

A Hybrid Tactic Model Intended for Video Compression Using Global Affine Motion and Local Free-Form Transformation Parameters

D. Raveena Judie Dolly¹ · G. Josemin Bala¹ · J. Dinesh Peter² 

Received: 2 May 2017 / Accepted: 4 October 2017 / Published online: 3 November 2017
© King Fahd University of Petroleum & Minerals 2017

Abstract Video compression marks its necessity when a huge sized video needs to be transmitted. The process starts with the identification of GoP (group of pictures), which depends on I- (intra), B- (bidirectional) and P- (predicted) frames determination. GoP is fixed, where consecutive frames are placed in an orderly manner based on the GoP size. Conventionally, B-frames lead to buffering of memory within the past and future frames consuming more computational time. Such issues are handled by an adaptive framework for determining frames based on matching criteria rather than fixed GoP. NSEW (North–South–East–West) affine translation (NAT) is proposed for replacing B with either I- or P-frame. The proposed framework involves video compression using affine motion-based free-form transformation and video decompression using warping methodologies for the purpose of compressing and decompressing the video sequence, based on the resulted I- and P-frames. B-spline transformation was also initiated at local level along with global affine transformation to improve the subjective quality of the decompressed video sequence. The methodology was investigated for the file size, computational time, peak-signal-to-noise ratio (PSNR) and Structural Similarity index (SSIM), which proved the superiority of the proposed technique. Further, the methodology was also investigated with

optimizing the affine motion parameters (AMP) using non-linear least squares, Broyden–Fletcher–Goldfarb–Shanno (BFGS) and limited-memory BFGS which yet again proved to be far more superior to conventional techniques with an average PSNR of 38.98 dB with LBFGS. To further improve the subjective quality, affine B-spline-based motion estimation using LBFGS was implemented and observed the average PSNR gain to be 42.03 dB.

Keywords Video compression · Group of pictures (GoP) · Adaptive frame determination · Affine motion parameters · Optimization · B-Splines

1 Introduction

The movement of still images called frames constitutes a video. The drastic or small changes due to camera motion can be observed in the consecutive frames of a video sequence which are categorized into intra- and inter- [P- and B-] frames, respectively. I-frames constitute the key reference frame for successive frames. P-frames can be predicted from a preceding I- or P-frames. The B-frames are interpolated from forward and backward frames.

The choice of the frame type decides the quality and compression ratio of the compressed video. The quality of video seems better when there are many number of intra-frames. On the contrary, when there are more number of inter-frames, the compression ratio will be better sacrificing the quality of the video. The frames in a GoP follow IP or IBP or IBBP frame patterns. Conventional algorithms employ prediction analysis for MPEG-4 and H.264 to reduce data between a series of frames. They are realized by a process called motion estimation and motion compensation. The correlation between two frames in terms of motion is represented by a motion vector.

✉ J. Dinesh Peter
dineshpeter@karunya.edu

D. Raveena Judie Dolly
dollydinesh@karunya.edu

G. Josemin Bala
josemin@karunya.edu

¹ Department of Electronics and Communication Engineering, Karunya University, Coimbatore, India

² Department of Computer Sciences Technology, Karunya University, Coimbatore, India

By adopting this technique, the encoded and the transmitted number of pixel values are considerably reduced. This process is a computationally intensive operation. Conventionally, if the file size is lowered by raising the compression level, the visual quality gets affected. The proposed video compression method involves affine parameter estimation for motion estimation and affine warping for motion compensation where the motion parameters are estimated and stored as compressed data. To further enhance the quality of the video, B-splines transformation is experimented at local level.

2 Related Works

The video compression standards follow I-, B- and P-frame choices. The GoP has an intra-frame with consequent inter-frames. To maximize the coding efficiency determination of frame types becomes necessary. Wang et al. 2005 [1] highlights that the GoP structure can be determined dynamically in real time by using the information obtained from the sum of absolute difference (SAD) and the mean of absolute difference (MAD). Paul et al. 2013 [2] insist in adopting multiple I-frames being the reference frames (MRFs) for improving the coding performance of the H.264 video coding standard over other contemporary video coding standards. For a video with repetitive motion, dynamic backgrounds and illumination changes, MRFs provide better predictions than single reference frame. On the contrary, the existence of more I-frames drastically reduces the coding performance for a video sequence which does not contain any scene change compared to the previous frames. Ohm et al. 2012 [3] suggests that if a nominal scene change is identified in between frames, the concerned frame is marked as I-frame. Therefore, an optimal number of I-frames have to be selected carefully in order to get better compression ratio without compromising the quality of decompression.

Mathew 2010 [4] in Texas Instruments application stated that B-frame requires a reference from the previous and future frames. Therefore, conventional standards adopt encoding the frame that follows a B-frame before encoding the B-frame. Moreover, the delay is increased with more number of B-frames. Paul et al. 2014 [5] suggested that when motions in the video source are complex, avoiding B-frames results in better performance. Apart from that, MPEG-2 code was also implemented and tested with and without B-frames. The compression time taken for *Rhinos* video using MPEG-2 in *Matlab* was about 60.5 s with B-frames. When all the B-frames were converted to P-frames, the time consumed was found to be 40.5 s. For the purpose of testing, *Vipmosaicking* video was considered and it was observed that it took 57 s with B-frames and 41 s without B-frames. Hence, replacement of B-frames was acclaimed.

Tabatabai et al. 1998 [6] suggested that affine transformation can be used for co-ordinate transformation in video compression as a future enhancement work. Lu et al. 2008 [7] proposed a new *Gestalt factor* called “*shift and hold*” for 2D object extraction algorithms. This ignited the authors in this proposed work for performing shifting and correlation for the replacement of B-frames. T Wiegand et al. 2005 [8] highlighted the utilization of affine motion model for traditional motion-compensated coders incorporating transformations such as rotation and shear, allowing higher adaptability to the underlying image.

Theoharis et al. 2008 [9] suggest that when an affine transformation is applied to a B-spline curve, it is sufficient only if the control points are transformed. Alavala 2009 [10] states that the process becomes simple when the transformation is applied to control points rather than applying to the curve. The proposed methodology is focused with adaptive frame determination. Once all the frames are adaptively identified as either I-, P- or B-frames, the proposed NSEW affine translation method described in Sect. 3.2 is imposed to replace all the B-frames with either I-frame or P-frame. A novel methodology using global affine motion parameters and local B-Spline parameters is proposed, where the global motion and local control parameters are found between first reference frame which is the I-frame and the adjacent P-frames until another I-frame is encountered. The process continues by considering the next I-frame as the reference frame. Apart from the global affine transformation, the B-spline transformation is applied locally for the quality enhancement of decompressed video. Both the global and local transformation parameters along with I-frames are stored during the process. A grid of b-spline control points is constructed which controls the local transformation of the video sequence. While decoding, image warping method utilizes these stored transformation parameters. The estimated parameters are applied to I-frame using interpolation operation for obtaining respective P-frames. Thus the complete set of frames are obtained for a particular video sequence. Nonlinear least squares, Broyden–Fletcher–Goldfarb–Shanno (BFGS) and limited-memory BFGS optimization methods are performed for refining the AMPs.

3 Proposed Methodology

The frames are adaptively identified during the initial stage instead of randomly selecting I-, B- and P-frames, as portrayed in Fig. 1. For an input video sequence, the B-frames are considered for replacement in order to reduce the buffering of memory between the past and future frames by using the proposed NSEW affine translation (NAT). The proposed methodology is applied to obtain global affine motion parameters and B-spline-based local control point between the

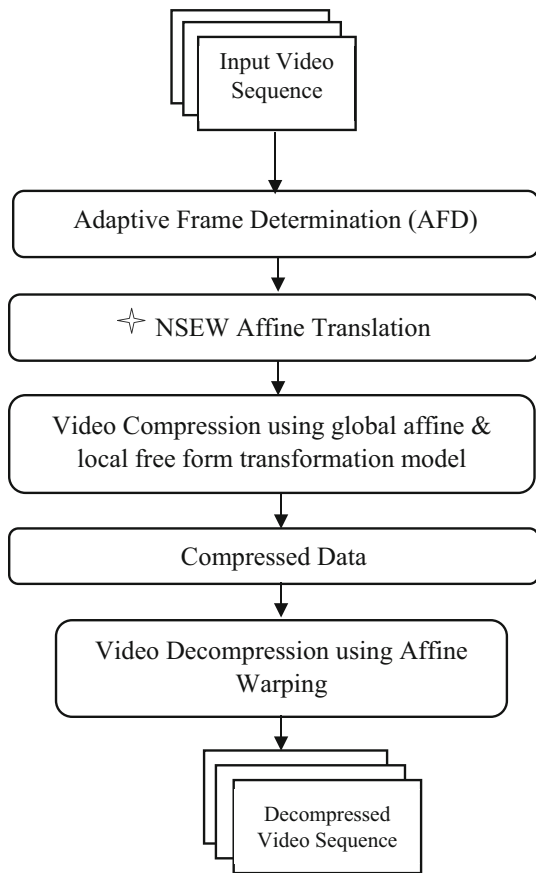


Fig. 1 Process flow of compression and decompression

successive frames. Block matching strategies, though being easy to implement, could produce poor results when multiple moving objects exist in one block.

The motion and control points estimation model estimates not only translational motions but also the non-translational motions such as zooming, rotation. The global motion and local control point estimation model are performed between I- and P-frames until another I-frame is encountered. These parameters are stored in a file with corresponding frame indexes along with JPEG encoded I-frames which is considered to be the compressed data. The resulted affine motion parameters are applied to the I-frame in order to get back the P-frame using a warping approach from which the decompressed video sequence is obtained.

3.1 Adaptive Frame Determination (AFD)

The investigation was carried out using adaptive frame determination (AFD) to improve the quality of video coding as suggested by Ding et al. 2008 [11], using test video sequences taken from a standard database [12]. Each frame of an input video sequence \forall is determined and saved with specific frame index numbers N_i . The first frame of a video sequence

N_1 is considered to be the I-frame. The similarity between the I-frame and its adjacent frame is computed and identified as either P- or B-frame. This process of frame determination is continued by computing the correlation between the adjacent frames. Correlation co-efficient, a matching criterion, is employed for estimating the correlation between frames, where both the images are normalized by subtracting the mean intensity value in each frame as given in Eq. (1).

$$C = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(I_{mn} - \bar{I})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2 (\sum_m \sum_n (I_{mn} - \bar{I})^2)}} \quad (1)$$

Here, A and I are matrices or vectors having the same size where $\bar{A} = \text{mean}(A)$ and $\bar{I} = \text{mean}(I)$. When two frames are exactly similar, the correlation factor would be “1”. The thresholds are fixed in such a way that the correlation between adjacent frames is considered based on the type of video sequence. The high-level pseudo-code for AFD is given below.

Based on the critical analysis of various motion types, the thresholds for (m_1, m_2) are empirically fixed as $(0.7, 0.9)$ for still camera | still background $(0.6, 0.8)$ for still camera | moving background and $(0.5, 0.75)$ for moving camera | moving background. The I-, B- and P-frames are adaptively determined based on scene changes. The first frame of a video is taken as the reference frame, and the succeeding frame is taken as the target frame. Correlation between the reference and the target frame is computed, and if the correlation factor exceeds a threshold value of m_2 , the target frame is considered as P-frame. If it lies between m_1 and m_2 , the target frame is considered as B-frame, else if it is less than m_1 , the target frame is considered as I-frame.

3.2 NSEW Affine Translation (NAT) for B-Frame Replacement

NSEW affine translation (NAT) method is proposed in order to avoid time delay in processing B-frames during compression and decompression. B-frames are replaced with either I- or P-frames using this method. Translation is executed in all possible directions with two fixed directional parameters b_1

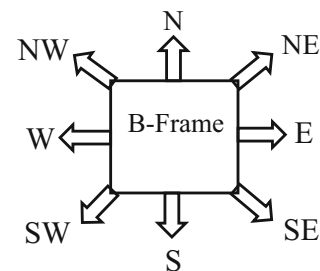
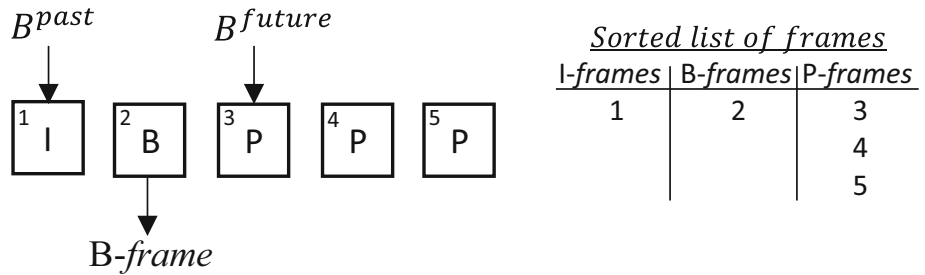


Fig. 2 Directions of translations

Fig. 3 Sorted list of frames and identification of B^{past} and B^{future} frames



and b_2 . The conventional affine transformation matrix is commonly written in homogeneous coordinates where (x_1, y_1) are considered as pixel intensity values located at a particular position in an input image and (x_2, y_2) represents that of an output image frame. The affine transformation is defined in Eq. (2).

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \times \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + T \tag{2}$$

This transformation executes the translation part alone, by utilizing the predefined T-matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{3}$$

In NAT, the two translation parameters b_1 and b_2 are provided with two fixed values each. In our investigations, the translation parameters are considered empirically as $b_1 \rightarrow \{10, -10\}$ and $b_2 \rightarrow \{10, -10\}$. The various directions of translations are North (N), South (S), East (E), West (W), North–West (NW), South–West (SW), South–East (SE), North–East (NE). This ideology is proposed in order to accomplish an optimum variable GoP and also to provide significant compression and decompression results. The directions of translation are portrayed in Fig. 2.

There exists an issue in NAT, that the transformed image may be out focused with respect to the transformation direction. Therefore, cropping technique is adopted for automatically cropping the region of interest based on the size of the image frame. The dimension of the cropped region is adaptively chosen. It is understood that the object of interest appears in the middle of the image frames of all video sequences. Therefore, the dimension of the cropped image is considered from the center of the image frame, excluding the outer portion of the image frame. The same nomenclature is used for cropping B^{Past} and B^{future} images. B^{Past} and B^{future} frames are either I- or P-frames with respect to frame index numbers which is depicted in Fig. 3.

These frames are considered for processing due to the proximity of I- and P-frames with respect to B-frames. The proposed framework for getting an optimal set of GoP using NAT is shown in Fig. 4.

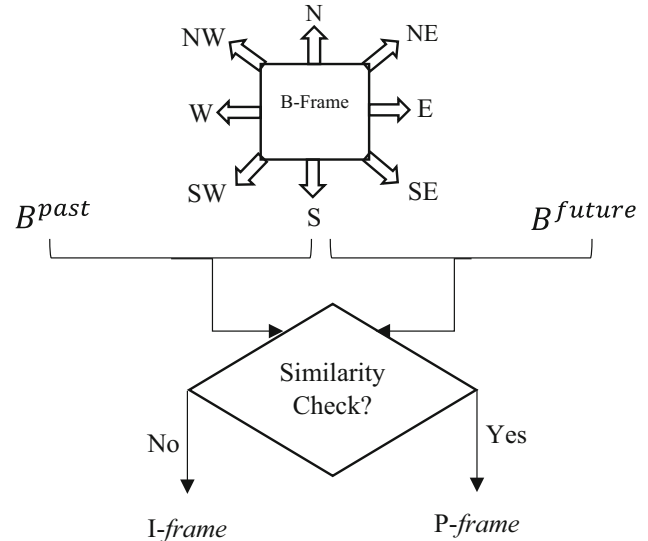


Fig. 4 NSEW affine translation

The cropped regions from B^{Past} and B^{future} frames with respect to cropped B-frame are considered for similarity check using cross-correlation technique. There exist 8 cropped B-frames each from different directions of translation. The processed B-frame is converted to I-frame; if there exists a correlation below a certain threshold m_2 , the decision for converting to P-frame is considered if at least one among the 8 translated versions of B-frames is similar to B^{Past} or B^{future} and moreover the correlation must also exceed m_2 . A new sorted list of frames is developed based on the new identification of frames.

3.3 Video Compression Using Global Affine Motion and Local B-Spline Control Point Estimation

3.3.1 Global Affine Motion Estimation Model

This methodology highlights the utilization of affine transformation to estimate global motion parameter set. This affine motion parameter set is composed of linear transformations like translation, scaling, rotation and shearing defined by Eqs. (4), (5) and (6). The linear transformation for the case

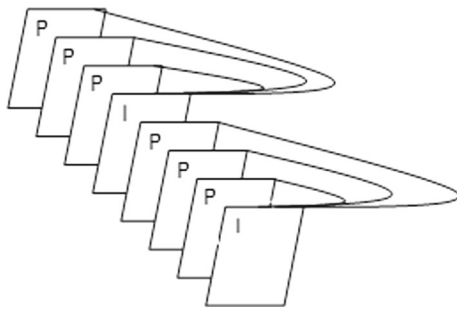


Fig. 5 Frame choice for performing affine transformation

of rotation can be achieved using the “A” matrix of Eq. (2) and can be mathematically represented as:

$$A = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix}, \quad T = \begin{vmatrix} 0 \\ 0 \end{vmatrix} \tag{4}$$

In terms of scaling s with respect to x and y axis,

$$A = \begin{vmatrix} s_x & 0 \\ 0 & s_y \end{vmatrix}, \quad T = \begin{vmatrix} 0 \\ 0 \end{vmatrix} \tag{5}$$

The linear transformation in terms of shearing sh with respect to x and y axis,

$$A = \begin{vmatrix} 1 & sh_x \\ sh_y & 1 \end{vmatrix}, \quad T = \begin{vmatrix} 0 \\ 0 \end{vmatrix} \tag{6}$$

The affine transformation matrix is denoted by six parameters, which are the two scale factors P_1 and P_4 , the two shearing factors P_2 and P_3 , and the two translation factors P_5 and P_6 . The affine transformation matrix can be defined as,

$$\begin{pmatrix} P_1 & P_2 & P_5 \\ P_3 & P_4 & P_6 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Once all the B-frames are converted into either I- or P-frames utilizing the proposed framework as mentioned in Sect. 3.2, the frames are arranged as per the new GoP generated order. There exist only I- and P-frames in the new GoP. At this stage, an affine transformation is employed between I-frame and P-frame as represented in Fig. 5.

The first I-frame is taken as the reference frame, and the next P-frame is taken as the target frame. The image frames are represented as $I_{x,y,t}$ (reference) and $I_{x,y,t-1}$ (target). Since the reference and target frames are from the same video sequence \forall , the chosen error function is the sum of squared difference (SSD). This error function is simple to compute and provides good minimization, and it also yields good parameter estimation. The error function can be written as:

$$E(\vec{P}) = \sum_{x,y} [I_{x,y,t} - I_{P_1x+P_2y+P_5, P_3x+P_4y+P_6,t-1}]^2 \tag{8}$$

Since the error function is nonlinear within the unknown affine parameters, it cannot be minimized analytically. Therefore, first order truncated Taylor series approximation is applied to obtain an approximate expression for the error [4] which is defined by:

$$E(\vec{P}) \approx \sum_{x,y} \left[I'_{x,y,t} - (P_1x + P_2y + P_5 - x) I''_{x,y,t} - (P_3x + P_4y + P_6 - y) I''_{x,y,t} \right]^2 \tag{9}$$

where $I''_{x,y,t}$ and $I''_{x,y,t}$ are spatial derivatives and $I'_{x,y,t}$ is the temporal derivative. The simplified form of this quadratic error norm can be written as

$$E(\vec{P}) = \sum_{x,y} [r - \vec{q}^T \vec{P}]^2 \tag{10}$$

where the vector \vec{q} is defined as $\vec{q}^T = [xI^x \ yI^x \ xI^y \ yI^y \ I^x \ I^y \ -I \ -1]$ and the scalar r is defined as $r = (I^t + xI^x + yI^y - I)$. This error function can now be minimized analytically by deducing the partial derivative with respect to its unknown parameters.

$$\frac{\partial E(\vec{P})}{\partial \vec{P}} = \sum_{x,y} \frac{\partial}{\partial \vec{P}} (r - \vec{q}^T \vec{P})^2 \tag{11}$$

$$\frac{\partial E(\vec{P})}{\partial \vec{P}} = \sum_{x,y} -2q(r - \vec{q}^T \vec{P}) \tag{12}$$

By equating Eq. (12) to zero, the solution for solving \vec{P} yields

$$\vec{P} = \left[\sum_{x,y} (\vec{q}\vec{q}^T) \right]^{-1} \left[\sum_{x,y} (\vec{q}r) \right] \tag{13}$$

An estimate of the error function during minimization can be regulated using iterated reweighted least squares technique (IRLS), where the weights are being modified on each iteration. During the implementation, the estimated parameters are fine-tuned on each iteration as follows

$$\vec{P}_{(j+1)} = \left[\sum_{x,y} (\vec{q}\omega(e_f, \sigma)\vec{q}^T) \right]^{-1} \left[\sum_{x,y} \vec{q}\omega(e_f, \sigma)r \right] \tag{14}$$

where $\vec{P}_{(j+1)}$ is the estimated affine parameter set at iteration $(j + 1)$. $\omega(e_f, \sigma)$ is the diagonal matrix whose diagonal elements are the weights of the chosen weight function and $e_f = r - \vec{q}^T \vec{P}$ is the residual. The iteration stops when the

average total squared error is less than 0.1, or whenever it reaches the maximum number of iteration, 100. At the final step of motion estimation, the error function is minimized conceding the desired affine motion parameters.

3.3.2 Local B-Spline Control Point Estimation Model

The word “spline” specifies that, it can smoothly pass through a set of points. Every control point is defined as an affine transformation. The objective function is formulated by a bidirectional distance cost, where errors from reference image to target image and from target image to reference image are considered. Thereby, sparse linear system is obtained.

A transformation from the target image to the reference image is defined as a mapping Suicheng Gu et al. 2014 [13], Szeliski et al. 1997 [14]. Assuming a point $p_u = [x_u, y_u, z_u]^T$ in a target image where $0 \leq x_u, y_u, z_u \leq 1$, its mapping $f(p_u)$ in the reference image is computed using

$$f(p_u) = A(p_u)\hat{p}_u \tag{15}$$

where $\hat{p}_u = [x_u, y_u, z_u, 1]^T$, T denotes transpose of a matrix. $A(p_u)$ is an affine transformation matrix function with a size of 3×4 , which is the weighted sum of a set of affine transformation matrices:

$$A(p_u) = \sum_{i=1}^n g(p_u, B_v)B_v \tag{16}$$

$\{B_v\}_{v=1}^M$ are the control points and each control point is a 3×4 affine transformation matrix. $g(p_u, B_v) \geq 0$ is the weight of the pair of point p_u and control point B_v . For each point p_u , $\sum_{i=0}^n g(p_u, B_v) = 1$. Here, the weighting function $g(p_u, B_v)$ is defined by the quadratic B-spline basis, and the model is called B-Spline affine transformation. The BSAT model is a more generalized model of the B-Spline transformation (BST), where every control point is considered as a displacement/deformation. If the left three columns of the transformation matrix B_v of each control point are replaced

with an identity matrix $\begin{bmatrix} 1 & 0 & 0 & dx_v \\ 0 & 1 & 0 & dy_v \\ 0 & 0 & 1 & dz_v \end{bmatrix}$, then the B-Spline Affine

transformation model degenerates to a B-spline transformation model. The B-spline affine transformation model is also a more generalized model of the affine transformation. If all the transformation matrices $\{B_v\}_{v=1}^M$ have the same elements, this will degenerate to an affine. For every control point, the BSAT model has twelve parameters. Such that B-spline affine transformation model delivers more complex transformation (freedom) than the BST with the same set of control points. The free-form deformation (FFD) entails of collective local and global motion model at every point (u, v, w) and can be

expressed as,

$$T(u, v, w) = T_{\text{global}}(u, v, w) + T_{\text{local}}(u, v, w) \tag{17}$$

Global transformation model pronounces the entire motion of the object. The best choice is a rigid transformation which is parameterized by 6 degrees of freedom, describing the rotations and translations of the object. The affine transformation, describing translation, rotation, scaling and shearing, is performed globally. To acquire better accuracy, the B-splines model is preferred since the deformation is much reduced. This is done by handling the underlying mesh of control points. The domain of the image volume is denoted as $\Omega = \{(u, v, w) | 0 \leq u < U, 0 \leq v < V, 0 \leq w < W\}$. The FFD can be written in 1-D cubic B-splines with $\varphi_{i,j,k}$ having uniform spacing,

$$T_{\text{local}}(u, v, w) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(a)B_m(b)B_n(c)\varphi_{i+1,j+m,k+n} \tag{18}$$

where, $i = \text{floor}(\frac{u}{n_x}) - 1$, $j = \text{floor}(\frac{v}{n_z}) - 1$, $a = (\frac{u}{n_x})\text{floor}(\frac{u}{n_x})$, $b = (\frac{v}{n_y})\text{floor}(\frac{v}{n_y})$, $c = (\frac{w}{n_z})\text{floor}(\frac{w}{n_z})$ and B_l represents the l^{th} basis functions of the B-spline. The control grid dimension is given as, $n_x \times n_y \times n_z$

$$B_0(a) = (1 - a)^3/6 \tag{19}$$

$$B_1(u) = (3u^3 - 6u^2 + 4)/6 \tag{20}$$

$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6 \tag{21}$$

$$B_3(u) = u^3/6 \tag{22}$$

Spline methods like thin plate splines, elastic body splines control globally. The change in $\varphi_{i,j,k}$ control point at a position alters most of the control points. B-splines are locally controlled where the change in $\varphi_{i,j,k}$ changes the local region alone. B-spline is efficient in computations even for larger control points. The B-spline free-form deformation considers the control points as its parameters. By altering the resolution of the mesh of control points φ , the degree of non-rigid deformation can be tuned. The resolution of the control point mesh depends on factors like degrees of freedom and computational complexity.

3.3.3 Novel Storage Structure for Storing Global Affine Motion Parameters and Local B-Spline Control Points

Estimated global affine motion parameters and local B-spline control points between the reference frame and the immediate succeeding P-frame need to be stored. This estimation process continues until it encounters the second

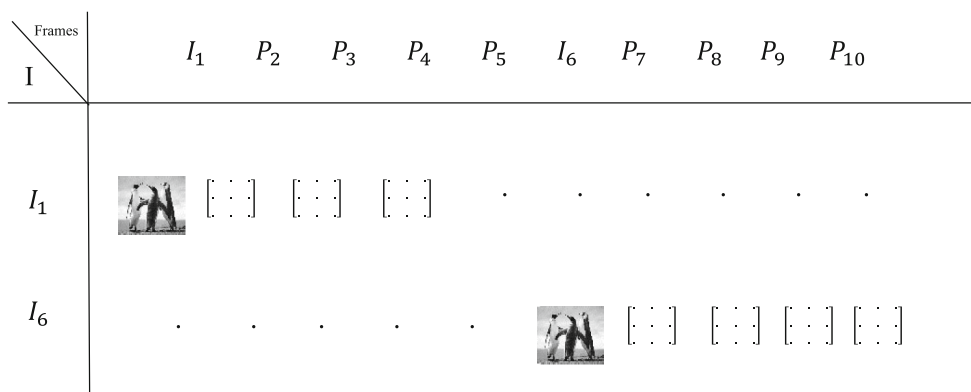


Fig. 6 The proposed storage structure for storing the estimated parameters (compressed data)

I-frame. Now, the second I-frame is considered as the reference frame for the next set of P-frames. The proposed nomenclature of stored estimated parameter’s pattern for $[I_1 P_2 P_3 P_4 P_5 I_6 P_7 P_8 P_9 P_{10}]$ is enlightened in Fig. 6, depicting the estimated parameter’s storage structure.

The JPEG encoded I-frames are stored as vector structure in respective cell arrays which are utilized for decompression. Once all the frames are processed, it is mandatory to remove the empty cell arrays. This structure can be stored as a file which is considered as the “*compressed data*”. To estimate the local deformation of the image, free-form deformation (FFD) model using B-spline is performed. Apart from the globally controlled method, where the change in control point at one position affects the position of most other control points, B-splines are locally controlled, where the change in control point affects the transformation only in the local region of the control point. They are computationally efficient even for a large number of control points [10]. This type of storage structure is easy to retrieve the consecutive frames for performing further inverse transformations.

3.4 Video Decompression Based on Warping

While decoding, the stored parameters are applied to the first decoded I-frame one by one in order to obtain the consecutive set of P-frames using affine and B-Spline inverse transformations and the process continues until the next decoded I-frame is encountered. Then the subsequent I-frame is considered and consecutive parameters are applied to obtain another set of P-frames. This process is repeated until the whole sequence of a video is obtained. This process is pictorially represented using a sample set of GoP in Fig. 6. The process of applying estimated parameters on I-frames to transform into reconstructed video sequence is referred as *Warping*.

Warping is a methodology employed for image manipulating operations. The target image pixels are constructed

by mapping their locations in the reference image. Generally, this inverse transformation does not result in exact pixel locations in the reference image. Therefore, target pixel values have to be computed by extracting and interpolating neighbor pixels by considering I-frame as the reference frame. P-frames are reconstructed based on the corresponding stored global motion parameters and local b-spline control points from *compressed data*. Interpolation techniques are assisted to achieve better approximations in finding target pixel values. *Bi-cubic* interpolation is proved to be a good interpolation technique which suits better for affine warping. The process flow of warping is depicted in Fig. 7. The Estimated parameters are applied to the reference frame so as to obtain the warped or the reconstructed frame.

The resulted P-frames are tested with the original frames, and the objective evaluations are done. In order to achieve optimized results, various optimization techniques are being used while estimating motion parameters and local control points.

4 Optimized Parameter Estimation for Global Affine Model

The choice of optimization techniques in parameter estimation is often a compromise between accuracy, robustness and computational cost. Optimization techniques to refine the AMPs, nonlinear least squares (LS) as mentioned by Gholipour et al. 2007 [15], Broyden–Fletcher–Goldfarb–Shanno (BFGS) by Xiao et al. 2008 [16] and limited-memory BFGS have been applied to the proposed approach for obtaining optimized results. In line search approach, the procedure picks a descent direction d_k and examines along this direction from the current point x_k for a lower or lowest function value gained at some distance α_k . Do until stopping criteria:

1. Determine a direction of search d_k ;

Fig. 7 The process flow of warping



- Find a step length α either as minimization problem (exact line search):

$$\min_{\alpha > 0} f(x_k + \alpha d_k)$$
or as an appropriate minimum approximation, $x_{k+1} = x_k + s_k d_k$.

Levenberg–Marquardt algorithm (LMA) is used for the purpose of solving nonlinear LS method. The LMA is found to be more robust than the Gauss–Newton algorithm as shown by Igarta [17]. The BFGS algorithm is an iterative method for solving unconstrained nonlinear optimization problems. BFGS has good performance even for non-smooth optimizations. This is considered to be effective rather than the most popular quasi-Newton update formula [1]. BFGS uses function values and gradients to build up a picture of the surface to be optimized. The success depends on how well the updating formula approximates the inverse of the true hessian at the current iteration. The hessian matrix is updated and stored, but in case of LBFGS, it stores information from past iterations to compute inverse hessian. LBFGS is a limited-memory version of BFGS which is suited to problems with very large numbers of variables. The conventional BFGS stores “ n ” approximations. Limited-memory BFGS (LBFGS) stores only a few vectors from which current values can be approximated. All the three suggested optimization techniques are applied for the estimation of optimized AMPs. After applying affine warping, the target P-frames are obtained and performances of these frames are analyzed.

5 Experimental Implementations and Discussions

The experimental analysis was made in Windows OS platform using MATLAB7 tool. The videos were taken from standard available databases [12] with video resolution: 176×144 (QCIF), frame rate: 25 frames/s and 5069 Kbps (bitrate). The standard video files are categorized based on the motion type as tabulated in Table 1. The benchmark HD (High-definition) sequences “*intoTree*” and “*oldTownCross*”

with 720p sequences at 18 Mbps were also taken into consideration for investigation.

5.1 Adaptive Frame Determination (AFD)

Based on the procedure discussed in Sect. 3.1, the adaptively determined frames are tabulated in Table 2. All the frames are sorted as I-, B-, and P-frames depending on the matching criteria mentioned in the procedure.










The analysis is made for the video from different categories such as still camera | still background, still camera | moving background and moving camera| moving background. From Table 2, it is obvious that for a still camera | still background videos, the similarity between adjacent frames is high. The existence of B-frames is high in high motion videos.

5.2 NSEW Affine Translation (NAT)

The main idea of NAT technique is to replace B-frames with either I or P-frames. Depending on the matching factor between frames and the procedure mentioned in Sect. 3.2, conversion of B-frames into either I- or P-frames is accomplished as shown in Table 3, which portrays about the replacement of B-frames with I- or P-frames.

Still camera | still background motion type is not considered for a replacement since they do not contain B-frames. Moreover, on the basis of observation, the B-frames of the still camera | moving background motion type is replaced with I-frames, but not converted to P-frames since these two video sequences encounter many new objects due to the constant motion of the background. In the case of *News* video sequence, the background keeps moving but a still camera is in place. If the camera is also on the move, there may be many B-frames taking the position of P-frames. Whenever there are nominal I-frames, the coding efficiency improves in terms of quality. It is observed that upon performing NAT methodology on the *moving camera | moving background* motion type, B-frames are replaced with either P or I-frames. *Coastguard* video sequence experiences many

Table 1 Categorization of test video files

#	Motion Type	Video File (Video Resolution: 176 × 144 (QCIF); Frame rate: 25 frames/sec; Bitrate: 5069 Kbps)
1	Still Camera Still Background	    Mother Container Akiyo Hall
2	Still Camera Moving Background	 News
3	Moving Camera Moving Background	  Foreman Coastguard
4	HD Video Sequence	  IntoTree OldTownCross

local dynamics and consequently the conversion to P-frames does not exist. HD sequence encounter nominal number of I-frames and many number of P-frames during the NAT process.

5.3 Performance Measures of the Proposed Technique

Evaluations based on various performance measures like file size, % file size saved and compression ratio are investigated. A comparative analysis of the original file size and the resulted file size after converting to grayscale is discussed. The first 100 frames in each video sequence are considered for performance analysis in order to ensure the uniformity among videos. The original file size of a video sequence is represented by the maximum number of frames that contained in that sequence. The resulted compressed data contain a number of JPEG encoded I-frames together with AMPs. Due to the fact that the JPEG I-frames alone are stored and the remaining are AMPs instead of frames, the size of the file is comparatively small. The percentage file size saved is calculated using Eq. (15).

$$\begin{aligned}
 \% \text{ file size saved} &= 100 \\
 &- \left[\left(\frac{\text{Resultedfile size (100 frames)}}{\text{File size (100 frames)}} \right) \right. \\
 &\quad \left. \times 100 \right] \tag{15}
 \end{aligned}$$

It can be noted that an average file size of 98% is saved when compared with the original video for a *still camera | still background* motion type. In the case of *still/moving camera | moving background*, the original file size and resulted file size using the proposed method are compared and it is observed that 97.94% of file size on an average is saved. 95.85% of file size is saved for *Moving camera | Moving background* motion type due to the existence of more number of I-frames. 44% of the file size is being saved for B-spline-based affine motion estimation. The conventional performance metric in any compression standard is the *compression ratio* denoted as in Eq. (16). This parameter was also evaluated during the course of our investigations. The outcomes are tabulated in Table 4.

$$\text{Compression ratio} = \frac{\text{File size (100 frames)}}{\text{Resulted file size (100 frames)}} \tag{16}$$

The compression ratio for *Moving camera | Moving background motion type* is observed to be less compared to *still camera | still background* and *still camera | moving background* due to presence of more number of I-frames and dynamisms involved in the video sequence. Hence, it is obvious that the proposed methodology offers good compression ratio and proves it's superiority toward reduction of file size.

Table 2 Adaptive Frame Determination

Motion type	Video File	I, B, P frame numbers
Still Camera Still Background	Mother, Container, Akiyo Total frames : 300	I-frame : 1 P-frame : 2 to 300
	Hall Total frames : 330	I-frame : 1 P-frames : 2 to 330
Still Camera Moving Background	News Total frames : 300	I-frame : 1 P-frame : 2-90,92-150, 152-240, 242-300 B-frame : 91,151,241
Moving Camera Moving Background	Foreman Total frames : 300	I-frame : 1 P-frame : 2-172,220-300 B-frame : 173-219
	Coastguard Total frames : 300	I-frame : 1 P-frame : 2-67,76-86,88-300 B-frame : 68-75, 87
HD Sequence	IntoTree Total frames: 500	I-frame : 1 P-frame : 2-13, 36-70, 85, 113-127, 146,187-222, 246-271, 302-356, 358-391, 424-456, 491-500 B-frame : 14-35, 71-84, 86-112, 128-145, 147-186, 223-245, 272-301, 357, 392-423, 457-490
	OldTownCross Total frames: 500	I-frame : 1 P-frame : 2-15, 24-51, 53-77, 92-118, 120- 145, 184-221, 270-294, 300-323, 358-392, 427-429, 452-476, 489-500 B-frame : 16-23, 52,78-91, 119, 146-183, 222-269, 295-299, 324-357, 393-426, 430- 451, 477-488

Table 3 Replacement of B-frames to I-frames or P-frames

Motion type	Video file	B-frames through AFD	B-frames to I-frames	B-frames to P-frames
Still Camera Moving Background	News	3	3	0
Moving Camera Moving Background	Foreman	47	26	21
	Coastguard	9	9	0
HD Sequence	IntoTree	241	40	201
	OldTownCross	217	46	171

Table 4 Comparisons on file size, % file size saved and compression ratio

Motion Type	Video file ¥	Original File Size (in MB)	File size for 100 frames (in KB)	Resulted file size for 100 Frames (in KB)		% file size saved		Compression Ratio	
				Affine model	B-Spline based affine model	Affine model	B-Spline based affine model	Affine model	B-Spline based affine model
still camera still background	Mother	1.03	355	7.5	110	97.9%	30.9%	47.3:1	3.2:1
	Container	1.28	440	9.4	222	97.9%	50.4%	46.8:1	2:1
	Akiyo	1.00	342	8.5	170	97.5%	49.7%	40.2:1	2:1
	Hall	1.42	443	8.4	218	98.1%	49.2%	52.7:1	2:1
still camera Moving background	News	1.35	461	8	232	98.3%	50.3%	57.6:1	2:1
Moving camera Moving background	Foreman	1.37	471	25	248	94.6%	52.6%	18.8:1	1.9:1
	Coastguard	1.52	520	15	329	97.1%	63.2%	34.6:1	1.5:1
HD sequence	Oldtowncross	1300	530000	7192	128900	98.6%	24.3%	73.6:1	4:1
	Intotree	1300	530000	6289	133521	99%	25.2%	84.2:1	4:1

5.4 Observations and Comparative Analysis on Accuracy Measures

The two vital performance indicators, PSNR (peak-signal-to-noise ratio) and SSIM (Structural Similarity index), have been utilized as metrics to check the video quality. The proposed work is compared with several other renowned works with reference to benchmark videos mentioned in Table 1.

5.4.1 Comparison in Terms of Peak-Signal-to-Noise Ratio (PSNR) with QCIF and HD Video Sequence

The visual quality of the reconstructed image is subjective. Therefore, it is essential to establish an empirical measure to compare the results of various compression standards on image quality. The performance metric under investigation is peak-signal-to-noise ratio (PSNR). The PSNR values are compared with the existing video compression standards and are tabulated in Table 5.

The average PSNR values taken into consideration here were those values resulted from Po et al. [18]; Oh et al. [19]; Liu et al. [20]; Igarta [17]; Kumar et al. [21]; Pandian et al. [22]; Al-Najdawi et al. [23]; Kordasiewicz et al. [24]; Ahmadi et al. [25]; Dimou et al. [26] using a maximum of 150 sequences in a video sequence for processing. In all these methodologies, the video encoding process adopts similar

methods of motion prediction, transform coding, quantization and entropy coding.

It is observed from the investigation that the proposed affine model with LBFGS optimizer yields superior results in all the video sequences except *Coastguard* video. This is due to the fact that many local dynamics visualized in the *Coastguard* video sequence might have affected the PSNR value. Therefore, it is difficult in reconstructing optimal image frames using the proposed method for this video. This is the case for almost all the compression standards too. The issue can be tackled by performing a local piecewise affine transformation in the proposed method sacrificing computational time. From the experimental results, it is observed that the proposed technique with LBFGS optimizer improves the average PSNR gain by 3.56 dB over the existing methods in comparison with various video sequences with different motion types. The affine B-spline-based motion estimation with LBFGS optimizer yield superior results than the affine model with LBFGS optimizer. An average PSNR gain of 6.84 dB is observed over the existing methods which prove it’s superiority over other traditional methods and the proposed affine model with LBFGS optimizer method.

Conventionally, B-frames provide good coding efficiency due to bidirectional prediction. During the proposed NAT process, B-frames undergo a process of conversion to either I- or P-frames based on the similarity factor. This process has been introduced to reduce the cost complexity and

Table 5 Comparison of the objective image quality of reconstructed image frames using average PSNR with various other state-of-the-art methodologies

Standards	Methods	Mother	Container	Akiyo	News	Foreman	Coastguard
H.264	(LM Po et al. 2009)	-	36.5	40	38.35	36.69	34.45
	(K J Oh et al. 2005)	36.6	36.46	38.73	37.06	37.83	-
	(K J Oh et al. 2005) (Lagrangian multiplier)	37.29	36.62	38.87	35.87	36.55	-
	(Liu J et al. 2013)	-	35.43	-	-	35.13	-
	(Michael Igarta 2004)	40.26	33.42	41.66	40.36	36.15	29.24
	(N Al- Najdawi et al. 2014) - Full Search	39.28	-	35.01	37.62	34.66	45.92
	(N Al- Najdawi et al. 2014) - Hierarchical Search	35.29	-	34.89	37.64	33.91	45.56
	(Immanuel Alex Pandian S et al. 2013)	-	33.43	36.36	-	29.74	-
	With AFD	40.3	36.01	41.72	40.67	37	30.09
MPEG-2	(Michael Igarta 2004)	35.19	33.42	33.58	33.4	33.61	27.38
	(Liu J et al. 2013)	-	30.86	-	-	27.04	-
	(Kumar V et al. 2013)	-	35.89	38.71	36.35	34.79	32.21
	With AFD	37.12	35.92	39.08	37.16	34.72	30.03
Affine Motion prediction	(Kordasiewicz R C et al. 2007)	-	-	-	-	33.02	-
	(Ahmadi A. et al 2011)	-	-	41.01	-	25.09	26.59
Proposed affine Model		41.93	38.66	44.26	41.32	33.64	30.01
Affine model with Optimizers	Non-linear LSQ	42.01	39.12	45.07	41.54	34.42	31.12
	BFGS	42.52	39.49	45.21	42.54	34.96	31.78
	LBFGS	43.74	40.28	45.99	42.91	35.79	32.12
B-Spline based affine model (average PSNR gain in dB)		48.6 (+8.9)	46.24 (+9.742)	49.87 (+9.61)	47.82 (+8.69)	37.12 (+3.11)	33.97 (+1.02)

delay, without compromising the reconstruction quality. For a moving background motion type, most of the B-frames are converted to I-frames. This is due to the fact that many new objects are introduced in consecutive frames as the sequence is fast moving. As seen in Table 5, the video quality of the proposed method outperforms other conventional methods (which uses the B-frame). Hence, the quality of the reconstructed frame does not suffer since the number

of I-frames is directly proportional to the quality of the reconstructed frame. The methodologies were also investigated for benchmark HD video sequences mentioned in Sect. 5. The outcomes of this investigation are tabulated in Table 6.

The proposed affine with LBFGS optimizer outperforms other techniques with a PSNR gain of 1.49 dB for HD video sequences. Hence, the integration of AFD and the NAT

Table 6 Comparison of the objective image quality of HD reconstructed image frames with various other state-of-the-art methodologies

Standards	Methods	OldTowncross	IntoTree
		PSNR(dB)	
<i>H.264</i>	<i>(Dimou A et al. 2007)</i>	33.085	32.27
<i>MPEG-2</i>		33.65	33.49
Proposed Affine Model		34.12	34.01
<i>Affine model with Optimizers</i>	<i>Non-linear LSQ</i>	34.71	34.93
	<i>BFGS</i>	34.92	35.01
	<i>LBFGS</i> <i>(average PSNR gain in dB)</i>	35.40	35.63
B-Spline based affine model (average PSNR gain in dB)		36.62 (+2.1)	36.01 (+1.72)

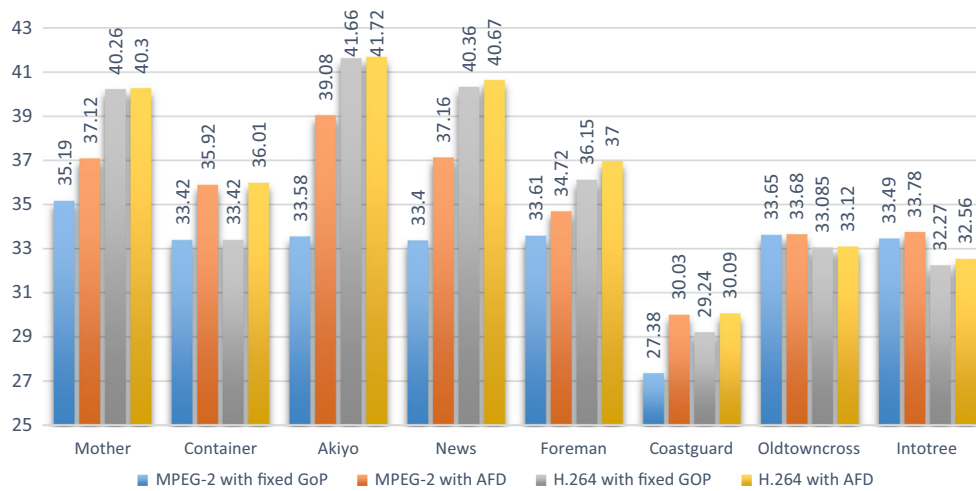


Fig. 8 Comparison of video reconstruction quality between methods using fixed GoP and AFD

process in affine model proves to improve the coding performance significantly. The proposed B-spline-based affine model with LBFGS optimizer yields superior results than the affine model with LBFGS optimizer with an average PSNR gain of 1.91 dB.

Investigation using AFD-based and fixed GoP-based conventional techniques The outcomes were further assessed and tested with conventional techniques like H.264 and MPEG-2. AFD has been employed in standard techniques, and these approaches have been compared with fixed GOP “IBBPBBPBBPBI” conventional methods. Figure 8 depicts the examined outcomes in terms of video reconstruction quality, where the PSNR (in dB) is conspired for many investigations with diverse video sequences.

It is obvious that the methodologies incorporating AFD offer good PSNR and are superior to other fixed GoP techniques.

5.4.2 Comparison in Terms of Structural Similarity Index (SSIM)

To validate the experimentation, SSIM index has also been employed for evaluating the similarity of the reconstructed sequence with the original sequence. This similarity index involves three components such as luminance similarity, contrast similarity and structural similarity. The SSIM values for the proposed method along with optimizers are compared with the average SSIM of the existing video compression standards observed from Igarta [17] as shown in Table 7. The proposed affine model resulted with 0.94 average SSIM index. The same proposed affine model with LBFGS optimizer has resulted with 0.979 SSIM index on an average. Affine B-spline-based motion estimation with LBFGS optimizer resulted with 0.99 SSIM index on an average proving its superiority.

Table 7 Comparison of average SSIM of the proposed method over other methods

Video file	MPEG-2	H.264	Proposed affine model	Affine model with optimizers			B-spline-based affine model
				Nonlinear LSQ	BFGS	LBFGS	
Mother	0.886	0.95	0.982	0.99	0.99	0.99	1
Container	0.892	0.95	0.98	0.99	0.99	0.99	1
Akiyo	0.876	0.96	0.99	0.99	0.99	0.992	1
News	0.9	0.96	0.988	0.99	0.99	0.99	1
Foreman	0.884	0.946	0.946	0.96	0.97	0.975	0.982
Coastguard	0.866	0.91	0.876	0.93	0.94	0.95	0.975
OldTowncross	0.91	0.9	0.89	0.93	0.94	0.949	0.98
IntoTree	0.93	0.92	0.921	0.95	0.96	0.97	0.99
Average SSIM index	0.895	0.937	0.94	0.966	0.97	0.979	0.99

Table 8 Computational time analysis

Video file	Computation Time (MPEG-2) (In Seconds)	Computational Time (proposed Approach) (In seconds)						
		AFD	NAT	Affine compression	Warping	Total Time	B-Spline based affine model (Total)	
QCIF	Mother	385.01	38.09	49.5	65	60.5	213.09	320
	Container	372.36	22.10	40	52	46	160.17	234
	Akiyo	371.86	25.1	30.5	41.5	39.5	136.60	219
	News	373.35	23	29.5	40	35	127.53	200
	Foreman	370.63	22	28.4	36	31.5	117.96	179
	Coastguard	373.88	30.05	38	45	40	153.08	220
HD	Oldtowncross	1020	148	165	210	197	720	918
	Intotree	1085	169	186	212	193	760	956

5.5 Comparison in Terms of Computational Time

Computational time is yet another vital performance metric in video compression. The significance of this metric evaluation is portrayed in this section. The computational time has been observed to be less in the proposed method when compared with MPEG-2 code which was implemented on the same platform. The time taken for implementing the proposed method without applying optimization technique is shown in Table 8.

The computational efficiency of the proposed method with respect to MPEG-2 standard video coding shows an average of 45.2% improvement for the proposed affine model and 25.4% for B-spline-based affine model.

Xu et al. (2016) [27] has developed a two-dimensional reversible method for intra-frame in H.264. It is widely used for many applications related to entertainment including broadcast, cable, satellite, DVD, telecom, streaming services. It supports movies encoded with H.264. It provides

enhanced compression performance. Mukherjee et al. (2016) [28] has engaged video compression algorithms for HDR (High dynamic range videos). Further, the proposed algorithm can be employed for HDR videos in future.

6 Conclusion

An adaptive frame determination methodology is proposed for determining I-, B- and P-frames of a video sequence. Furthermore, NSEW affine translation is proposed for the purpose of increasing the computational efficiency by avoiding B-frames in computations. In order to effectively compress a video, a hybrid approach incorporating global and local transformation parameters from affine and B-Spline transformation models was proposed for compression and decompression. In order to improve the quality of the reconstructed image, optimizers like BFGS, LBFGS and nonlinear LSQ are utilized for achieving optimal affine motion param-

eters. A comparative analysis is made for standard video sequences of different motion types, and the results are documented in this work. The results indicate that significant amount of enhancement in terms of average PSNR and SSIM have been achieved in all cases. It is also observed that the proposed approaches achieve a significant reduction in computational time without sacrificing the quality of the video. Moreover, reduced file size yields good compression ratio, proving the superiority of the proposed methodology. Further, the approach can also be applied in the field of Medicine by utilizing local transformations for the purpose of compression. A robust maximum likelihood approach can also be utilized to enhance the quality of medical images.

Acknowledgements Authors would like to express their sincere gratitude toward Karunya University for the research facilities and infrastructure that has been provided, without which this work would not have been a reality. The authors also express their heartfelt gratitude and thank all the reviewers for their valuable comments in enhancing the article.

References

1. Wang, Y.L.; Wang, J.X.; Lai, Y.W., Su, A.W.Y.: Dynamic GOP structure determination for real-time MPEG-4 advanced simple profile video encoder. In: Proceedings of IEEE International conference on Multimedia and Expo, Amsterdam, pp. 293–296 (2005)
2. Paul, M.; Lin, W.; Lau, C.T.; Lee, B.S.: Video coding with dynamic background. *EURASIP J. Adv. Signal Process.* **11**, 1–17 (2013)
3. Ohm, J.R.; Sullivan, G.J.; Schwarz, H.; Tan, T.K.; Wiegand, T.: Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.* **22**, 1669–1684 (2012)
4. Mathew, M: Overview of temporal scalability with scalable video coding (SVC). Texas Instruments Application report, pp. 1–7 (2010)
5. Paul, M.; Lin, W.; Lau, C.T.; Lee, B.S.: A long term reference frame for hierarchical B-picture based video coding. *IEEE Trans. Circuits Syst. Video Technol.* (2014). doi:10.1109/TCSVT.2014.2302555
6. Tabatabai, A.J.; Jasinski, R.S.; Naveen, T.: Motion estimation methods for video compression—a review. *J. Frankl. Inst.* **335**(8), 1411–1441 (1998)
7. Lu, Y.; Li, Z.N.: Automatic object extraction and reconstruction in active video. *Pattern Recognit.* **41**, 1159–1172 (2008)
8. Wiegand, T.; Steinbach, E.; Girod, B.: Affine multipicture motion-compensated prediction. *IEEE Trans. Circuits Syst. Video Technol.* **15**, 197–209 (2005)
9. Theoharis, T.; Papaioannou, G.; Platis, N.; Patrikalakis, N.M.: *Graphics and Visualization: Principles and Algorithms*. A. K. Peters/CRC Press, Taylor & Francis Group, Wellesley (2008)
10. Alavala, C.R.: *CAD/CAM Concepts and Applications*. PHI Learning Private limited, New Delhi (2009)
11. Ding, J.R.; Yang, J.F.: Adaptive group-of-pictures and scene change detection methods based on existing H.264 advanced video coding information. *IET Image Process.* **2**(2), 85–94 (2008)
12. Test videos. <http://media.xiph.org/video/derf/> (2014). Accessed 16 Aug 2014
13. Gu, S.; Xin, M.; Sciurba, F.C.; Wang, C.; Kaminski, N.; Pu, J.: Bi-directional elastic image registration using B-spline affine transformation. *Comput. Med. Imaging Graph.* **48**(4), 306–314 (2014)
14. Szeliski, R.; Coughlan, J.: Spline-based image registration. *Int. J. Comput. Vis.* **22**(3), 199–218 (1997)
15. Gholipour, A.; Kehtarnavaz, N.; Briggs, R.; Devous, M.; Gopinath, K.: Brain functional localization: a survey of image registration techniques. *IEEE Trans. Med. Imaging* **26**(4), 427–451 (2007)
16. Xiao, Y.; Wei, Z.; Wang, Z.: A limited memory BFGS type-method for large-scale unconstrained optimization. *Comput. Math. Appl.* **56**, 1001–1009 (2008)
17. Igarta, M.: A study of MPEG-2 and H.264 video coding. Thesis submitted for Master of Science in Electrical and Computer Engineering, Purdue University, US (2004)
18. Po, L.M.; Ng, K.H.; Cheung, K.W.W.; Wong, K.M.; Uddin, Y.M.S.; Ting, C.W.: Novel directional gradient descent searches for fast block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **19**(8), 1189–1195 (2009)
19. Oh, K.J.; Ho, Y.S.: Adaptive rate-distortion optimization for H.264. In: *PCM 2005, Part II. LNCS*, vol. 3768, pp. 617–628. Springer, Berlin (2005)
20. Liu, J.; Qiao, F.; Wei, Q.; Yang, H.: A novel video compression method based on underdetermined blind source separation. In: *Multimedia and Ubiquitous Engineering. Lecture Notes in Electrical Engineering*, vol. 240, pp. 13–20 (2013)
21. Kumar, V.; Sharma, K.G.; Jalal, A.S.: Macro-block mode decision in MPEG-2 video compression using machine learning. In: *Lecture Notes in Electrical Engineering*, pp. 149–158 (2013)
22. Pandian, S.I.A.; Bala, G.J.; Anitha, J.: A pattern based PSO approach for block matching in motion estimation. *Eng. Appl. Artif. Intell.* **26**, 1811–1817 (2013)
23. Al-Najdawi, N.; Al-Najdawi, M.N.; Tedmori, S.: Employing a novel cross diamond search in a modified hierarchical search motion estimation algorithm for video compression. *Inf. Sci.* **268**, 425–435 (2014)
24. Kordasiewicz, R.C.; Gallant, M.D.; Shirani, S.: Affine motion prediction based on translational motion vectors. *IEEE Trans. Circuits Syst. Video Technol.* **17**(10), 1388–1394 (2007)
25. Ahmadi, A.; Pouladi, F.; Salehinejad, H.; Talebi, S.: Fast two-stage global motion estimation: a blocks and pixels sampling approach. *Intell. Interact. Multimed. Syst. Serv.* **11**, 143–151 (2011)
26. Dimou, A.; van der Vleuten, R.J.; de Haan, G.: Picture-quality optimization for the high definition TV broadcast chain. Technical Note PR-TN 2007/00338- Koninklijke Philips Electronics Nv 2007 (2007)
27. Xu, D.; Wang, R.: Two-dimensional reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. *Signal Process. Image Commun.* **47**, 289–302 (2016)
28. Mukherjee, R.; Debattista, K.; Bashford-Rogers, T.; Vangorp, P.; Mantiuk, R.; Bessa, M.; Waterfield, B.; Chalmers, A.: Objective and subjective evaluation of High dynamic range video compression. *Signal Process. Image Commun.* **47**, 426–437 (2016)

