

A New Diamond Search Algorithm for Adaptive Pixel Based Motion Estimation in Video Coding

Srinivas Bachu* and Manjunathachari K

Department of ECE, GITAM University, GITAM School of Technology
Hyderabad, Telangana, India
bachusrinivas@gmail.com

Abstract

We are intended to propose a video compression based on Modified Pixel-Based Motion Estimation (MPME) technique to solve the current problems in the literary works. The proposed system will be comprised of four phases, Displacement Calculation, Motion Estimation, Encoding and Decoding. In displacement calculation the I-frame and the P-frame has to been constructed from the video frames. In second phase, the motion vector has been finding out by using the Modified Pixel Based Motion estimation. In here we are creating four movement vector predictors (West Predictor, Neighbor Predictor, Median Predictor and Centre Predictor) with the help of diamond based search pattern. The proposed motion estimation technique is conducted on a pixel-by-pixel basis and the predicted P-frame has been developed using motion estimation technique. The residual is calculated with the original P-frame and the predicted P frame, and the residual is passed into next phase. After the motion vector estimation each frame is encoded by using JPEG-LS encoder. Next the encoded values of compressed video frames are decoded by using the decoding technique and for that JPEG-LS decoder is used. The proposed video compression using MPME will be implemented in the platform MATLAB and the performance of the video codec procedure is assessed by contrasting the consequence of proposed system with the conventional motion estimation algorithms.

Keywords: Diamond Search Algorithm, Motion Estimation, Mean Square Error, Compression Ratio, Peak Signal to Noise Ratio

1. Introduction

Video Coding is developed into a mature field, with a few techniques available for a wide range of applications [9]. Video compression refers to decreasing the amount of information used to represent video images [1]. It mostly used in video conference and real time. In actuality, movement estimation based encoders are the most generally utilized in video compression techniques [3]. Video sequences are moving pictures which are tested at regular intervals generally, 25 frames for each second and put away as grouping of frames [14]. The motion can be estimates with the encoder, which occurred between the reference and current frames. This procedure is called motion estimation (ME) [5]. Motion estimation is a key segment for high quality video compression, which is categorized by its huge computation complexity and memory requirements [10]. It forms the core of video compression and video processing applications [4]. Since motion estimation is effective in removing temporal redundancies; it has become an integral part of all high-compression video codec's [6].

Video compression efficiency and the compression quality are determined by the motion estimation algorithm, development of Quick Movement Estimation Calculation for continuous application becomes compelling [12]. The efficiency of a motion

*Corresponding Author

estimation calculation is profoundly related with the speed, compression improvement ratio and the image quality of decoded video [15]. Block motion estimation algorithms have been widely used in video coding because of its simplicity and coding efficiency of motion vectors [16]. Motion estimation is a critical component of block-based video compression method, since the utilization of temporal redundancy between consecutive frames of a video grouping takes into consideration noteworthy pressure of video information [2]. In this situation, block in the reference frame that yields the least cost or highest correlation is normally selected as a prediction of the processed block and one dislodging vector depicting the area of the best coordinated square is the only information needed by the decoder to predict the processed block [7].

In the motion estimation algorithm, to look for the most similar block with the current frame in reference frames, MSE (Mean Square Error) and SAD (Sum of Absolute Difference) as estimation methods are being used [17]. The target of video coding in most video applications is to diminish the measure of video information for putting away or transmission purposes without influencing the visual quality [11]. In order to improve a video quality under a limited bit rate, most of video codec's employs fractional samples in a process of movement estimation or compensation [18]. Extracted block pixel value can be subtracted and analyzed for motion estimation [19]. Video codec scheme that expects to address both the parts of coding effectiveness and usage cost in a very much adjusted relationship [13]. Video compression is useful because it helps to reduce the utilization of costly resources such as Google drive, information storage on servers and transmission bandwidths [8]. It is vital for engineers of inserted frameworks, processors and tools targeting video applications [20].

2. Related Works

Xuena Bao *et. al.*, [21] have developed a hardware efficient fast algorithm with a lossless frame recompression (FRC) scheme and early-level end procedure for huge search range (SR) motion estimation utilized in beyond high definition video encoder. It also reduces the bandwidth.

Humaira Nisar *et. al.*, [22] have presented a motion estimation algorithm that is suitable for all kinds of video sequences. The proposed calculation includes a multistage approach that includes movement vector expectation and movement order utilizing the qualities of video sequences. It gives better feature quality and less multifaceted nature when contrasted with other ME calculations.

Avishek Saha *et. al.*, [23] have focused on reducing the number of search locations in block matching based motion estimation. This algorithm eliminates neighbouring blocks which are having low probability of being best match to the candidate block with the spatial correlation method. The proposed calculation is free of any such at first altered inquiry designs. The decision to wipe out the range was taken progressively in view of a preset edge T_h . The extent to which the neighborhood can be eliminated is configured using the shift parameter d . Thus, reduction in the number of search positions changes powerfully depending upon data content. An Adaptive Neighborhood Elimination Algorithm (ANEAL) has been proposed whereby the T_h and d parameters are updated adaptively.

Lili Hsieh *et. al.*, [24] have presented two effective quick movement estimation calculations with a two-phase prescient pursuit taking into account joint spatial-transient connections are proposed to decrease the inquiry intricacy. In the main stage, a harsh inquiry from the given movement vectors connected with six spatially and transiently associated pieces endeavor's to establish a beginning stage of the satisfactory pursuit extend that is nearer to the worldwide ideal. In the next stage, block-based gradient descent search and predictive partial search algorithms were used for fine search to elaborately search the adequate range from the beginning stage for the best movement

vector. Recreation results have demonstrated that the calculations viably diminish the normal number of checked focuses to just 1.55% and 0.78% when contrasted with the full pursuit strategy and yield an incredible execution change regarding PSNR, computational complexity and bit rates when contrasted with full inquiry and some notable quick pursuit techniques.

P.R. Hill *et. al.*, [25] have proposed a scheme for sub-pixel movement estimation taking into account the entire pixel SAD dispersion. Both half-pixel and quarter-pixel searches were guided by a model free estimation of the SAD surface utilizing a two dimensional part framework. While giving an identical rate mutilation execution, this methodology give or take parts the quantity of quarter-pixel search positions surrendering a general velocity of give or take 10% contrasted with the EPZS quarter-pixel technique.

3. Motivating Scenario

In the literature work, many motion estimation techniques have been proposed for effective video compression. One of the video compressions based on motion estimation technique is given in [23]. In this paper an Adaptive Neighborhood Elimination Algorithm (ANEA) is used, which is based on the spatial correlation property of collocated macro blocks and this algorithm is independent of any such initially fixed search patterns. In here preset threshold and shift parameter are used, they are updated adaptively in this ANEA. But this technique has the drawback that the compression efficiency is poor comparing to full search method and the superior performance ANEA can demonstrated on both ME Package and the H.264 reference software only. On the other hand, one technique for video compression based on motion estimation is given in [25].

In this paper the sub pixel motion estimation have been done using the kernel method. Sub-pixel motion estimation is implemented within these standards using interpolated values at 1/2 or 1/4 pixel. Both half-pixel and quarter-pixel searches are guided by a model-free estimation of the SAD surface using a two dimensional kernel method. It yield good accuracy with better performance but it has some drawbacks. While comparing this kernel method to other methods results its speed up is only 10% higher than the other methods. So there is a need of improved motion estimation technique to develop the improved video compression process. All the aforementioned problems in the current literature motivated to do the research work in this area.

4. Video Compression Technique based on Modified Pixel-Based Motion Estimation (MPME)

We proposed a video compression technique based on Modified Pixel-Based Motion Estimation (MPME). Efficiently compressing the video is the main objective of this research. Here motion estimation technique is conducted on a pixel-by-pixel basis. The proposed technique will comprised of four stages namely, Displacement Calculation, Motion Estimation, Encoding and Decoding.

A video is a compilation of an enormous amount of still frames which exhibits a motion like feature when the frames are changed in quick sequence. In this process, to compress the entire video, each frame of the video is compressed.

Let $v_i: i = 1, 2, \dots, n$ be a database video,

Where n is the complete number of test videos present in the database.

The video is an agreement of edges of size $H \times B$ that can be represented as $f_j(x, y) : j=1,2,\dots,N_f, x=0,1,2,\dots,H-1$ and $y=0,1,2,\dots,B-1$, where, N_f is the total number of frames present in the i^{th} database video.

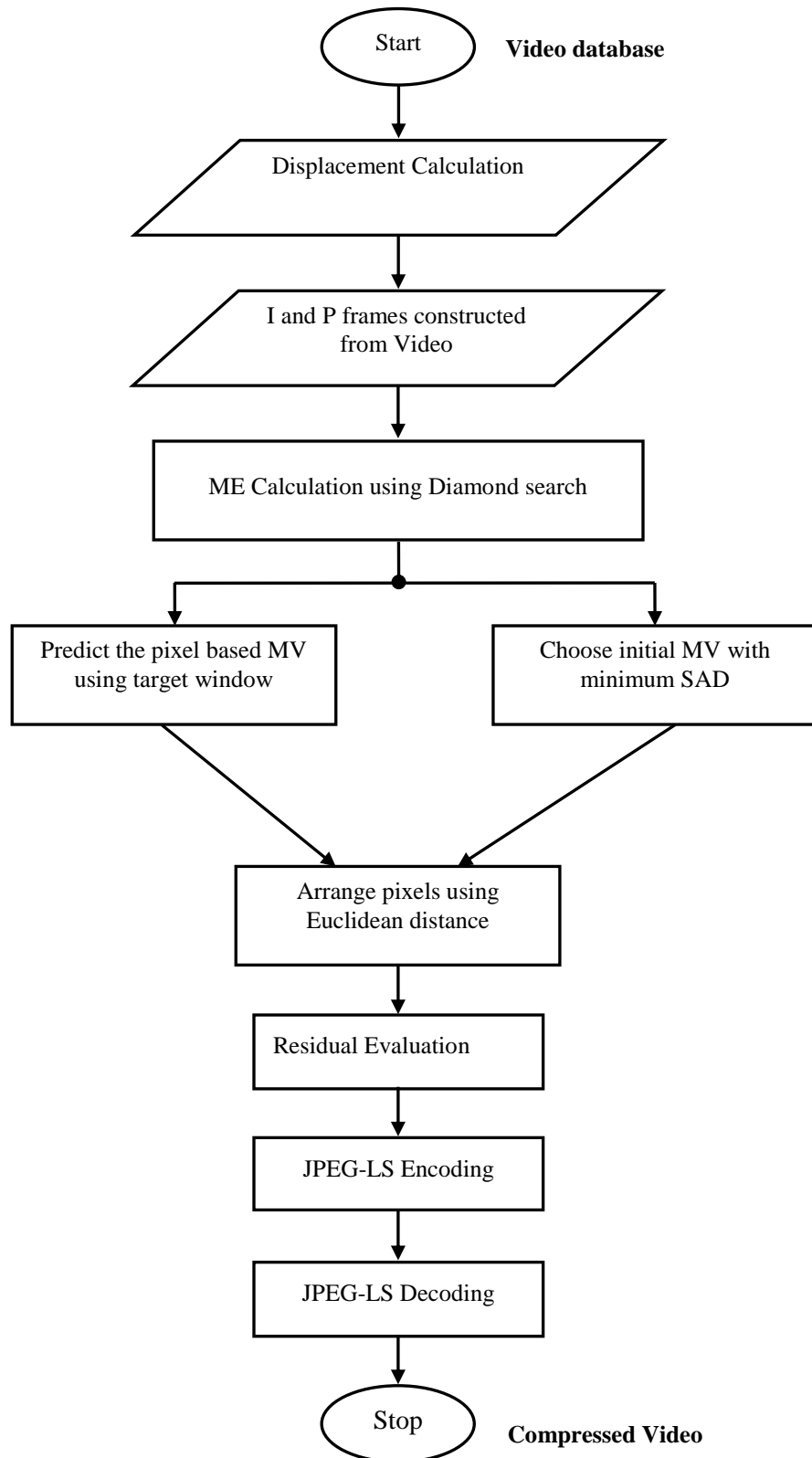


Figure 1. Flowchart of MPME Technique

The proposed MPME method is conducted on a pixel-by-pixel basis. Pixel based approach cares for each individual pixel and possibly achieves good motion accuracy and it is more prediction based. The flowchart of Modified Pixel based Motion Estimation (MPME) technique is shown in Figure 1.

Our proposed Modified Pixel-Based Motion Estimation (MPME) technique will comprise four stages namely: Displacement Calculation, Motion Estimation using diamond search pattern, Encoding and Decoding.

4.1. Displacement Calculation

A set of frames representing a moving scene is a set of sequential samples. In average scene there will be part of closeness between successive frames. This implies we will have achievement in anticipating one edge from the other. A first frame is the reference frame and the next successive frames are the predicted frames. The reference frames is the I-frame and the predicted frames are the P-frames. In Displacement calculation the I-frame and the P-frame has been constructed from the video frames.

$$(f_i(x, y)) \text{ Represent the I-Frame} \quad (1)$$

$$(f_p(x, y)) \text{ Represent the P-frame} \quad (2)$$

4.1.1. RGB Color Model: Here we are using color video representing the RGB color model. For each I frame and the predicted P frames compression occurs at the R component, G component and the B component. Mean value of these components were taken for each frame and then the motion estimation is calculated for these I frame and the predicted P frames.

4.2. Motion Estimation

Then for these frames motion estimation is calculated using the diamond search pattern. Motion estimation is a method that acquires the displacement between the pixels of the I-frame ($f_i(x, y)$) and the P-frame ($f_p(x, y)$).

The predicted frame is reconstructed using the Motion Vectors, after performing the motion estimation. Finally the residual is evaluated, which is nothing but the difference between the original P-frame and the predicted P-frame.

In order to produce better quality predictive frames, the residual is transmitted along with the MV's. From the MV, we choose the initial MV with the minimum sum of absolute difference in a target window.

$$\overrightarrow{M}t_{V_{init}}(p, q) = \text{argminSAD}(p, q) \quad (3)$$

$$\text{SAD}(p, q) = \sum_{(a, b) \in \text{Trgw}} |p_i(a, b) - p_{i-1}(a + p, b + q)| \quad (4)$$

Where (p, q) is the initial MV candidates, Trgw is the target window, $p_i(a, b)$ and $p_{i-1}(a, b)$, $(a, b) \in \text{Trgw}$ denotes the pixel within the target window in the current frame i and previous frame $i-1$. The pixel-based predictor utilizes an objective window to assess the movement between the current frame i and previous frame $i-1$. We define the target window as upper-left pixels of the current pixel $p_i(a, b)$ in frame i . The request of the pixels is organized by Euclidean separation to the present pixel. The shape and the size of the target window can be adjusted if desirable. The residual is calculated with the original P-frame and the predicted P frame. The $P_{ref}(s, t)$ and $P_{prc}(s, t)$ are the pixel

values of the reference and the predicted frame images respectively. And the residual is passed into next phase of encoding.

4.3. PEG-LS Encoding

JPEG-LS is the newly ITU/ISO standard for lossless picture pressure among different existing lossless pressure plans. This calculation is of moderately low multifaceted nature, low storage JPEG-LS is a lossless compression standard for continuous-tone still images. Low Complexity Lossless Compression is the core algorithm of the modeling part of JPEG-LS.

Encoding Procedure

According to the JPEG-LS standard, the image is read and then sampled in a raster scan order. Inside this standard there are prediction, residual modeling and context-based coding of the residuals. The encoding mode of this standard has two types which are described in the following sections

4.3.1. Regular Mode: The regular mode occurs when a pixel has a value that is different from its previously scanned neighborhood pixel of the same image line. Pixels that are encoded in this mode are called regular-mode pixels. Regular mode pixels achieve encoding by predicting their values based on their context numbers. Then the prediction error is encoded by a Golomb coder.

Regular Mode steps:

Step: 1 Modeling:

1 Determine the context

1.1 Compute the local gradient(D_1, D_2, D_3)

1.2 Select the mode(Regular/Run)

1.3 Quantize the local gradient(into 9 regions)

1.4 Merge the quantized gradient

(calculate the context number of the current sample by mapping)

If it is regular mode do the following..... Otherwise go to run mode.

Step: 2. Prediction

2.1 Detect the edge

(median edge detector is used to find X)

2.2 Correct the prediction using adaptive correction mechanism

2.3 Compute the prediction error using the original pixel value and corrected Prediction value

2.4 Reduce the prediction error within the range

Step: 3 Prediction errors encoding

3.1 Compute the golomb code variable

3.2 Error mapping (map the error value to non negative integer)

3.3 Encode the mapped error using limited length Golomb function

i. Fix one parameter M

- ii. Find quotient $q = \text{int}(\text{mapped error}/M)$ $r = \text{mapped error} \bmod M$
- iii. Generate code word
 - < Quotient word >
 - < Remainder code >
 - < Quotient code >
 - 'n' no of 1 bits($n=q$)+ a zero
 - < Remainder code >
- i. If $M = 2^k$, remainder as binary format
- ii. If $M \neq 2^k$, set $b = \lceil \log_2(M) \rceil$
 - Calculate cut off $2^b - M$
 - If $r < 2^b - M$ $r = b-1$ bits
 - If $r \geq 2^b - M$ code = $r + 2^b - M$ using b bits

Step: 4 Update the context variable

4.3.2. Run Mode: The run mode occurs when a pixel has the same value as its previously scanned neighborhood pixel of the same image line. Pixels that are encoded in this mode are called run-mode pixels. The encoding procedure for this mode is slightly different than that of the regular mode. First of all there is no prediction in this mode. It encodes only the count of the consecutive run-mode pixels that constitutes a run-segment.

Run Mode steps:

Step: 1 Determine the context

- 1.1 Compute the local gradient(D_1, D_2, D_3)
- 1.2 Select the mode(Regular/Run)
- 1.3 Quantize the local gradient (into 9 regions)
- 1.4 Merge the quantized gradient (calculate the context number of the current sample by mapping)
- 3.4 Run scanning and run-length coding (encodes the run segment value)
- 3.5 Run interruption coding
 - 3.5.1 Find interrupted pixel
 - 3.5.2 Find the index

Step: 2 Predictions

- 2.1 Detect the edge (median edge detector is used to find X)
- 2.2 Correct the prediction using adaptive correction mechanism
- 2.3 Compute the prediction error using the original pixel value and corrected Prediction value.

Step: 3 Reduce the prediction error within the range

Step: 4 Temp variables are calculated for computing k.

Step: 5 Context variable $q = 365 + \text{index}$. Again do the same in regular mode.

4.4. JPEG-LS Decoding

Decoder is responsible for recovering the original image from the compressed data format. Decoder is very similar to the encoder the main difference is that in the decoder the current pixel x is not known. The predictor block is used in both encoder and the decoder. The main function is to predict the current pixel.

5. Results and Discussions

Our proposed Video Compression technique has been validated by experiments with a variety of video sequences. The proposed system has been implemented in the working platform of MATLAB (Version 7.12) with machine configuration as follows:

Processor: Intel core i5

OS: Windows 7

CPU speed: 3.20 GHz

RAM: 4GB

Here we are using Modified Pixel-Based Motion Estimation (MPME) technique for the purpose of Video Compression. The videos from Database are subjected to displacement calculation in order to construct the I Frame and the P Frame. The next phase is to predict the motion vector using modified pixel based motion estimation. West Predictor, Neighbor Predictor, Median Predictor and Centre Predictor are the four predictors used here. Then the residual is calculated between the original P frame and the predicted P frame. After the motion vector estimation each frame is encoded by using JPEG-LS encoder. Next the encoded values of compressed video frames are decoded by using the decoding technique and for that JPEG-LS decoder is used.

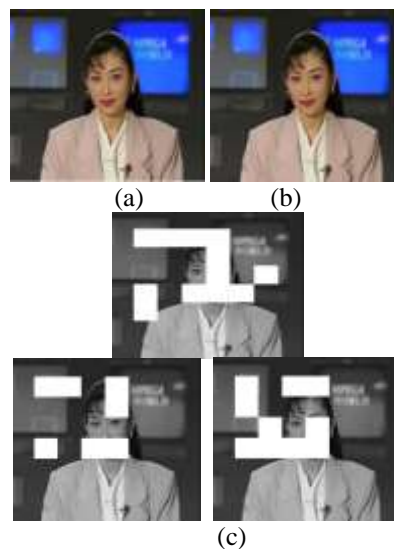


Figure 2. The Sample Output Obtained from the Motion Estimation Process in a Database Video 2

a) I-frame b) P-frame c) estimated motion in P-frame

Five sample videos ie, miss, akiyo, grandma, football, carphone, were taken from the database. The results will be analyzed to demonstrate the performance of the proposed video compression technique with the existing techniques. Initially, the video comprises of set of frames. The initial frame is the I-frame and the frame next to the I-frame is the predicted frame *i.e.*, P-frame are shown in Figure 2.

Performance Analysis





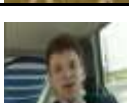
To analyze the result of our proposed Modified Pixel-Based Motion Estimation (MPME) technique system, the Compression Ratio (CR), Peak Signal to Noise Ratio (PSNR) and Average No. of Search Points parameters are used. To evaluate the performance of the motion estimation technique in the proposed system, the Peak Signal to Noise Ratio (PSNR) based on the Mean Square Error (MSE) is used as a quality measure and its value can be determined using the following equation.

$$PSNR = 10 \log \left(\frac{(255)^2}{MSE} \right) dB \tag{5}$$

$$MSE = \frac{1}{k} \sum (P_{ref}(s,t) - P_{prc}(s,t))^2 \tag{6}$$

Here is the total number of pixels in the image. The summation of PSNR, over the image frames, will then be divided by the total number of frames to obtain the average value.

Table 1. PSNR Values, Compression Ratio, Average no of Search Points for the Proposed JPEG-LS Encoder and the Existing LSK, Huffman and Run Length Encoder

Video Frames	Average no of Search Points =0.156							
	Proposed(MPME using DS)		LSK		Huffman		Run Length	
	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR
	69.823	0.546	29.791	0.575	3.702	0.736	-71.216	0.949
	59.500	0.625	23.471	0.380	6.686	0.910	-82.167	0.975
	65.033	0.862	26.347	0.365	4.510	0.837	-72.651	0.948
	54.227	0.516	30.633	0.238	8.858	0.784	-80.735	1
	52.423	0.507	26.583	0.331	6.888	0.935	-82.613	0.993

Compression Ratio

Compression Ratio (CR) is the ratio between the numbers of bits required storing the image before compression (I) and the number of bits required to store the image after compression (O).

$$CR = I / O \quad (7)$$

Here I is the size of the original frame and O is the size of the compressed frame. The performance of the proposed system is compared with other existing schemes using the PSNR value, Average no of Search Points, Compression Ratio over more number of frames obtained from the experiment. Here, we compare our proposed Modified Pixel-Based Motion Estimation (MPME) technique with the existing technique.

Discussion

From Table 1, Proposed JPEG-LS encoder technique have higher PSNR rate when compared to the other existing LSK, Huffman and Run length encoder. Average Number of search points for the proposed technique and the existing LSK, Huffman and Run length encoder techniques are similar. But the Table 1, shows that our proposed algorithm has higher compression ratio value than the other existing LSK, Huffman and Run length algorithms for the 5 different kinds of dataset images. Hence from the performance analysis our proposed system gives better motion estimation than any other existing techniques.

5. Conclusion

In this paper we addressed the progress of video compression in Multimedia. We have deliberated to propose a new efficient video compression system based on the Modified Pixel-Based Motion Estimation (MPME) technique in order to reduce the shortcoming in the existing method. Moreover, in comparative analysis, our proposed technique performance is compared with the existing motion estimation algorithm. From the comparison result it has been shown that the PSNR and the compression ratio of our proposed Pixel-Based Motion Estimation the method is higher than the existing techniques. Thus our proposed Pixel-Based Motion Estimation (MPME) technique more precisely and efficiently perform the video compression.

References

- [1] F. Jonathan, D. Severine and B. Dominique, "Motion Compensation based on Tangent Distance Prediction for Video Compression", *Journal of Signal Processing: Image Communication*, vol. 27, no. 2, (2012), pp. 153–171.
- [2] C. Jing and P. David, "Fast and Accurate Block based Motion Estimation Algorithms using Particle Swarm Optimization", *International Journal of Information Sciences*, no. 197, pp. 53–64.
- [3] A. Thazni and J. D. Raveena, "Motion Estimation and Motion Compensated Video Compression using DCT and DWT", *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 12, (2012), pp. 667-670.
- [4] M. Deepa and S. V. Thomas, "A Novel Based Approach for Finding Motion Estimation in Video Compression", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 8, (2012), pp. 514-520.
- [5] A. P. Immanuel, B. J. Becky and G. Alma, "A Study on Block Matching Algorithms for Motion Estimation", *International Journal on Computer Science and Engineering*, vol. 3, no. 1, (2011), pp. 34-41.
- [6] S. Modukuri, R. Maher, S. P. Sharkawy and M. El, "High Speed Search Algorithms for Block-Based Motion Estimation Video Compression", In *Proceedings of International Conference on computer Engineering and System*, Cairo, (2006), pp. 308-312.
- [7] K. Panusoponet ad K. R. Rao, "Efficient Motion Estimation for Block Based Video Compression", In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, no. 4, (1997), pp. 2677-2680.
- [8] J. Dharmendra, K. Fremin, J. Justin and P. Stevew, "Motion Estimation Algorithms for Baseline Profile of H.264 Video Codec", *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 4, (2013), pp. 727-731.
- [9] A. Ishfaq, H. Yong and L L. Ming, "Video Compression with Parallel Processing", *Journal of Parallel Computing*, vol. 28, no. 7, (2008), pp. 1039–1078.
- [10] A. A. Benboubker, "Shape-Adaptive Motion Estimation Algorithm for MPEG-4 Video Coding", *International Journal of Computer Science Issues*, vol. 7, no. 2, (2010), pp. 56-61.

- [11] O. T. Ayedi and W. A. Mohamed, "New Low Complexity DCT based Video Compression Method", In Proceedings of International Conference on Telecommunication, Marrakech, pp. 202-207.
- [12] C. A. Kumar, T. Sudhanshu and M. Rana, "An Improved Adaptive Cross Pattern Search (IACPS) Algorithm for Block Motion Estimation in Video Compression", In Proceedings of International Conference on Computational Intelligence and Multimedia Applications, Sivakasi, no. 3, (2007), pp. 89-96.
- [13] M. Detlev, S. Heiko, B. Sebastian, B. Benjamin and H. Philipp, "Video Compression using Nested Quad-tree Structures, Leaf Merging and Improved Techniques for Motion Representation and Entropy Coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, (2010), pp. 1-9.
- [14] S. Jeyakumar and S. Sundaravivelu, "Implementation of Parallel Motion Estimation Model for Video Sequence Compression", In Proceedings of International Conference on computing, Comm. and Networking, Thiruchendur, (2008), pp. 1-5.
- [15] Y. C. Zhang, S. Wan and S. Tingzhi, "Fast Sub-Pixel Motion Estimation based on Directional Information and Adaptive Block Classification", In Proceedings of Fifth International Conf. on Visual Information Engg., Xian China, (2008), pp. 622 – 627.
- [16] C. Y. Lani, W. Hui, L. Siu and W. Chi, "A Block Motion Vector Estimation Using Pattern Based, Pixel Decimation", In Proceedings of IEEE International Symposium on Circuits and Systems, Hong Kong, no. 2, (1997), pp. 1153-1156.
- [17] K. Dong, H. Park, C. Moon, B. C. Sang and J. H. Lee, "A Fast Motion Estimation Algorithm for SAD Optimization in Sub-pixel", In Proceedings of International symposium on Integrated circuits, Singapore, (2007), 528-531.
- [18] S. J. Yoon, K. Kim, B. Woo and L. Jeong, "The Analysis of a Fractional Pixels Generated by a Combination of Linear FIR Filters in a Video Codec", In Proceedings of 13th International Conference on Advanced Communication Technology, Seoul, (2011), pp. 660-663.
- [19] S. Jitvinder, K. C. Ranjit, A. Lim and J. Salim, "Image Pixel Comparison using Block Based Positioning Subtraction Technique for Motion Estimation", In Proceedings of Fifth Asia Modeling Symposium, Kuala Lumpur, (2011), pp. 89-93.
- [20] M. Mozammil and M. I. Zakariya, "Analysis of Video Compression Algorithms on Different Video Files", In Proceedings of Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, (2010), pp. 257-262.
- [21] B. Xuena, Z. Dajiang, L. Peilin and G. Satoshi, "An Advanced Hierarchical Motion Estimation Scheme With Lossless Frame Recompression and Early-Level Termination for Beyond High-Definition Video Coding", IEEE Transactions on Multimedia, vol. 14, no. 2, (2012), pp. 237-249.
- [22] N. Humaira, S. Aamir, M. Sun and C. Tae, "Content Adaptive Fast Motion Estimation based on Spatio Temporal Homogeneity Analysis and Motion Classification", Pattern Recognition Letters, vol. 33, no. 1, (2012), pp. 52–61.
- [23] S. Avishek, M. Jayanta and S. Shamik, "A Neighborhood Elimination Approach for Block Matching in Motion Estimation", Signal Processing: Image Communication, vol. 26, vol. 8, (2011), pp. 438–454.
- [24] W. L. Hsieh, C. Shiung and H. L. Chuan, "Motion Estimation using Two-Stage Predictive Search Algorithms based on Joint Spatio-Temporal Correlation Information", Journal on Expert Systems with Applications, vol. 38, no. 9, (2011), pp. 11608–11623.
- [25] P. R Hill and D. R Bull, "Sub-pixel Motion Estimation using Kernel Methods", Signal Processing: Image Communication, vol. 25, no. 4, (2010), pp. 268–275.

Authors



Srinivas Bachu, Research Scholar from GITAM University Hyderabad, and received M.Tech degree from JNT University, Hyderabad, has 11 years of teaching experience. At present he is working as an Associate Professor, Department of ECE at Guru Nanak Institutions Technical Campus (Autonomous), Telangana. He is the Life Member of ISTE, AMIE, and IAENG. He published 3 Text Books and 13 Research Papers at Refereed International Conferences & Journals. He is one of the Reviewer Board Member in four International Journals. His areas of research interest are Signal, Image and Video Processing.



K. Manjunathachari, awarded Ph.D from JNT University, Kakinada, and M.Tech from JNT University, Hyderabad, has an experience of more than 20 years of teaching and 3 years of industry. At present Dr. K. Manjunathachari working with the GITAM University, GITAM School of Technology, Hyderabad, Telangana, as a Professor and Head of the ECE department. He published one Text Book and more than 20 Research Papers at Refereed International Conferences & International Journals. His areas of research interest are Signal and Image Processing.