Motion Estimation in Video Coding using Simplified Optical Flow Technique

Eskinder Ayele and Sanjay Dhok
Center for VLSI and Nanotechnology, Visvesvaraya National Institute of Technology, India

Abstract: *We propose an interpolation-free sub-pixel Motion Estimation (ME) technique that particularly aims at providing accurate Motion Vectors (MVs), where conventional sub-pixel interpolation ME algorithms used in video coding are too complex and time consuming. The proposed algorithm is a combination of block matching algorithms and simplified optical flow, which is Taylor approximation. The technique does not require any pixel interpolation and it is much faster than conventional ME methods. Statistical results illustrate that the new technique performs quickly and accurately with a compatible performance with respect to the benchmarking full search algorithm.*

Keywords: *Sub-pixel ME, block matching, interpolation, interpolation-free, optical flow.*

Received November 11, 2013; accepted November 25, 2014

1. Introduction

The amalgamation of Motion Estimation (ME) and Motion Compensation (MC) forms a significant component of video codec. ME is the method of finding Motion Vectors (MVs) that describe the transformation between successive frames in a video sequence. MC is the course of applying the MVs to the current frame to synthesize the transformation to the reference frame. Since each frame in a typical video sequence is mostly made up of some changed regions of another frame, by exploiting strong inter-frame correlation along the temporal dimensions, one can remove temporal redundancy and achieve video compression.

In video compression, the performance of the coding scheme is directly affected by the performance of ME algorithms and it is the most demanding and time consuming stage [1]. Therefore, block based ME is still relevant research topic. Most of the existing ME algorithms require sub-pixel interpolation of inter-pixel values which unfortunately increases the overall complexity, data flow and decreases estimation accuracy. However, the proposed ME algorithm enables the estimating of a MV with reduced computation cost while maintaining high sub-pixel accuracy.

ME algorithms in video compression perform both estimation and compensation simultaneously. In the estimation part, the algorithm has to implicitly interpolate the frame if sub-pixel accuracy is required. However, the associated cost is higher for more precise MVs. For example, if we want to achieve 0.125-pel accuracy, then we need to enlarge the image by 8 times along each direction. Although, state-of-art algorithms can selectively choose where to interpolate, their cost is still high [4]. In this paper, we are mainly concerned with block based ME techniques followed by optical flow in sub-pixel level in order to get accurate MVs. The proposed technique can be useful for motion deblurring. In order to restore a video, good approximate of the motion blur Point Spread Function (PSF), and hence the MV, is required [10]. ME technique can also be used for video filtering applications such as video denoising, video stabilization [22] and super-resolution [24].

To improve the sub-pixel ME performance of block-matching methods, an interpolation-free scheme of applying the simplified optical flow (Taylor approximation) to ME is proposed. This method mainly consists of two processing stages. First, a conventional ME method is applied to obtain the MV at integer-pixel level and we shift the image block by the integer pixels along x and y directions. Here, since the shift is an integer factor, no interpolation is required. The second stage is the use of Taylor series approximation method to refine the search and improve the MV to sub-pixel accuracy based on the shift information. Experimental result shows that the proposed technique effectively reaches comparable PSNR performance with smoother MV fields as 1/4-pel conventional algorithm but with significant saving on computation cost. In order to, substantiate and demonstrate the efficiency of the algorithm, we will present some error analysis and discuss their insinuations.

The remainder of the paper is organized into the following seven sections. Section 2 gives a literature review of existing block-matching algorithms and a theoretical analysis of the existing research on sub-pixel ME. Section 3 provides a brief introduction and analysis of simplified optical flow and its procedure to estimate motion using optical flow equation. The proposed method, ME using simplified optical flow technique is described in section 4. Section 5 provides little error analysis and comparison to prove the expediency of our algorithm. Section 6 shows the simulation results and discussions and finally, section 7 concludes this paper.

2. Review of Sub-pixel Motion Estimation

2.1. Block Matching Algorithms

Because of the intensive computations and the large amount of resources required by ME, various algorithms have been developed in the past two decades. Methods that are most widely used today are block-based techniques, called Block-Matching Algorithms (BMAs), which find a matching block from one frame in another frame [25, 26]. In this technique, the frame is divided into 8×8 pixels blocks and scanned in a raster scanning order. Because the matching process is computationally intensive and because the motion is not expected to be significant between adjacent frames, the matching process is limited to a search window of a much smaller size than the image frame [6, 11]. Figure 1 illustrates the idea of block matching. The search window is centered at the centre of the current block in question. To account for the pixels at the boundaries, we have to pad both frames by 8 pixels all around the boundaries. The vector of displacement that results in the least value for the Sum of Absolute Difference (SAD) metric is the estimate of the MV for that block.

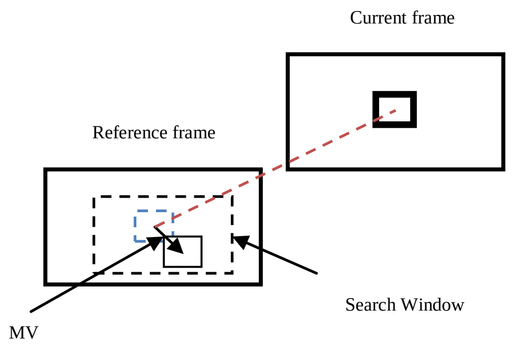


Figure 1. BMA process within a search window.

Till now a lot of research works have been done on developing fast and efficient BMAs. The methods used for ME in video compression differ in the matching criteria, the search strategy, and the determination of block size. Different searching approaches such as Exhaustive Search (ES), Cross Search, Three Step Search (TSS), New Three Step Search (NTSS) [12], Simple and Efficient TSS (SES) [17], Four Step Search (4SS) [20], Diamond Search (DS) [29], or Adaptive Rood Pattern Search (ARPS) [19] may be employed in BMA to evaluate possible candidate MVs over a predetermined neighbourhood search window to find the optimum MV.

2.2. Sub-pixel Motion Estimation Methods

It is obvious that actual scene motion has arbitrary accuracy and is oblivious to the pixel grid structure resulting from spatial sampling at Charge-Coupled Device (CCD) arrays or other A/D post-acquisition operation stages. The analyses, such as the work in [7], have established that sub-pixel accuracy has a significant impact on motion compensated prediction error performance for a wide range of natural moving scenes. There are two significant issues to develop fast and effective sub-pixel ME algorithms. First, the computation overhead by sub-pixel ME has become relatively significant while the complexity of integer-pixel search has been greatly reduced by fast algorithms. For instance, there have been integer-pixel ME algorithms [16, 23, 28] that only need between three and five integer search points to calculate the final integer MV. The second issue is reducing sub-pixel search point can greatly save the computation time for sub-pixel interpolation. Typical sub-pixel searches require interpolating sub-pixel values for computing the SAD.

In general, the sub-pixel ME process contains two stages: Integer pixel search over a large area and sub-pixel search around the best selected integer pixel. According to Chen's analysis in [3], the integer-pixel matching error surface is far from a unimodal surface inside the searching window due to the complexity of the video content. However, for the sub-pixel matching error surface, the unimodal surface assumption holds in most cases because of the smaller search range of sub-pixel ME as well as the high correlation between sub-pixels due to the sub-pixel interpolation.

In [15], a novel fast sub-pixel ME algorithm is proposed which performs a "rough" sub-pixel search before the partition selection, and performs a "precise" sub-pixel search for the best partition. By reducing the searching load for the large number of non-best partitions, the computation complexity for sub-pixel search can be greatly decreased. There has been much research on fast sub-pixel ME [3, 21, 27]. Most of these methods perform the sub-pixel search in two steps. First, predict a Sub-Pixel MV (SPMV). Second, perform a small area search around the SPMV to obtain the final SPMV. In this paper we propose a method for obtaining high-accuracy sub-pixel motion estimates using a simplified optical flow (Taylor approximation) technique that merely aims at providing accurate MVs without interpolation.

3. Simplified Optical Flow (Taylor Approximation)

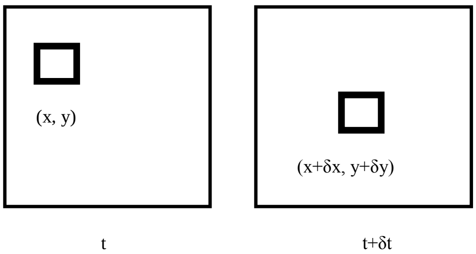
Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene. It indicates how much each image pixel moves between adjacent frames in which the moving patterns cause temporal varieties of the image brightness with the assumption that all temporal intensity changes are due to motion [2]. So, it can be used to estimate the MVs in a video sequence for the local image motion based upon local derivatives in a given sequence of frames from time-varying image intensity. However, optical flow basically works when the MVs are less than 1 pixel.

A common starting point for optical flow estimation is to assume that pixel intensities are translated from one frame to the next by assuming that $I(x, y, t)$ is the centre pixel in a NxN neighbourhood and moves by δx, δy in time δt to $I(x+\delta x, y+\delta y, t+\delta t)$.

Since $I(x, y, t)$ and $I(x+\delta x, y+\delta y, t+\delta t)$ are the images (frames) of the matching point, they are the same. So, we have:

$$I(x, y, t) = I(x+\delta x, y+\delta y, t+\delta t) \quad (1)$$

This assumption forms the basis of the 2D motion constraint equation and illustrated in Figure 2 below. It is true to a first approximation (small local translations) provided δx, δy, and δt are not too big.



a) Frame at time't'. b) Frame at time't+ δt'.

Figure 2. The image at position (x, y, t) is the same as at (x+δx, y+δy, t+δt).

In Taylor approximation we knew that, if you know a function and some of its derivatives at one point, you can approximate the function at nearby points. Formulae for approximating a function $F(t)$ for t near any fixed point t_0. The crudest approximation was just a constant.

$$F(t_0+\Delta t) \approx F(t_0) \quad (2)$$

Now, considering two successive frames as $f(x, y)$ and $g(x, y)$ [13], we describe a simplified version of classical optical flow [5].

$$g(x, y) = f(x+\Delta x, y+\Delta y) \quad (3)$$

Gradient-based approaches [9] proceed by taking the Taylor series expansion of the right hand side of Equation 3, yielding:

$$\begin{aligned} g(x, y) &= f(x+\Delta x, y+\Delta y) \\ &\approx f(x, y)+\Delta x\, \partial/\partial x\, f(x, y)+ \Delta y\, \partial/\partial y\, f(x, y) \end{aligned} \quad (4)$$

Thus, we assume that the displaced frame $f(x, y)$ is well approximated by Equation 4 that is called the first order Taylor series approximation by ignoring the higher-order terms.

Of course, one cannot recover $(\Delta x, \Delta y)$ from one gradient constraint since Equation 4 is one equation with two unknowns, Δx and Δy. One common way to further constrain $(\Delta x, \Delta y)$ is to use gradient constraints from nearby pixels, assuming they share the same 2D velocity. With many constraints there may be no velocity that simultaneously satisfies them all, so instead we find the displacement that minimizes the constraint errors. The optimal shift can be found by solving the Least-Squares (LS) estimator minimization problem, which minimizes the squared errors [18]:

$$E(\Delta x, \Delta y) = \sum_{x,y}\left(g(x,y) - f(x,y) - \Delta x \frac{\partial}{\partial x} f(x,y) - \Delta y \frac{\partial}{\partial y} f(x,y) \right)^2 \quad (5)$$

Since, this is a linear least-square problem, the optimal Δx, and Δy can be determined by setting the derivative of the objective function to zero. Therefore, the minimum of $E(\Delta x, \Delta y)$ can be found from its critical points, where its derivatives with respect to $(\Delta x, \Delta y)$ are zero; i.e.

$$\frac{\partial E(\Delta x, \Delta y)}{\partial x} = \sum_{x,y}\left[\frac{\partial f}{\partial x}(g-f) - \left(\frac{\partial f}{\partial x}\right)^2 \Delta x - \frac{\partial f}{\partial x}\frac{\partial f}{\partial y}\Delta y \right] = 0$$

$$\frac{\partial E(\Delta x, \Delta y)}{\partial \Delta y} = \sum_{x,y}\left[\frac{\partial f}{\partial y}(g-f) - \frac{\partial f}{\partial x}\frac{\partial f}{\partial y}\Delta x - \left(\frac{\partial f}{\partial y}\right)^2 \Delta y \right] = 0$$

Consequently we can setup the following system of linear equations for $u= (\Delta x, \Delta y)^T$ in matrix form as:

$$M\,u = b \quad (6)$$

Where the elements of M and b are:

$$M = \begin{bmatrix} \sum_{x,y}\left(\frac{\partial f}{\partial x}\right)^2 & \sum_{x,y}\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} \\ \sum_{x,y}\frac{\partial f}{\partial x}\frac{\partial f}{\partial y} & \sum_{x,y}\left(\frac{\partial f}{\partial y}\right)^2 \end{bmatrix}, \quad b = \begin{bmatrix} \sum_{x,y}(g-f)\frac{\partial f}{\partial x} \\ \sum_{x,y}(g-f)\frac{\partial f}{\partial y} \end{bmatrix}$$

When M has rank 2, then the least-square estimate is $u=M^{-1}b$. Therefore, by solving this system of linear equations we can determine the optimal solution by implicitly assumed that $|\Delta x|<<1$ and $|\Delta y|<<1$ in order to make Taylor approximation valid. To solve Equation 6 and for the computation of partial derivatives, one has to estimate the gradients $\partial f/\partial x$ and $\partial f/\partial y$. Since we are dealing with digital images defined on integer pixels, the gradients can be approximated using finite differences [9, 14], as given by:

$$\frac{\partial f}{\partial x} \approx \frac{1}{4}\begin{Bmatrix} f[m+1,n,k]-f[m,n,k]+f[m+1,n+1,k] \\ -f[m,n+1,k]+f[m+1,n,k+1]-f[m,n,k+1] \\ +f[m+1,n+1,k+1]-f[m,n+1,k+1] \end{Bmatrix} \quad (7)$$

$$\frac{\partial f}{\partial y} \approx \frac{1}{4}\begin{Bmatrix} f[m,n+1,k]-f[m,n,k]+f[m+1,n+1,k] \\ -f[m+1,n,k]+f[m,n+1,k+1]-f[m,n,k+1] \\ +f[m+1,n+1,k+1]-f[m+1,n,k+1] \end{Bmatrix} \quad (8)$$

Thus, the procedure to estimate motion using optical flow equation can be described as follows:

1. For each pixel at location [*m*, *n*] Є *B*, compute the gradients $\partial f/\partial x$ and $\partial f/\partial y$ using the approximations in Equations 7 and 8.
2. Compute the quantities of *M* and *b*.
3. Solve for the estimate $u = (\Delta x, \Delta y)^T$ from $u = M^{-1}b$.

However, another practicality worth mentioning is that some image smoothing is generally useful prior to numerical differentiation and can be incorporated into the derivative filters [8].

4. Proposed Technique

Generally, the method of sub-pixel block-based ME comprises of following steps:

1. Estimating a first integer-pixel accuracy MV, where it maps a reference block to a current block.
2. Computing image gradient information or interpolation of the reference block.
3. Determining a second MV to sub-pixel accuracy by adding a SPMV to the first MV, wherein the SPMV is estimated based on interpolation or the image gradient information.

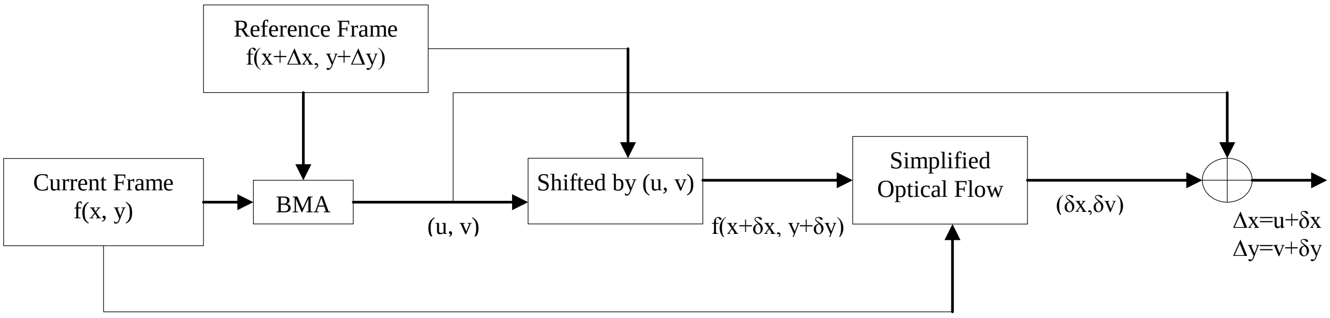


Figure 3. Block diagram of sub-pixel ME using a combination of BMAs and Taylor series approximation (simplified optical flow).

In this paper, the proposed ME technique is an amalgamation of BMA and the simplified optical flow (Taylor approximation). It is implemented in three steps as illustrated in figure 3. In the first step, we use a Block Matching Algorithm (BMA) to generate integer pixel displacements *u* and *v* (MVs). If (Δx, Δy) is the true displacement, then (u, v) determined by BMA should be a good integer estimate of (Δx, Δy). In the second step a simple shifting algorithm is used. While (*u*, *v*) is determined, we shift the image block by (*u*, *v*) pixels along the *x* and *y* directions respectively. Since, the shift is an integer factor, no interpolation is required. The third step of the algorithm is to use Taylor series approximation to refine the search. Since the shifted frame $f(x+\delta x, y+\delta y)$ be different from the true frame by only (δx, δy), with $|\delta x|<1$ and $|\delta y|<1$ pixels the Taylor series approximation algorithm is applicable.

Thus, the overall fractional-pixel accuracy displacement can be determined as:

$$\Delta x = u + \delta x \text{ and } \Delta y = v + \delta y \tag{9}$$

Here, the first step is implemented with FS, TSS, and bilateral ME BMAs. Therefore the involvement of this paper is to determine the SPMV using BMA algorithm with no interpolation requirement. However, SPMV that provided any BMA which uses interpolation for sub-pixel accuracy will increase the computation complexity for sub-pixel search level. Hence, the technique used in this paper can greatly enhance the speedup ratio in terms of time with insignificant effect on PSNR performance.

5. Error Analysis and Comparison

In order to compare the efficiency of the proposed algorithm with standard ME algorithms used in video compression, we present the following analytical error analyses which demonstrate the expediency of our algorithm. Therefore, we first confirm that the proposed technique can achieve lower absolute difference error than any classical BMAs.

Even if the algorithm is implemented in 2D, just for simplicity the following equations are derived in 1D and the derivations are also valid in 2D. Considering a 1-dimensional function of *f*(*x*) and *g*(*x*), the optimal displacement $\overline{\Delta x}$ is the solution of $\frac{d}{d\Delta x}E(\Delta x) = 0, where\ E(\Delta x) = \sum_x \left(g(x) - f(x) - f'(x)\Delta x\right)^2$ is assumed to be the sum square error. So $\overline{\Delta x}$ can be calculated as:

$$\overline{\Delta x} = \frac{\sum_x f'(x)[g(x) - f(x)]}{\sum_x [f'(x)]^2} \tag{10}$$

Suppose that, the factual displacement is given by Δx, and then the absolute difference error can be determined as:

$$\begin{aligned} \left|\overline{\Delta x} - \Delta x\right| &= \left|\frac{\sum_x f'(x)[g(x) - f(x)]}{\sum_x [f'(x)]^2} - \Delta x\right| \\ &= \left|\frac{\sum_x f'(x)[g(x) - f(x)] - \Delta x \sum_x [f'(x)]^2}{\sum_x [f'(x)]^2}\right| \\ &= \left|\frac{\sum_x f'(x)[g(x) - f(x) - \Delta x f'(x)]}{\sum_x [f'(x)]^2}\right| \end{aligned} \tag{11}$$

Now, applying the generalized mean value Theorem in Equation 11, which states that there exists ξ such that:

$$g(x) = f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2} f''(\xi)(\Delta x)^2 \qquad (12)$$

Equation 11 can rewrite as:

$$\left|\overline{\Delta x} - \Delta x\right| = \left| \frac{\sum_x f'(x)[\frac{1}{2} f''(\xi)(\Delta x)^2]}{\sum_x [f'(x)]^2} \right| \le \frac{1}{2} \max_{\xi} \left|f''(\xi)\right| \frac{\sum_x \left|f'(x)\right|}{\sum_x \left|f'(x)\right|^2} (\Delta x)^2 \qquad (13)$$

So as to bound the above expression.

1. We noted that *max* $f(x)=1$ and *min* $f(x)=0$. So *max* $f'(x)=1$, *min* $f'(x)=-1$, and hence *max* $|f''(x)|=2$.
2. Assuming FS is applied to determine the nearest integer at the first stage, then Δx is at most 0.5, thus $(\Delta x)^2 \le 0.25$.
3. Since moving objects usually show positive and negative gradients at front and tail moving edges respectively, the overall sum of these gradients is small, but the sum of squared gradients is large.
4. From our know-how, the typical value of $\frac{\sum_x |f'(x)|}{\sum_x |f'(x)|^2} < \frac{1}{20}$.

Therefore, putting these aforementioned together, the error is bound by $\left|\overline{\Delta x} - \Delta x\right|_{proposed} \le 0.0125$.

For comparison, considering a full search algorithm with the assumption that it rounds off the computed MV to its closest 1/8 fractional accuracy, the error bound is given by $\left|\overline{\Delta x} - \Delta x\right|_{FS} \le \left(\frac{1}{2}\right)\left(\frac{1}{8}\right) = 0.0625$. Hence, compared to the proposed technique, this error is much larger.

As a conclusion, since full search is more accurate (though slower) than most classical BMAs, this results implies that the proposed technique can find a more accurate MV than other methods and hence the PSNR increases.

4. Experimental Results and Discussion

First, to explore whether the proposed algorithm can detect the true MVs or not, we verify the above analytical error analysis results by experiments. We randomly generate true MVs and shift images according to these true MVs. To shift an image with fractional pixels, we first shift the image by the closest integer pixel, and use linear interpolation to approximate the value at that fractional pixel location. To mimic a real video sequence, Gaussian noise is also added. Figure 4 shows the sum square error of the estimated MV and the true MVs. As observed, the proposed technique gives considerably smaller sum square error at all scheduled noise variance levels.

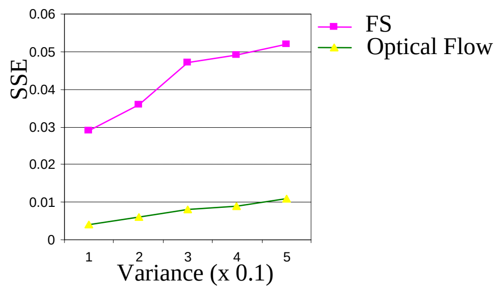


Figure 4. Sum square error vs noise variance levels.

The objective of the second simulation is to investigate whether that our algorithm can attain PSNRs as some of the existing algorithms. Here, we choose to contrast with full search because it is the most robust and most accurate BMA. If the proposed technique can achieve similar PSNR as that of FS with interpolation, while reducing its execution time, then the proposed technique can also achieve similar performance in conjunction with other ME methods. Thus, if one wishes to compare with a customized BMA, we can apply this ME for the first step to determine the integer-pixel MV, and use Taylor approximation for the second step to determine the SPMV. Provided that the original ME requires interpolation, the proposed technique can improve the efficiency.

The experimentations are performed on six standard video sequences with frame size of 288x352. The MVs are estimated using FS to 1/4-pixels accuracy and FS with Taylor approximation. Given the estimated MVs, the motion compensated frames are generated to compute the PSNR. Table 1 shows simulation of the average PSNR and computation times of six standard videos for FS to 1/4pixels accuracy and simplified optical flow technique. All simulations were done on Matlab-7.9 using a Pentium 4 desktop with 3.0GHz CPU and 1.0GMb of RAM. The experimental results show that the simplified optical flow technique can provide slightly higher PSNR and it further significantly reduces the computation time.

Table 1. Comparisons of average PSNR and computational time for full search to 1/4-pixels accuracy and simplified optical flow technique.

Video Sequences	Average Computation Time(Sec)		Average PSNR (dB)	
	FS to 1/4-Pixels Accuracy	Simplified Optical Flow Technique	FS to 1/4-Pixels Accuracy	Simplified Optical Flow Technique
Foreman	70.156	5.349	37.946	38.021
Football	59.909	4.569	33.576	33.672
Salesman	70.562	5.380	37.757	37.850
Missa	72.586	5.534	41.810	42.006
Susie	55.083	4.534	38.967	39.057
Trevor	47.492	3.621	38.285	38.477

Since, the MVs estimated by the proposed technique become more accurate, the PSNR of the motion compensated frames is comparable to the one performed by full search to 1/4-pixels accuracy as shown in Figure 5. It shows the experimental results for 12 sequences of Susie video. Here, the PSNR values are computed from the motion compensated version of the second frame of each sequence in order to evaluate the performance of the ME only. The results show that the proposed method provides slightly better. As seen the results of Susie video from Figure 6 the computation time of the proposed technique is appreciably shorter.

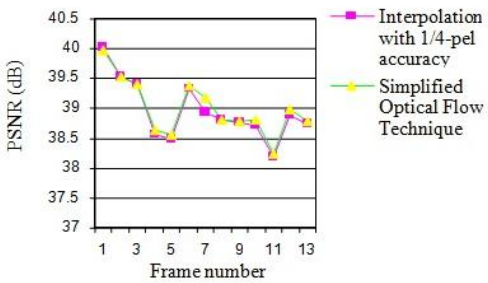


Figure 5. PSNR vs frame number experimental results for 12 sequences of Susie video sequences.

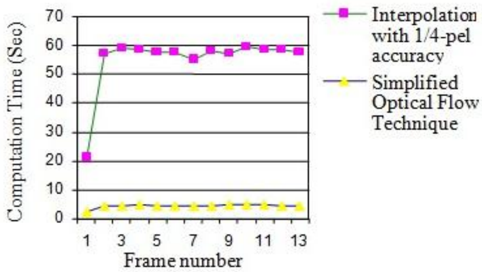


Figure 6. Computation time vs frame number experimental results for 12 sequences of Susie video sequences.

7. Conclusions

This paper proposes a fast and interpolation free sub-pixel ME algorithm that particularly aims at providing accurate MVs. First, in order to obtain the MV at integer-pixel level a conventional BMA is applied, and the result is customized based on motion information from adjacent pixels. Second, a simplified optical flow technique is applied to improve the MV to sub-pixel accuracy based on the modified reference and current frames information. Analytical and experimental results show that the proposed technique can provide more accurate MVs and reaches comparable PSNR performance as conventional 1/44pixels accuracy with significant saving on computation cost when compared to FS with interpolation. For further investigation, one can use the quad-tree partitioning of a frame, it provides a better level of adaptation to scene contents compared to fixed block size approaches. It can also be implemented for motion deblurring, where no interpolation is required.

References

[1] Ayele E., Chaudhari R., and Dhok S., "Fast Motion Estimation for Quad-Tree Based Video Coder Using Normalized Cross-Correlation Measure," *International Journal of Image processing*, vol. 7, no. 4, pp. 569-572, 2013.

[2] Barron J. and Thacker N., "Tutorial: Computing 2D and 3D Optical Flow," *Tina Memo*, no. 12, 2005.

[3] Chen Z., Zhou P., and He Y., "Fast Integer-Pixel and Fractional Pel Motion Estimation for JVT," *Tsinghua University*, 2002.

[4] Choi B., Han J., Kim C., and Ko S., "Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407-416, 2007.

[5] Fleet D. and Weiss Y., *Mathematical Models in Computer Vision: The Handbook*, Springer, 2005.

[6] Gharavi H. and Mills M., "Block-Matching Motion Estimation Algorithms: New Results," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 5, pp. 649-651, 1990.

[7] Girod B., "Motion-Compensating Prediction with Fractional-Peel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604-612, 1993.

[8] Horn B., *Robot Vision*, MIT Press, 1986.

[9] Horn B. and Schunck B., "Determining Optical Flow," *Artificial Intelligence*, vol. 17 no. 1-3, pp. 185-203, 1981.

[10] Irani M. and Peleg S., "Improving Resolution by Image Registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231-239, 1991.

[11] Jain J. and Jain A., "Displacement Measurement and its Application in Inter-Frame Image Coding", *IEEE Transactions on Communications*, vol. 29, no. 12, pp. 1799-1808, 1981.

[12] Li R., Zeng B., and Liou M., "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 4, no. 4, pp. 438-442, 1994.

[13] Li X., "Video Processing via Implicit and Mixture Motion Models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 8, pp. 953-963, 2007.

[14] Lim J., *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.

[15] Lin W., Panusopone K., Baylon D., Sun M., Chen Z., and Li H. "A Fast Sub-Pixel Motion Estimation Algorithm for H.264/AVC Video Coding," *IEEE Transactions on Circuits and*

Systems for Video Technology, vol. 21, no. 2, pp. 237-242, 2011.

[16] Lin W., Panusopone K., Baylon D., and Sun M., "A New Class Based Early Termination Method for Fast Motion Estimation in Video Coding," *in Proceeding of IEEE International Symposium Circuits and Systems*, Taipei, pp. 625-628, 2009.

[17] Lu J. and Liou M., "A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 7, no. 2, pp. 429-433, 1997.

[18] Lucas B. and Kanade T., "An Iterative Image Registration Technique with an Application in Stereo Vision," *in Proceeding of 7th International Joint Conference on Artificial Intelligence*, Vancouver, pp. 674-679, 1981.

[19] Nie Y. and Ma K., "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1442-1448, 2002.

[20] Po L. and Ma W., "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 6, no. 3, pp. 313-317, 1996.

[21] Suh J. and Jeong J., "Fast Sub-Pixel Motion Estimation Techniques Having Lower Computational Complexity," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 3, pp. 968-973, 2004.

[22] Tico M. and Vehvilainen M., "Constraint Motion Filtering for Video Stabilization," *in Proceeding of IEEE International Conference on Image Processing*, Genova, pp. 569-572, 2005.

[23] Tourapis H. and Tourapis A., "Fast Motion Estimation within the H.264 Codec," *in Proceeding of International Conference Multimedia Expo*, Baltimore, pp. 517-520, 2003.

[24] Trimeche M. and Yrjanainen J., *Visual Content Processing and Representation*, Springer-Verlag Berlin Heidelberg, 2003.

[25] Wang Y., Ostermann J., and Zhang Y., *Video Processing and Communications*, Prentice Hall, 2002.

[26] Werda I., Chaouch H., Samet A., Ayed M., and Mamoudi N., "Optimal DSP Based Integer Motion Estimation Implementation for H.264/AVC Baseline Encoder," *The International Arab Journal of Information Technology*, vol. 7, no. 1, pp. 96-104, 2010.

[27] Xu X. and He Y., "Improvements on Fast Motion Estimation Strategy for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 285-293, 2008.

[28] Zhou Z. and Sun M., "Fast Macro-Block Inter Mode Decision and Motion Estimation for H. 264/MPEG-4 AVC," *in Proceedings of International Conference on Image Processing*, Singapore, pp. 789-792, 2004.

[29] Zhu S. and Ma K., "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*; vol. 9, no. 2, pp. 287-290, 2000.

Eskinder Ayele received his BTech degree in electrical engineering from DEC, Ethiopia and MTech degree in Control and Instrumentation from IIT, Madras, India in 2001 and 2005 respectively. Currently, he is a PhD scholar in Centre for VLSI and Nanotechnology, VNIT, Nagpur, India. His research interests include Signal and Image Processing, Electronic Instrumentation, and Control Systems.

Sanjay Dhok is Associate Professor in Centre for VLSI and Nanotechnology at Visvesvaraya National Institute of Technology, Nagpur (India). He received his PhD in Electronics Engineering from VNIT Nagpur, India. He is a member of IEEE society. He has published many research papers in national and international journals and conferences. His area of interest includes Signal Processing, Image Processing, Data Compression, Wireless Sensor Networks and VLSI Design.