



True-Motion Estimation Algorithm and Its Application to Motion-Compensated Temporal Frame Interpolation

Shashi Kiran.S¹, Srinivas Babu.N², Ruckmani Divakaran³, Mohmad Ashiq.A⁴, Arun Kumar.B⁵, Rohit.J⁶

Assistant Professor, Dr. TTIT, K.G.F^{1,2}

HOD of ECE, Dr. TTIT, K.G.F³

Students, Dr. TTIT, K.G.F^{4,5,6}

Abstract: Motion estimation (ME) has a vital role in video coding and several video processing applications, such as denoising, de-interlacing, and frame rate up-conversion (FRUC) or frame interpolation. ME is employed to exploit the temporal correlation between video frames either to reduce the temporal redundancy for video coding applications or to improve the visual video quality for video processing applications. One might argue that some of these video processing applications may potentially utilize the existing motion vectors (MVs) from the decoder via MV post-processing to keep the complexity low; however, this may not usually be a feasible option. This infeasibility could be due to either difficulty of using MVs or lack of available MVs. As video coding and video processing applications are often implemented separate intellectual properties (IPs) in hardware[12], it may be very difficult to share the MVs between decoder and other video processing applications due to bandwidth, latency, storage, and design specification reasons. Besides, some of these video processing applications may be employed either before the encoding or after the decoding, and some of them may be employed at both places; if it is employed before the encoding then MVs are not available, as a result ME needs to be performed. For example, FRUC is employed only at the display side after the decoder; de-interlacing and de-noising, however, can be utilized in both places. Where as in true motion estimation the mainly it goes to detect the motion object as closely as possible by using the block matching algorithm, and then after the estimation of the true motion vector fields it helps to produce the motion compensated temporal frame interpolation. This methods is gives the more video quality and the smoothness with the flow of frames. The main aim of this paper is to determine the motion (moving) object in the video sequences this method is called as true motion estimation by adopting the implicit and explicit smoothness constraint on block matching algorithm. After finding true motion vector also called as coherent motion vector field is used to produce the good temporal interpolated frames between existing frames this gives good video with easily flowing one after the other by smoothly and continuously. After getting the interpolated frames the performance metrics like PSNR (peak signal to noise ratio) and SSIM (structural similarity) between the interpolated frames and the original frames.

Keywords: PSNR (peak signal to noise ratio) and SSIM (structural similarity), FRUC, motion vectors (MVs).

I. INTRODUCTION

Motion estimation (ME) has a vital role in video coding and several video processing applications, such as denoising, de-interlacing, and frame rate up-conversion (FRUC) or frame interpolation. ME is employed to exploit the temporal correlation between video frames either to reduce the temporal redundancy for video coding applications or to improve the visual video quality for video processing applications. One might argue that some of these video processing applications may potentially utilize the existing motion vectors (MVs) from the decoder via MV post-processing to keep the complexity low; however, this may not usually be a feasible option. This infeasibility could be due to either difficulty of using MVs or lack of available MVs. As video coding and video processing applications are often implemented separate intellectual properties (IPs) in hardware[12], it may be very difficult to share the MVs between decoder and other video processing applications due to bandwidth, latency, storage, and design specification reasons. Besides, some of these video processing applications may be employed either before the encoding or after the decoding, and some of them may be employed at both places; if it is employed before the encoding then MVs are not available, as a result ME needs to be performed. For example, FRUC is employed only at the display side after the decoder; de-interlacing and de-noising, however, can be utilized in both places. Where as in true motion estimation the mainly it goes to detect the motion object as closely as possible by using the block matching algorithm, and then after the estimation of the true motion vector fields it helps to produce the motion compensated temporal frame interpolation. This methods is gives the more video quality and the smoothness with the flow of frame.



II. EXISTING SYSTEM LIMITATIONS

The achievable visual quality using MV post processing may not be as good as the quality that can be attained using true MVs since the received MVs are generally generated at the encoder by employing ME to minimize the prediction errors instead of to track the projected object motion. Better image and video quality is obtained when the MVs that track the projected object motion are used in video processing applications, such as denoising, de-interlacing, and FRUC. For example, to increase the frame rate MV post processing may be preferred for resource-limited handheld devices, whereas ME algorithm can be employed for high visual video quality demanding applications, such as broadcasting, television and multimedia players[1],[2]. Besides, human visual system (HVS) tolerates artifacts better in small displays compared with large displays due to the increased angular resolution; hence, handheld devices can tolerate lower complexity inferior methods compared with large displays.

It is not easy to accurately estimate motion since ME is an ill-posed problem [12,13]. Furthermore, to distinguish ME schemes used in video processing applications from regular ME techniques used in video coding, they are usually termed as true-motion estimation (TME) algorithms to emphasize the fact that their objective is to track the projected object motion rather than to reduce the temporal redundancy. Projected object motion usually results in a coherent motion vector field (MVF) except on the motion boundaries; however, regular ME algorithms may not necessarily produce a coherent MVF since their objective is to minimize the prediction error or number of bits required to code the prediction error. Hence, regular ME algorithms are susceptible to give wrong motion trajectories. To obtain a coherent MVF, TME algorithms further impose smoothness constraints on MVs by using their spatiotemporally neighboring blocks.

III. PROPOSED TME (TRUE MOTION ESTIMATION)

The proposed TME algorithm uses two consecutive frames and the previous MVF to estimate the current MVF. Similar to other practical video applications, it employs BMA to obtain a straightforward, efficient, and hardware-friendly implementation. To alleviate the computational-complexity of full search (FS), BMAs usually employ FME algorithms. FME algorithms generally utilize one or a combination of the following categories: reduction of search points (SPs), simplification of matching criterion, multi-resolution search, predictive search, and fast FS[9],[19]. Out of these categories, predictive search and multi-resolution search inherently impose smoothness constraints on MVF. Since multi-resolution search requires additional storage compared with predictive search we prefer predictive-search in the proposed scheme.

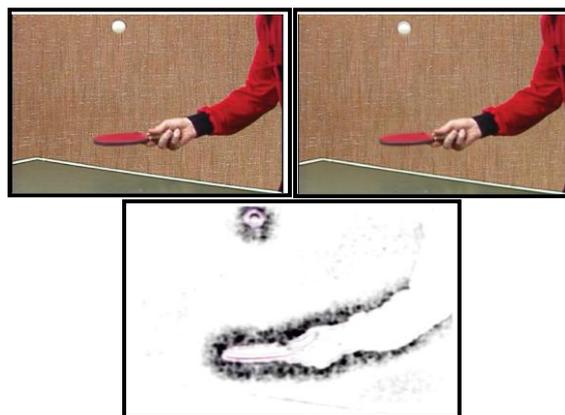


Fig3: Motion detection

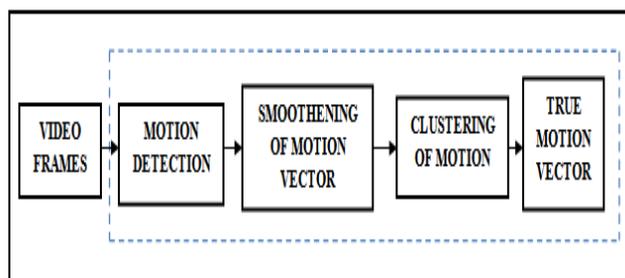


Fig1:Encoder



There are plethora of algorithms for FME, we choose Clustering algorithm due to its performance and ability to provide sub-sample accuracy at low computational-complexity. [19] Clustering algorithm along with predictive search provides an effective FME that implicitly impose smoothness constraint on MVF. In the following subsections, how explicit smoothness constraint is enforced, selection of the predictor MVs to impose implicit smoothness constraint, and the proposed true-motion estimation making use of both explicit and implicit smoothness constraints on block matching algorithm.

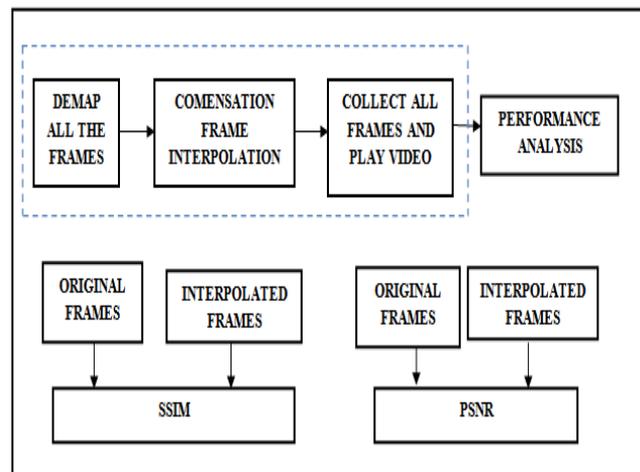


Fig2: Decoder

Video: Video is the continuous moving frames, collect all the frames from video. For the calculations of the motion detection.

Motion detection: Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. with the help of the eight neighbors search method and produces the motion vectors .

Smoothing of motion vector: In statistics and image processing, to smooth a data set is to create an approximating function that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures/rapid phenomena. In smoothing, the data points of a signal are modified so individual points (presumably because of noise) are reduced, and points that are lower than the adjacent points are increased leading to a smoother signal. Smoothing may be used in two important ways that can aid in data analysis the aim of smoothing is to give a general idea of relatively slow changes of value with little attention paid to the close matching of data values, while curve fitting concentrates on achieving as close a match as possible.

Clustering of motion vector: Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). To cluster the MVs in the spatiotemporal neighborhood, it is desired to have a clustering algorithm that would adaptively give different number of clusters based on the maximum cluster distance specified at input, along with the additional constraint that this will take a fixed number of iterations.

True Motion Vectors: After clustering of motion vector the true motion is obtained the motion vector field which is mainly tracked object motion with the back ground is same as image and moving object is in white blocks. After the encoding of the data with the predicted value and the other information in decoder side the original video frames are produced by assembling the all blocks.

DEMAP All Frames: After encoding the predicted value of true motion vector and other blocks are get together for producing the original video frames. in decoder side.

Interpolation: Interpolation is a method of constructing new data points within the range of discrete set of known data points, Obtained dense motion fields at the interpolation instant are used to generate the interpolated frame. The additional extra frames are produces in b/w the existing frames to produce the better video quality of frames.

IV. METHODOLOGY

TME algorithms can be broadly classified into pixel-domain and transform-domain algorithms

Pixel-domain: Block Matching Algorithm: A Block Matching Algorithm (BMA) is a way of locating matching blocks in a sequence of digital video frames for the purposes of motion estimation. Selected block matching algorithm because



it is very easy to estimate the motion by converting the frames into the different block. Then apply process on those block.

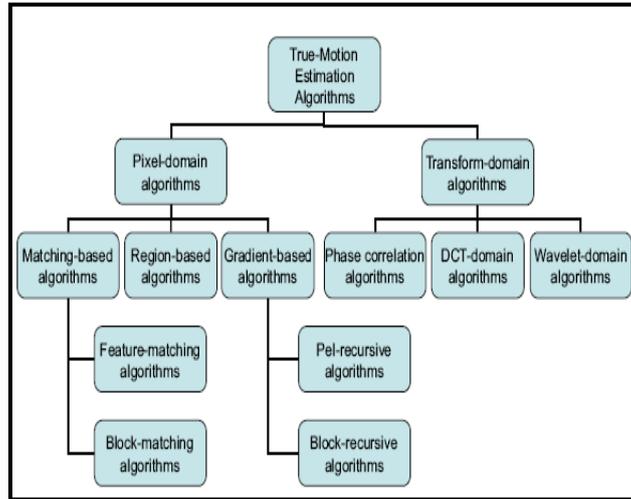


Fig 4:Classification of motion estimation algorithms

A. BLOCK MATCHING ALGORITHM

The purpose of a block matching algorithm is to find a matching block from a frame i in some other frame j, which may appear before or after i. This can be used to discover temporal redundancy in the video sequence, increasing the effectiveness of inter frame video compression and television standards conversion. Block matching algorithms make use of an evaluation metric to determine whether a given block in frame j matches the search block in frame i. Many researchers employed TME in various video applications, such as deinterlacing, denoising, FRUC, and video coding, where BMA is employed for more efficient and hardware-friendly implementation. Some researchers employed other TME schemes as well, such as gradient-based, object-based, and transform-domain algorithms. Among the TME methods, BMA is the most widespread algorithm used in video coding and video processing applications partly due to its straightforward, simpler, more efficient, and hardware-friendly implementation.

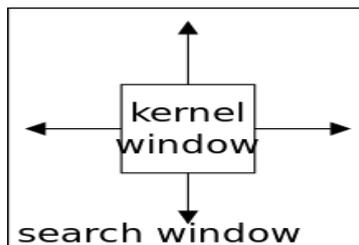


Fig 5: Block matching concept

BMAs target to minimize an objective function, which is also called block distortion measure (BDM), in the form

$$E_D(d) = \sum_{n \in B} \rho(I_{k_1}[n+d], I_{k_2}[n])$$

where I_{k_1} and I_{k_2} are images at times k_1 and k_2 , d is the displacement of n or MV at n , B is the set of all pixels in I_{k_2} , and $\rho(\cdot, \cdot)$ is the matching criterion. Usually, $\rho(x, y) = |x - y|^p$ is used by setting $p = 1$ for implementation simplicity, which is usually referred to as sum of absolute differences or errors (SAD or SAE). To achieve smoother MVFs, several different methods have been attempted in the literature. Imposing smoothness constraint can be done either explicitly or implicitly, or both. Approaches impose implicit smoothness constraint through the use of multi-resolution or predictive search for BMA. Enforcing explicit smoothness constraint is generally achieved by one of the following three ways: 1) adding a penalty term to (1), 2) using a different matching criterion, $\rho(\cdot, \cdot)$, and 3) post-processing the MVF obtained from (1).

Obtaining a robust TME algorithm for real-time video processing applications is a very challenging task. The algorithm should not only produce a smooth MVF but also satisfy some additional constraints for hardware implementation, such as algorithm regularity, low computational-complexity, and low memory bandwidth. Most of the reduction of the computational-complexity is usually achieved by decreasing the number of SAD calculations of the BMA through



various fast ME strategies, such as reduction of search point, predictive search, and multi resolution search. Similarly, memory bandwidth is affected by the block size, search range used in the algorithm, and the block processing scan pattern.

B.MOTION ESTIMATION

This is the very important one part of my project to find out the moving object motion in the video sequences. In case of motion estimation the first of all we need to collect the all frames in the video. After collecting all the frames it is necessary to convert the all frames which is in the coloured format into the gray scale images. First the reference frames is to be taken from first frames of video sequences then the motion is estimated by using all other frames by using the following steps

1. Video is collection of the frames so first I need to collect the all frames from video.
2. After collecting the all frames .then going to convert the coloured image into the gray scale image.
3. The first frames is taken as the reference frame and all other frames are considered as the anchor frames.
4. In next convert the both reference and anchor frames into the 4*4 block size.
5. Then the converted 4*4 block is now need to perform the Euclidian norm which gives the difference value which is matched the block or not.
6. The correlated output which is the value of the block after the Euclidian norm operation.
7. The correlated value is more than that of threshold valued then that block has some motion .then that block is need to show as whitened and this called motion block then the remaining block is produced as like only.
8. Totally the above process is going to repeat for all blocks and all other frames.

Then after the find out the motion block collect the frames and produce the video which is gives the where the motion is happening in the sequence wise. Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel. The motion vectors may be represented by a translational model or many other models that can approximate the motion of a real video camera, such as rotation and translation in all three dimensions and zoom.

Imposing smoothness

Using a modified matching function is usually motivated by intuition, whereas adding a penalty term can be theoretically justified by using a Bayesian MAP estimator[4],[11], which imposes certain prior distribution on the model parameters. The a posteriori probability distribution of the motion field

$$P(D_K = d_K | I_K = I_K; I_{K-1}) \dots \dots \dots (5.1)$$

Equation (4.1) is used to obtain the MAP estimate by rewriting its Bayes equivalent as

$$P(I_K = I_K | D_K = d_K; I_{K-1}) \cdot P(D_K = d_K; I_{K-1}) \dots \dots \dots (5.3)$$

where D_k is a vector random field, d_k is one of its realization, I_k is a scalar random field, and I_k is one of its realization.

Then, the MAP estimate of d_k is computed as follows:

$$d_k^{\wedge} = \arg_d \max (p(I_K = I_K | D_K = d_K; I_{K-1}) \cdot p(D_K = d_K; I_{K-1}))$$

Where the first term is related to the observation model measuring how well d_k models the change, and the second term serves as a motion model explaining the prior information contribution of the random field D_k , such as its smoothness.

To solve (5.3), it is assumed that the displaced frame difference (DFD) is a zero-mean Gaussian distribution; hence, the first term can be written as a product of zero-mean Gaussians. In addition, it is assumed that D_k is a MRF; so, the second term is a Gibbs distribution specified by cliques and a potential function. Using these assumptions and two-element cliques for MRF, (4.3) can be recast as

$$d_k^{\wedge} = \arg_d \min (\sum_{l \in N_n} |I_K[n] - I_{K-1}[n+d]|^2 + \lambda \sum ||$$

$$d[n] - d[l] ||^2)$$

Where N_n is a set of neighbors of n (e.g, 4- or 8-neighbors), and $||\bullet||$ denotes Euclidian norm. λ is equal to $2\sigma^2$, where σ^2 is the variance of the DFD; λ controls the amount of smoothness. The first term measure's how well each MV candidate d matches by comparing the intensity variation, and the second term penalizes the deviation from its neighbors. For minimization of this function different approaches are proposed, such as simulated annealing, iterated conditional modes, highest confidence first. However, they are very complex, and incompatible for real-time applications. In addition, even though this minimization gives rise to smooth MVF, it is inconvenient since the MVF is smooth even at the object boundaries.



C. CLUSTERING OF MOTION BLOCK

In this step the grouping of motion blocks which is similar in nature, which is also gives the perfect motioned object. To cluster the MVs blocks in the spatiotemporal neighborhood, it is desired to have a clustering algorithm that would adaptively give different number of clusters based on the maximum cluster distance specified at input, along with the additional constraint that this will take a fixed number of iterations. The algorithm takes the sample set X , maximum number of clusters K , and the maximum cluster distance D as input; after execution it outputs the cluster means m and cluster labels c . Since the sample set X is formed by zero MV and MVs of the spatiotemporal neighbors, zero MV is used in the initialization. Mean of the first cluster is set to zero and others to undefined. Similarly, label of the first sample is set to zero and others to undefined. Initially, number of formed clusters is set to one; depending on the displacement of MVs, the number will adaptively increase until K .

Elements in the set, starting from second till last, are tested for the closest cluster and then the maximum distance to the elements of this cluster are calculated; if they are in the D proximity of a cluster, then they become a member of this cluster, otherwise they start a new cluster. Finally, cluster mean of the assigned cluster is updated and continued with the next sample in the set [9]. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). This gives the motion block grouping and also it specifies the true motion of blocks with their indices.

Motion-compensated frame interpolation

This is the method of producing the extra frames in between the existing frames for the purpose of better video quality by increasing the frames rate depends on the motion. To enable better quality multiple interpolation frames between two existing frames UME is employed instead of BME; to better handle occlusion regions effectively for better quality both forward and backward TME are used. Occluded areas exist in one of the two successive frames; depending on these areas being covering or uncovering corresponding former or latter frame has the necessary information, and the interpolated frame should be constructed accordingly. In addition, to prevent blocking artifacts in the interpolated frame, dense motion field at the interpolation instant will be obtained and used for temporal interpolation.

D. OBTAINING DENSE MOTION FIELD

Applying backward and forward TME between frames I_k and I_{k+1} will result in MVFs d_{k+1}^b and d_k^f , respectively, where superscript denotes the direction. Although temporally shifting them to obtain the MVFs at $k+\alpha$ is more straightforward compared with projecting them, it gives inaccurate MVs especially for large values. Hence, these MVFs will be projected to $k + \alpha$ for better interpolated image quality. Although the use of UME results in a contiguous MVF at its associated anchor frame, the projected MVF at interpolation instant may not be contiguous. In fact, it will have overlap and whole regions due to differing MV values of neighboring blocks.

Interpolation: Interpolation means that adding extra things in between the existing one. In the video processing the interpolation means the adding the extra frames in between the existing frames, for the purpose of smoothening the motion object and to get the good video quality. Obtained dense motion fields at the interpolation instant are used to generate the interpolated frame. Existence of occlusion areas between neighboring frames complicates the interpolation process since they may exist in either one of them. Occlusion areas are either covering or uncovering depending on the former or latter frame has the necessary information. An uncovering area at frame $k + 1$ does not have its correspondence at frame k ; hence, the backward MVs in this area may not be accurate. To accurately find the motion of this occlusion area frame $k + 2$ has to be used as it would have the corresponding area. Similarly, for an covering area at frame k , frame $k - 1$ has to be utilized to find the corresponding MV of this covering area. Therefore, to effectively handle occlusion regions between two frames, one additional frame in each direction is required, resulting in total four frames.

This, however, increases the complexity and storage requirements of the solution. To enable a low complexity solution, we employ only two neighboring frames and gracefully obtain the interpolated images. To handle the occlusion areas using two frames, some methods employ occlusion detection mechanisms and use their detection results to interpolate the occlusion area by selecting the pixels from the corresponding neighboring frame depending on the area is covering or uncovering. However, such hard switching methods at pixel or block level may produce strong local temporal artifacts that are very annoying during the playback. It's usually preferred to have a global degeneration than a strong local distortion. As a result, occlusion areas are dealt with implicitly using both forward and backward MVFs. In the proposed method hard switching is avoided by employing soft decision for occlusion areas. Both forward and backward MVFs are used to generate the interpolated frame by mixing the forward and backward interpolated frames. This results in increasing the frames in video for producing the better quality video by adopting the true motion estimation.

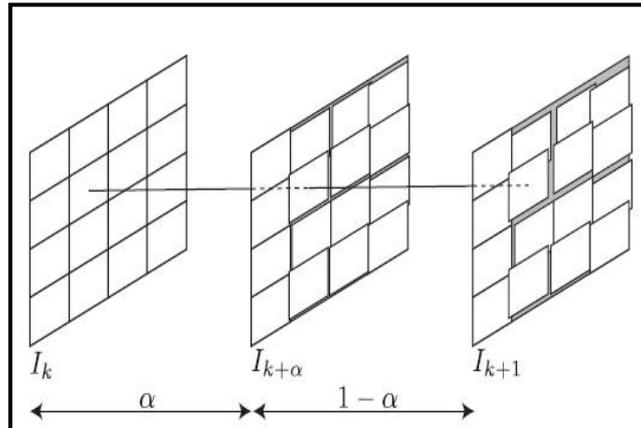


Fig 6: Interpolation of frame by using other two frames.

E.PERFORMANCE METRICS

The distortion introduced into image/video can be classified into three categories in regard to the human visual perception as sub-, near-, and supra-threshold distortions [22], [23]. Sub threshold distortion is below just-noticeable difference (JND) and usually unperceivable by the HVS; near-threshold distortion is slightly above JND; and, supra-threshold distortion generally appears in a structured form, such as blockiness, ghosting, and blurring. While PSNR has good performance on measuring near-threshold distortion, SSIM has good performance on measuring both near- and supra-threshold artifacts [23]. Since supra-threshold dominates the human visual Perception PSNR simply measures the mean squared error; however, SSIM is constructed by incorporating luminance, contrast, and structure information at each pixel locality. Therefore, both PSNR and SSIM values are used. Typical values for the PSNR in between the two images are between 30 and 50 dB. So I tried to get the PSNR in between these values.

F.PSNR (PEAK SIGNAL TO NOISE RATIO)

PSNR simply measures the mean squared error. Typical values for the PSNR in between the two images are between 30 and 50 dB. So I tried to get the PSNR in between these values. PSNR is most commonly used to measure the quality of reconstruction of lossy compression codec's. The signal in this case is the original data, and the noise is the error introduced by compression. PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality. PSNR is most easily defined via the mean squared error (MSE). Given a noise-free $m \times n$ monochrome image I and its noisy approximation K , MSE is defined as

$$m-1 \ n-1$$

$$MSE = 1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

The PSNR is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with B bits per sample, MAX_I is $2^B - 1$. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Alternately, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space. PSNR is calculated and produced for interpolated frames and the original frames.

G.SSIM (STRUCTURAL SIMILARITY)

The Structural Similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared provided the other image is regarded as of



perfect quality This will be compared with the interpolated frames and original frames. To define the image is fully matched with other image means the metric index SSIM should be nearer to 1. And it is not similar means it should be zero. The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size $M \times N$ is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \dots\dots\dots(5.8)$$

with

1. μ_x the average of x ;
2. μ_y the average of y ;
3. σ_x^2 the variance of x ;
4. σ_y^2 the variance of y ;
5. σ_{xy} the covariance of x and y ;
6. $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
7. L the dynamic range of the pixel-values (typically this is $2^{\#bits \text{ per pixel}} - 1$);
8. $k_1 = 0.01$ and $k_2 = 0.03$ by default.

V. SIMULATION RESULTS APPLICATIONS

This chapter gives the implemented results, that is how to find out true motion and its result gives the tracked object motion. And the reconstructed video in decoder side then it gives the interpolated video sequences and frames then lastly the performance metrics like the PSNR, SSIM, and MSE and before interpolation how many number of frames is there and after the interpolation the number of frame present.



Fig 7:(a) frame 1

Fig 7: (b) frame 30



Fig 7: (c) true motion detection.

Above fig. shows that the detected motion object between frame 1 and frame 30. It gives the detected motion is in whitened blocks and the remaining areas are kept as same like both the frames.

The PSNR and SSIM for different video frames are given in fig. 7.4 and 7.5 respectively. And the finally SSIM map for all different frames is given in the fig. 6.6 which shows the how much map interpolated frames with the original frames. Totally I conclude by using the tracked object motion motion compensated temporal frame interpolation frames is produce and the performance metric is also calculated and shown it in as graph.

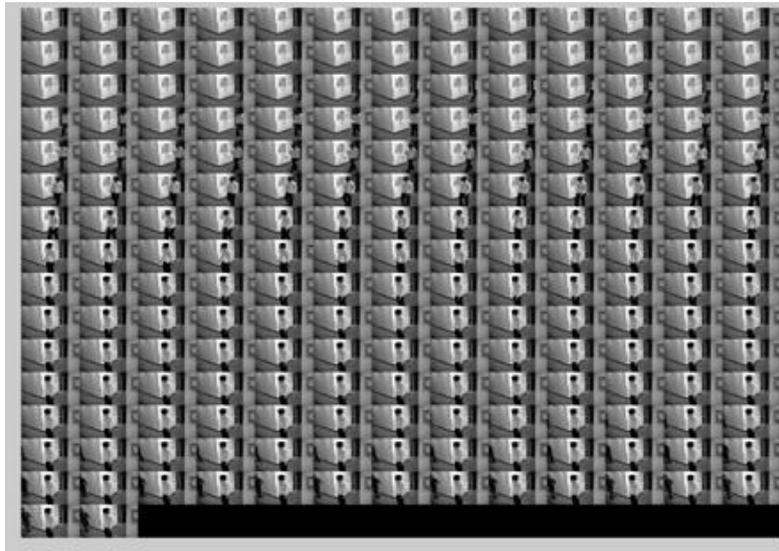


Fig 8: Total number of the frames after interpolation.

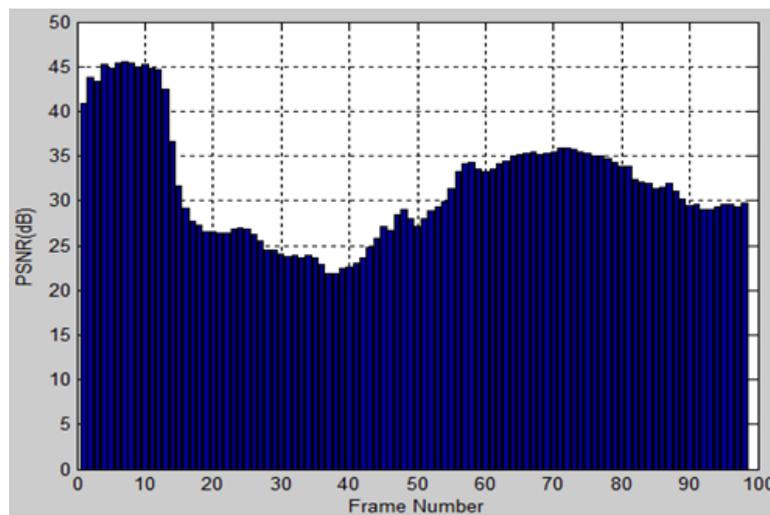


Fig 9: PSNR for all different frames.

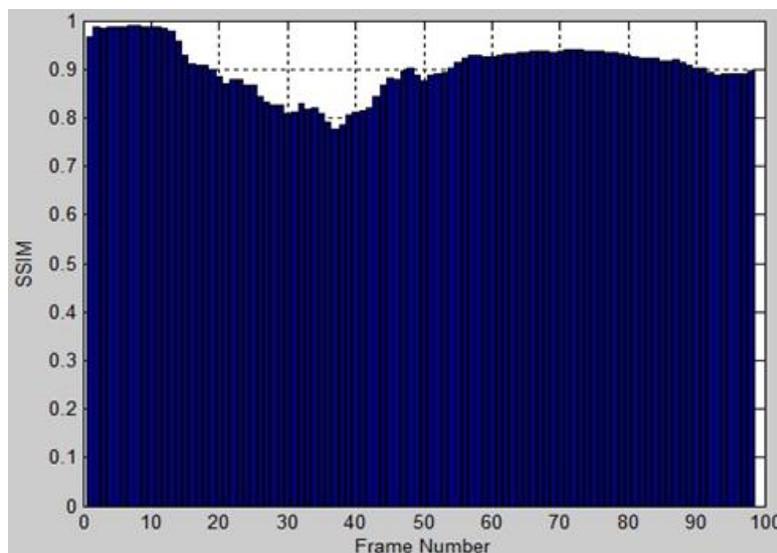


Fig 10: SSIM value for all different frames.

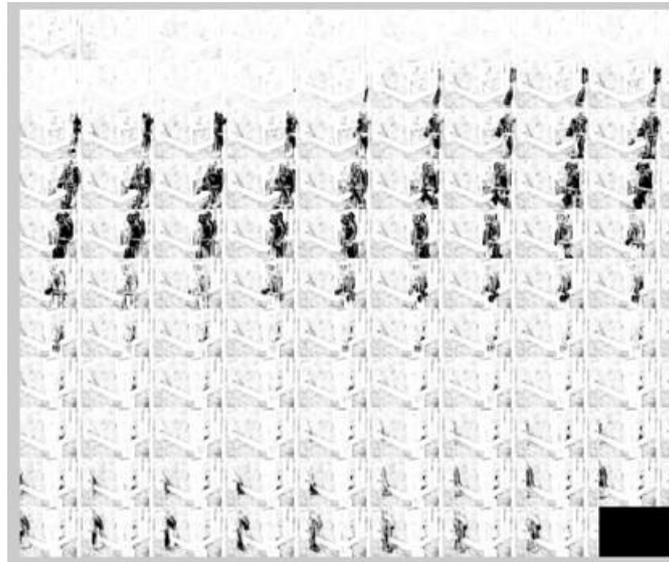


Fig 11: SSIM map for the interpolated frames.

VI.APPLICATIONS

Detection of motioned object is plays a vital role in video processing and video coding technique, as my project says that true motion estimation gives a tracked object motion as closely as possible, so using this coherent motion vector field which is clustered is helps in application called as motion compensated temporal frames interpolation. This means the adding extra frames in between the existing frame for increasing the visual quality of video. This overall application is used in the broadcasting the video, multimedia players and what are all video producing devices are there we can implement these technique for increase the overall video quality.

Compression of video:

This is the main concept in video coding application for reducing the temporal redundancy this means only the motioned object is coded and remaining stationary background is kept as it is by giving the predicting value to it .This true motion is also helps in video coding by tracking the motion object as closely as possible.Also it make the utilization of the lower bandwidth.

Target tracking and traffic analysis (traffic surveillance)

In generally intelligent traffic system (ITS) has become one of worldwide hotspot in recent year. Video based traffic flow parameter detection system has several apparent advantages such as easily investigated and lowest cost .the true motion estimation is gives the counting of vehicles, estimating the speed tracking it and detection of event of traffic violation. The clear picture of video and flow of vehicle can be detected by using this true motion estimation technique. Also this is helps in CCTV surveillance as the theft is moving very fastly means that person can be tracked by using this method and that person picture is obtained clearly by adopting the spatial interpolation technique.

Texture extraction, Object extraction from the video sequences

Some of the text and object which is moving in video sequences. By using the true motion estimation technique it is possible to extract the texture and different object in the video sequences by finding out the true motion estimation with the clustered the motion object with the basics of the some characteristics, like it is alphabets, or number etc,

VII.CONCLUSION

A True-motion estimation algorithm and its application to MCFRUC is presented Computational-complexity, regularity, and memory bandwidth are considered when designing the algorithm so that a low-complexity. The proposed TME algorithm imposes implicit and explicit smoothness constraints on BMA after finding out motion estimation the clustering of motion blocks is used which is gives the coherent motion vector field.Then by using the true motion estimation the motion compensated temporal frame is obtained .lastly the performance parameter is also calculated .The SSIM of both interpolated and original frame is getting near to one, this shows the interpolated frame are good after the production of video, then the good video sequences are getting and it play very smoothly.



REFERENCES

- [1] A.-M. Huang and T. Nguyen, "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 740–752, Apr. 2009.
- [2] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "Truemotion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.
- [3] G. de Haan and P. Biezen, "An efficient true-motion estimator using candidate vectors from a parametric motion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 1, pp. 85–91, Feb. 1998.
- [4] G. de Haan, "IC for motion-compensated de-interlacing, noise reduction, and picture-rate conversion," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 617–624, Aug. 1999.
- [5] Y. Ling, J. Wang, Y. Liu, and W. Zhang, "A novel spatial and temporal correlation integrated based motion-compensated interpolation for frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 863–869, May 2008.
- [6] A.-M. Huang and T. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 694–708, May 2008.
- [7] B.-T. Choi, S.-H. Lee, Y.-J. Park, and S.-J. Ko, "Frame rate upconversion using the wavelet transform," in *Proc. Int. Conf. Consum. Electron.*, Los Angeles, CA, Jun. 2000, pp. 172–173.
- [8] W. H. Lee, Y. Choi, K. Choi, and J. B. Ra, "Frame rate up conversion via image fusion based on variational approach," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 885–888.
- [9] S. Dikbas, T. Arici, and Y. Altunbasak, "Fast motion estimation with interpolation-free sub-sample accuracy," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 7, pp. 1047–1051, Jul. 2010.
- [10] A. Beric, G. de Haan, J. van Meerbergen, and R. Sethuraman, "Toward an efficient high quality picture-rate up-converter," in *Proc. Int. Conf. Image Process.*, vol. 2, Sep. 2003, pp. 363–366.
- [11] J. Konrad, "Motion detection and estimation," in *Handbook of Image and Video Processing*, A. Bovik, Ed., 2nd ed. Burlington, MA: Elsevier, 2005, ch. 3.10, pp. 253–274.
- [12] Y. Wang, Y.-Q. Zhang, and J. Ostermann, "Video Processing and Communications", 1st ed. Upper Saddle River, NJ: Prentice-Hall, 2001.
- [13] A. M. Tekalp, "Digital Video Processing". Upper Saddle River, NJ: Prentice-Hall, 1995.
- [14] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *IEEE Trans. Broadcast.*, vol. 49, no. 3, pp. 241–248, Sep. 2003.
- [15] S.-C. Tai, Y.-R. Chen, Z.-B. Huang, and C.-C. Wang, "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications," *J. Display Technol.*, vol. 4, no. 2, pp. 188–197, Jun. 2008.
- [16] Y.-R. Chen and S.-C. Tai, "True motion-compensated de-interlacing algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1489–1498, Oct. 2009.
- [17] J.-S. Kim and L.-S. Kim, "Noise robust motion refinement for motion compensated noise reduction," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 5, pp. 1581–1583, May 2008.
- [18] M. M. Richter, C. Dolar, and H. Schroder, "Coding artifact reduction by temporal filtering," in *Proc. 13th Int. Symp. Consum. Electron.*, Kyoto, Japan, May 2009, pp. 6–10.