K.; Investigation, Z.A. and A.J.H.; Methodology, Z.A.; Resources, H.A.-A.; Supervision, P.L.; Validation, A.J.H.; Visualization, T.B. and R.A.-S.; Writing – original draft, Z.A., A.J.H. and W.K.; Writing – review & editing, J.L., P.L. and D.A.-J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nations, U. *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables*; Departement of Economic and Social Affaire: New York City, NY, USA, 2017.

2. Bulkeley, H.; Betsill, M. Rethinking sustainable cities: Multilevel governance and the'urban'politics of climate change. *Environ. Politics* **2005**, *14*, 42–63. [CrossRef]
3. Lyer, R. Visual loT: Architectural Challenges and Opportunities. *IEEE Micro* **2016**, *36*, 45–47.
4. Andreaa, P.; Caifengb, S.; I-Kai, W.K. Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. *J. Ambient Intell. Smart Environ.* **2019**, *11*, 5–22.
5. Goyal, A. Automatic Border Surveillance Using Machine Learning in Remote Video Surveillance Systems. In *Emerging Trends in Electrical, Communications, Information Technologies*; Hitendra Sarma, T., Sankar, V., Shaik, R., Eds.; Lecture Notes in Electrical Engineering; Springer: Berlin/Heidelberg, Germany, 2020; Volume 569.
6. Hodges, C.; Rahmani, H.; Bennamoun, M. Deep Learning for Driverless Vehicles. In *Handbook of Deep Learning Applications. Smart Innovation, Systems and Technologies*; Balas, V., Roy, S., Sharma, D., Samui, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 136, pp. 83–99.
7. Yang, H.; Lee, Y.; Jeon, S.; Lee, D. Multi-rotor drone tutorial: Systems, mechanics, control and state estimation. *Intell. Serv. Robot.* **2017**, *10*, 79–93. [CrossRef]
8. Australian Casino Uses Facial Recognition Cameras to Identify Potential Thieves—FindBiometrics. FindBiometrics. 2019. Available online: https://findbiometrics.com/australian-casino-facial-recognition-cameras-identify-potential-thieves/ (accessed on 9 May 2019).
9. Lu, H.; Gui, Y.; Jiang, X.; Wu, F.; Chen, C.W. Compressed Robust Transmission for Remote Sensing Services in Space Information Networks. *IEEE Wirel. Commun.* **2019**, *26*, 46–54. [CrossRef]
10. Fan, Y.; Shang, Q.; Zeng, X. In-Block Prediction-Based Mixed Lossy and Lossless Reference Frame Recompression for Next-Generation Video Encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 112–124.
11. Sharman, J. Metropolitan Police's facial recognition technology 98% inaccurate, figures show. 2018. Available online: https://www.independent.co.uk/news/uk/home-news/met-police-facial-recognition-success-south-wales-trial-home-office-false-positive-a8345036.html (accessed on 5 April 2019).
12. Burgess, M. Facial recognition tech used by UK police is making a ton of mistakes. 2018. Available online: https://www.wired.co.uk/article/face-recognition-police-uk-south-wales-met-notting-hill-carnival (accessed on 11 March 2019).
13. Blaschke, B. 90% of Macau ATMs now fitted with facial recognition technology—IAG. IAG, 2018. [Online]. Available online: https://www.asgam.com/index.php/2018/01/02/90-of-macau-atms-now-fitted-with-facial-recognition-technology/ (accessed on 9 May 2019).
14. Chua, S. Visual loT: Ultra-Low-Power Processing Architectures and Implications. *IEEE Micro* **2017**, *37*, 52–61. [CrossRef]
15. Fox, C. Face Recognition Police Tools 'Staggeringly Inaccurate'. 2018. Available online: https://www.bbc.co.uk/news/technology-44089161 (accessed on 9 May 2019).
16. Ricardo, M.; Marijn, J.; Devender, M. Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. *Gov. Inf. Q.* **2018**. [CrossRef]
17. Gashnikov, M.V.; Glumov, N.I. Onboard processing of hyperspectral data in the remote sensing systems based on hierarchical compression. *Comput. Opt.* **2016**, *40*, 543–551. [CrossRef]
18. Liu, Z.; Gao, L.; Liu, Y.; Guan, X.; Ma, K.; Wang, Y. Efficient QoS Support for Robust Resource Allocation in Blockchain-based Femtocell Networks. *IEEE Trans. Ind. Inform.* **2019**. [CrossRef]
19. Wang, S.Q.; Zhang, Z.F.; Liu, X.M.; Zhang, J.; Ma, S.M.; Gao, W. Utility-Driven Adaptive Preprocessing for Screen Content Video Compression. *IEEE Trans. Multimed.* **2017**, *19*, 660–667. [CrossRef]
20. Lu, Y.; Li, S.; Shen, H. Virtualized screen: A third element for cloud mobile convergence. *IEEE Multimed. Mag.* **2011**, *18*, 4–11. [CrossRef]
21. Kuo, H.C.; Lin, Y.L. A Hybrid Algorithm for Effective Lossless Compression of Video Display Frames. *IEEE Trans. Multimed.* **2012**, *14*, 500–509.
22. Jaemoon, K.; Chong-Min, K. A Lossless Embedded Compression Using Significant Bit Truncation for HD Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 848–860. [CrossRef]
23. Srinivasan, R.; Rao, K.R. Predictive coding based on efficient motion estimation. *IEEE Trans. Commun* **1985**, *33*, 888–896. [CrossRef]
24. Huang, Y.-W.; Chen, C.-Y.; Tsai, C.-H.; Shen, C.-F.; Chen, L.-G. Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. *J. Vlsi Signal Process.* **2006**, *42*, 297–320. [CrossRef]

25. Horn, B.; Schunck, B. Determining Optical Flow. *Artifical Intell.* **1981**, *17*, 185–203. [CrossRef]
26. Richardson, I.E.G. *The H.264 Advanced Video Compression Standard*, 2nd ed.; John Wiley & Sons Inc: Fcodex Limited, UK, 2010.
27. ISO/IEC. Information Technology–Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s–Part 2: Video. Available online: https://www.iso.org/standard/22411.html (accessed on 1 February 2020).
28. ISO/IEC. Information Technology–Generic Coding of Moving Pictures and Associated Audio–Part 2: Video. Available online: https://www.iso.org/standard/61152.html(accessed on 1 February 2020).
29. ITU-T and ISO/IEC. Advanced Video Coding for Generic Audiovisual Services; H.264, MPEG, 14496–10. 2003. Available online: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11466(accessed on 1 February 2020).
30. Sullivan, G.; Topiwala, P.; Luthra, A. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In Proceedings of the SPIE conference on Applications of Digital Image Processing XXVII, Denver, CO, USA, 2–6 August 2004.
31. Sullivan, G.J.; Wiegand, T. Video Compression—From Concepts to the H.264/AVC Standard. *Proc. IEEE* **2005**, *93*, 18–31. [CrossRef]
32. Ohm, J.; Sullivan, G.J. High Efficiency Video Coding: The Next Frontier in Video Compression [Standards in a Nutshell]. *IEEE Signal Process. Mag.* **2013**, *30*, 152–158. [CrossRef]
33. Suliman, A.; Li, R. Video Compression Using Variable Block Size Motion Compensation with Selective Subpixel Accuracy in Redundant Wavelet Transform. *Adv. Intell. Syst. Comput.* **2016**, *448*, 1021–1028.
34. Kim, C. *Complexity Adaptation in Video Encoders for Power Limited Platforms*; Dublin City University: Dublin, Ireland, 2010.
35. Suganya, A.; Dharma, D. Compact video content representation for video coding using low multi-linear tensor rank approximation with dynamic core tensor order. *Comput. Appl. Math.* **2018**, *37*, 3708–3725.
36. Chen, D.; Tang, Y.; Zhang, H.; Wang, L.; Li, X. Incremental Factorization of Big Time Series Data with Blind Factor Approximation. *IEEE Trans. Knowl. Data Eng.* **2020**. [CrossRef]
37. Yu, L.; Wang, J.P. Review of the current and future technologies for video compression. *J. Zhejiang Univ. Sci. C* **2010**, *11*, 1–13. [CrossRef]
38. Onishi, T.; Sano, T.; Nishida, Y.; Yokohari, K.; Nakamura, K.; Nitta, K.; Kawashima, K.; Okamoto, J.; Ono, N.; Sagata, A.; et al. A Single-Chip 4K 60-fps 4:2:2 HEVC Video Encoder LSI Employing Efficient Motion Estimation and Mode Decision Framework with Scalability to 8K. *IEEE Trans. Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 1930–1938. [CrossRef]
39. Barjatya, A. *Block Matching Algorithms for Motion Estimation*; DIP 6620 Spring: Berlin/Heidelberg, Germany, 2004.
40. Ezhilarasan, M.; Thambidurai, P. Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression. *J. Comput. Sci.* **2008**, *4*, 282–289.
41. Sayood, K. *Introduction to Data Compression*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2006.
42. Hussain, A.; Ahmed, Z. A survey on video compression fast block matching algorithms. *Neurocomputing* **2019**, *335*, 215–237. [CrossRef]
43. Shinde, T.S.; Tiwari, A.K. Efficient direction-oriented search algorithm for block motion estimation. *IET Image Process.* **2018**, *12*, 1557–1566. [CrossRef]
44. Xiong, X.; Song, Y.; Akoglu, A. Architecture design of variable block size motion estimation for full and fast search algorithms in H.264/AVC. *Comput. Electr. Eng.* **2011**, *37*, 285–299. [CrossRef]
45. Al-Mualla, M.E.; Canagarajah, C.N.; Bull, D.R. *Video Coding for Mobile Communications: Efficiency, Complexity and Resilience*; Academic Press: Cambridge, MA, USA, 2002.
46. Koga, T.; Ilinuma, K.; Hirano, A.; Iijima, Y.; Ishiguro, Y. *Motion Compensated Interframe Coding for Video Conferencin*; the Proc National Telecommum: New Orleans, LA, USA, 1981.
47. Reoxiang, L.; Bing, Z.; Liou, M.L. A new three-step search algorithm for block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **1994**, *4*, 438–442. [CrossRef]
48. Lai-Man, P.; Wing-Chung, M. A novel four-step search algorithm for fast block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **1996**, *6*, 313–317. [CrossRef]

49. Shan, Z.; Kai-Kuang, M. A new diamond search algorithm for fast block matching motion estimation. In Proceedings of the ICICS, 1997 International Conference on Information, Communications and Signal Processing, Singapore, 12 September 1997; Volume 1, pp. 292–296.
50. Jianhua, L.; Liou, M.L. A simple and efficient search algorithm for block-matching motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **1997**, *7*, 429–433. [CrossRef]
51. Nie, Y.; Ma, K.-K. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Trans Image Process.* **2002**, *11*, 1442–1448. [CrossRef] [PubMed]
52. Yi, X.; Zhang, J.; Ling, N.; Shang, W. Improved and simplified fast motion estimation for JM (JVT-P021). In Proceedings of the Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) 16th Meeting, Poznan, Poland, July 2005.
53. Ananthashayana, V.K.; Pushpa, M.K. joint adaptive block matching search algorithm. *World Acad. Sci. Eng. Technol.* **2009**, *56*, 225–229.
54. Jong-Nam, K.; Tae-Sun, C. A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan. *IEEE Trans. Circuits Syst. Video Technol.* **2000**, *10*, 1040–1048. [CrossRef]
55. Chen-Fu, L.; Jin-Jang, L. An adaptive fast full search motion estimation algorithm for H.264. *Ieee Int. Symp. Circuits Syst. ISCAS* **2005**, *2*, 1493–1496.
56. Jong-Nam, K.; Sung-Cheal, B.; Yong-Hoon, K.; Byung-Ha, A. Fast full search motion estimation algorithm using early detection of impossible candidate vectors. *IEEE Trans. Signal Process.* **2002**, *50*, 2355–2365. [CrossRef]
57. Jong-Nam, K.; Sung-Cheal, B.; Byung-Ha, A. Fast Full Search Motion Estimation Algorithm Using various Matching Scans in Video Coding. *IEEE Trans. Syst. Mancybern. Part C Appl. Rev.* **2001**, *31*, 540–548. [CrossRef]
58. Xiao, J.; Zhu, R.; Hu, R.; Wang, M.; Zhu, Y.; Chen, D.; Li, D. Towards Real-Time Service from Remote Sensing: Compression of Earth Observatory Video Data via Long-Term Background Referencing. *Remote Sens.* **2018**, *10*, 876. [CrossRef]
59. Allauddin, M.S.; Kiran, G.S.; Kiran, G.R.; Srinivas, G.; Mouli GU, R.; Prasad, P.V. Development of a Surveillance System for Forest Fire Detection and Monitoring using Drones. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 9361–9363.
60. Kuru, K.; Ansell, D.; Khan, W.; Yetgin, H. Analysis and Optimization of Unmanned Aerial Vehicle Swarms in Logistics: An Intelligent Delivery Platform. *IEEE Access* **2019**, *7*, 15804–15831. [CrossRef]