

SUBPIXEL MOTION ESTIMATION WITHOUT INTERPOLATION

Stanley H. Chan , Dũng T. Võ and Truong Q. Nguyen

ECE Dept, UCSD, La Jolla, CA 92092-0407

<http://videoprocessing.ucsd.edu>

Email: h5chan@ucsd.edu, d3vo@ucsd.edu, nguyent@ece.ucsd.edu

ABSTRACT

We propose a fast subpixel motion estimation method for motion deblurring, where conventional motion estimation algorithms used in video codings are too complex. The new algorithm is a combination of block matching and optical flow. It does not require any interpolation and it does not provide motion compensated frames. Thus it is much faster than conventional methods. Statistical results show that the new algorithm performs quickly and accurately. It also demonstrates compatible performance with the benchmarking full search algorithm, yet uses significantly less amount of time.

Index Terms— motion estimation, three step search, block matching, optical flow, motion deblurring.

1. INTRODUCTION

Motion estimation (ME) is the essential element in video processing. Especially in coding, the performance of motion estimation algorithms can directly affect the performance of the coding scheme. Therefore, despite its long history, motion estimation is still an important research topic for the video engineers. However, techniques for video coding is not always applicable to other applications. One of the examples is motion deblurring.

Motion blur appears in almost all practical video systems - either a moving camera with stationary scene or a stationary camera with moving scene (or both are moving, of course!). In order to restore a video, a good estimate of the motion blur point spread function (PSF), and hence the motion vector, is required. Now there are three issues. First, to generate a PSF, all we needed is the motion vector, whereas a motion compensated frame is irrelevant. This is quite different from video coding because for coding, motion compensated frames are necessary for prediction error computation. Therefore, almost all motion estimation algorithms in coding perform estimation and compensation simultaneously. Thus, if we use these methods for deblurring, then it will be wasteful because motion compensated frames are not needed.

The second issue is the repeated process of motion estimation. It has been a well known problem in image processing community that estimating a motion vector based on a pair of blurred images are seldomly accurate. Therefore, a more reasonable strategy in motion deblurring is to iteratively perform the motion estimation and deblurring alternatively - we first make an initial estimate of the motion vector, and refine it as we get sharper images from the deblurring procedure, and repeats. Since this would require repeated motion estimations, a one pass frame-to-frame motion estimation available from video coding is not suitable.

The third issue is subpixel interpolation. As we mentioned in the first issue, motion estimation algorithms in video coding perform estimation and compensation simultaneously. In the estimation part, the algorithm has to implicitly interpolate the image if subpixel accuracy is required. However the associated cost is higher for more precise motion vectors. For example, if we want to achieve a 0.125 pixel 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. Since we do not need motion compensated frames anyway, we investigate an algorithm that does not require any interpolation.

Because of these unique features, we propose a new hybrid motion estimation scheme. Our proposed method combines classical block matching and optical flow. It first determines a coarse motion vector using block matching, then refines itself using local approximations. This will be described in Section III.

Since the purpose of our algorithm is quite different from conventional video coding schemes, it is difficult to have a fair comparison with existing algorithms. However, in order to confirm and demonstrate the efficiency our algorithm, we will present several simulations results and discuss their implications. This will be discussed in Section IV.

2. CLASSICAL METHODS REVIEW

2.1. Block Matching Algorithms

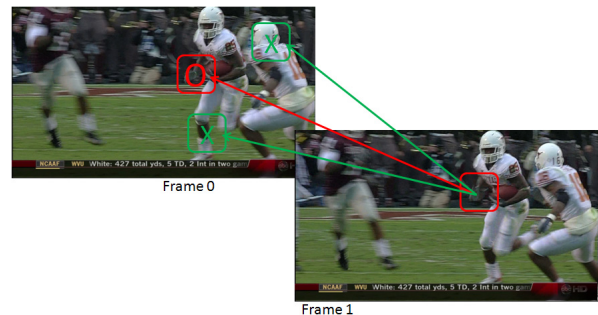


Fig. 1. Illustration of block matching methods.

Block matching method, as named, is a method that finds the best matched block in the search space. As shown in Fig. 1, if we want to find the best matching block in the frame 0, then one of the most straight forward methods is to search over all possible positions

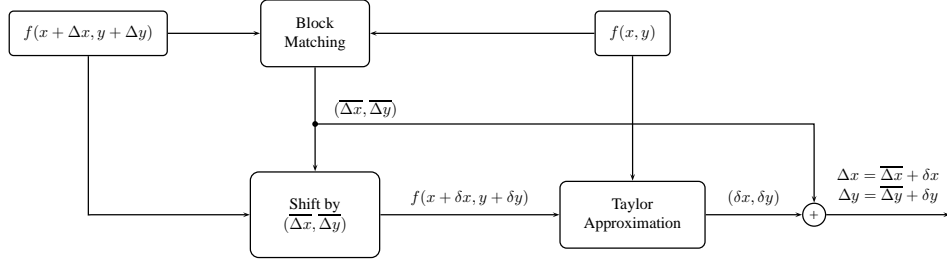


Fig. 2. Block diagram of motion estimation using a combination of block matching algorithms and Taylor approximation (simplified optical flow).

in the search range. Such exhaustive search is referred as full search [1].

To reduce the number of search points, we can use more advanced searching techniques, e.g. three step search [2] and bilateral ME [3]. Fig. 3 illustrates the concept of three step search [2] - at the first step, nine candidates centered at $(0, 0)$ with initial step size are tested to find the best candidate with the least error. At the second step, the center is moved to the best candidate and another eight candidates are picked with half of the previous step size. In the third step, the first and second step are repeated in many stages of picking the candidates until the step size meets the accuracy requirement of motion estimation.

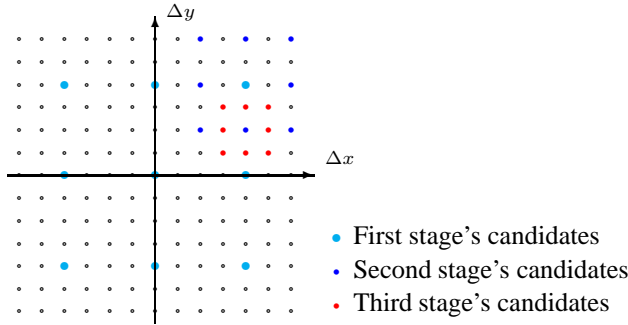


Fig. 3. An example of three step search scheme [2].

2.2. Optical flow

We describe a simplified version of classical optical flow [4]. Given two consecutive frames $f(x, y)$ and $g(x, y)$ we have [5]

$$g(x, y) = f(x + \Delta x, y + \Delta y) \approx f(x, y) + \Delta x \frac{\partial}{\partial x} f(x, y) + \Delta y \frac{\partial}{\partial y} f(x, y), \quad (1)$$

which is the first order Taylor series approximation.

The optimal shift can be found by solving the minimization problem

$$\underset{\Delta x, \Delta y}{\text{minimize}} \quad \Phi(\Delta x, \Delta y) \quad (2)$$

where

$$\Phi(\Delta x, \Delta y) = \sum_{x, y} \left(g(x, y) - f(x, y) - \Delta x \frac{\partial}{\partial x} f(x, y) - \Delta y \frac{\partial}{\partial y} f(x, y) \right)^2. \quad (3)$$

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. Thus we have

$$\frac{\partial \Phi}{\partial \Delta x} = 0 \quad \text{and} \quad \frac{\partial \Phi}{\partial \Delta y} = 0,$$

and consequently we can setup the following system of linear equations

$$\begin{pmatrix} \sum_{x, y} \left(\frac{\partial f}{\partial x} \right)^2 & \sum_{x, y} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \sum_{x, y} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} & \sum_{x, y} \left(\frac{\partial f}{\partial y} \right)^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \sum_{x, y} (g - f) \frac{\partial f}{\partial x} \\ \sum_{x, y} (g - f) \frac{\partial f}{\partial y} \end{pmatrix}. \quad (4)$$

Therefore, by solving this system of linear equations we can determine the optimal solution. Note that in order to make Taylor approximation valid, we implicitly assumed that $|\Delta x| \ll 1$ and $|\Delta y| \ll 1$. For the computation of partial derivatives, one can approximate them using finite difference, i.e. $\frac{\partial f}{\partial x} = f(x + 1, y) - f(x, y)$ and $\frac{\partial f}{\partial y} = f(x, y + 1) - f(x, y)$.

3. PROPOSED METHOD

Our proposed motion estimation method is a combination of block matching algorithm and the simplified optical flow (See Fig. 2). In the first step, we use a block matching algorithm to determine integer pixel displacements $\overline{\Delta x}$ and $\overline{\Delta y}$. If $(\Delta x, \Delta y)$ is the true displacement, then $(\overline{\Delta x}, \overline{\Delta y})$ determined by block matching algorithm should be a good integer estimate of $(\Delta x, \Delta y)$.

When $(\overline{\Delta x}, \overline{\Delta y})$ is determined, we shift the image block by $\overline{\Delta x}$ pixels along x direction, and $\overline{\Delta y}$ pixels along y direction. Since the shift is an integer factor, no interpolation is needed.

The second step of the algorithm is to use Taylor series approximation to refine the search. Since the shifted image $f(x + \delta x, y + \delta y)$ differs from the true image by only $(\delta x, \delta y)$, where $|\delta x| < 1$ and $|\delta y| < 1$, the Taylor series approximation is approximately valid. So the overall displacement can be determined as

$$\Delta x = \overline{\Delta x} + \delta x \quad \text{and} \quad \Delta y = \overline{\Delta y} + \delta y \quad (5)$$

Note that the first step can be implemented with *any* block matching algorithm, for example full search, three step search, phase plane correlation, or bilateral ME. Therefore the contribution of this paper is that provided any block matching algorithm which uses interpolation for subpixel accuracy, we can determine the subpixel motion vector using the same algorithm without interpolation. Hence the speed can be greatly enhanced.

4. ANALYSIS AND DISCUSSION

As we stated in the introduction, our algorithm is not designed for video coding. So comparing with algorithms used in video compression/ video transmission may not reflect the efficiency of our algorithm. Therefore, we present a few analytical and experimental results to prove the usefulness of our algorithm.

4.1. Analytical Error Analysis

In order to analyze the performance of the proposed algorithm, we first prove that the proposed method can achieve lower error than any classical block matching algorithms. The following equations are derived in 1-dimension, but the derivations are also valid in 2D.

In 1-dimension, the optimal displacement $\widehat{\Delta x}$ is the solution of $\frac{d}{d\Delta x}\Phi(\Delta x) = 0$, where $\Phi(\Delta x) = \sum_x (g(x) - f(x) - f'(x)\Delta x)^2$. So $\widehat{\Delta x}$ can be found as

$$\widehat{\Delta x} = \frac{\sum_x f'(x)[g(x) - f(x)]}{\sum_x [f'(x)]^2}. \quad (6)$$

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

$$\begin{aligned} |\widehat{\Delta x} - \Delta x| &= \left| \frac{\sum_x f'(x)[g(x) - f(x)]}{\sum_x [f'(x)]^2} - \Delta x \right| \\ &= \left| \frac{\sum_x f'(x)[g(x) - f(x)] - \Delta x \sum_x [f'(x)]^2}{\sum_x [f'(x)]^2} \right| \\ &= \left| \frac{\sum_x f'(x)[g(x) - f(x) - \Delta x f'(x)]}{\sum_x [f'(x)]^2} \right| \\ &= \left| \frac{\sum_x f'(x)[\frac{1}{2}f''(\xi)(\Delta x)^2]}{\sum_x [f'(x)]^2} \right| \\ &\leq \frac{1}{2} \max_{\xi} |f''(\xi)| \frac{\sum_x |f'(x)|}{\sum_x [f'(x)]^2} (\Delta x)^2, \end{aligned} \quad (7)$$

where the Generalized Mean Value Theorem is applied in the fourth line, which states that there exists ξ such that

$$g(x) = f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2} f''(\xi)(\Delta x)^2. \quad (8)$$

The next step is to bound the above expression. First, note that $\max f(x) = 1$ and $\min f(x) = 0$. So $\max f'(x) = 1$, $\min f'(x) = -1$, and hence $\max |f''(x)| = 2$. Next, suppose full search is applied to determine the nearest integer at the first stage, then Δx is at most 0.5, thus $(\Delta x)^2 \leq 0.25$. Furthermore, moving objects usually show positive gradient at front moving edges, and negative gradient at tail moving edges. Therefore, the overall sum of these positive and negative gradients is small, but the sum of squared gradients is large. From our experience, the typical value of $\frac{\sum_x |f'(x)|}{\sum_x [f'(x)]^2}$ is less than $\frac{1}{20}$. Therefore, putting these together, the error is bound by $|\widehat{\Delta x} - \Delta x|_{\text{proposed}} \leq 0.0125$.

To compare with full search, note that full search rounds off the computed motion vector to its closest 1/8 fraction, so the error bound is given by $|\widehat{\Delta x} - \Delta x|_{\text{full}} \leq (\frac{1}{2}) (\frac{1}{8}) = 0.0625$. Compared to the proposed method, this error is much larger. Since full search is more accurate (though slower) than most classical block matching algorithms, this results implies that the proposed method can find a more accurate motion vector than other methods.

4.2. Experimental Error Analysis

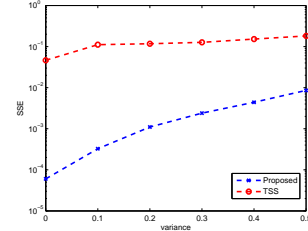


Fig. 4. Sum square error of the motion vectors compared to the true motion.

In this section we verify the above analytical results by experiments. We randomly generate 1000 true motion vectors and shift 1000 images according to these true motion vectors.¹ Gaussian noise is also added to imitate a real video sequence. The objective of this experiment is to investigate whether the proposed algorithm can detect the true motion vectors.

Fig. 4 shows the sum square error of the estimated motion vector and the true motion vectors. The block matching algorithm for this experiment is three step search (TSS). As shown, the proposed method gives significantly smaller error at all listed noise variance levels. Nevertheless, the average computing time for TSS is 1.4112 seconds (based on 1/8 pixel accuracy), whereas the proposed method is only 0.0538 seconds.²

4.3. Motion Compensation Comparison

The third comparison is to show that our algorithm can achieve PSNRs as some of the existing algorithms. Here we choose to compare with full search because (1) the proposed method works for any block matching algorithm. Thus if one wishes to compare with a customized block matching ME, we can implement this ME for the first step to determine the integer pixel motion, and use Taylor approximation for the second step to determine the subpixel motion. As long as the original ME requires interpolation, the proposed method can improve the efficiency. In other words it is independent of the original ME; (2) Full search is the most robust and most accurate block matching algorithm. If the proposed method can achieve similar PSNR as that of full search, while reducing its implementation time, then the proposed method can also achieve similar performance in conjunction with other ME methods.

We tested the algorithms on three real video sequences. Let adjacent frames be denoted by $f(x, y)$ and $g(x, y)$ respectively, then using the algorithm we can reconstruct a motion compensated frame $\tilde{f}(x, y)$. The residual between the $f(x, y)$ and $\tilde{f}(x, y)$ will be used to compute the PSNR, which is defined to be $PSNR = 10 \log_{10} (\frac{1}{MSE})$, where $MSE = \frac{1}{MN} \sum_{y=1}^N \sum_{x=1}^M (f(x, y) - \tilde{f}(x, y))^2$.

¹To shift an image with fractional pixels e.g. 3.124 pixels, we first shift the image by the closest integer pixel (e.g. 3 pixels), and use linear interpolation to approximate the value at that fractional pixel location (e.g. 0.124 pixels).

²More studies about the speed can be found at <http://videoprocessing.ucsd.edu/~stanleychan>

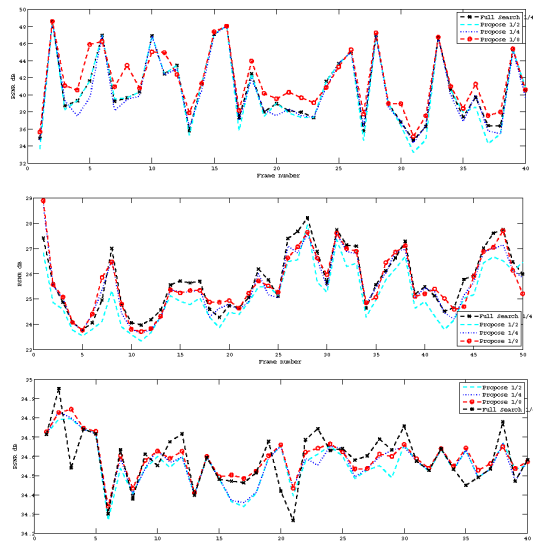


Fig. 5. PSNR curves of three video sequences. (Top) Train. (Middle) New York City. (Bottom) Stockholm. The average computing time for proposed method is 0.5 seconds, whereas the full search is 15 seconds.



Fig. 6. Sample frame of testing video sequences. (Left) Train. (Middle) New York City. (Right) Stockholm. Size: 280×280

The motion vectors are estimated using full search (to 1/4 pixels) and full search with Taylor approximation (truncated to 1/2, 1/4 and 1/8 pixels, respectively). Given the estimated motion vectors, the motion compensated frames are generated using standard H.264 interpolation filters.

The goal of this simulation is to show that (1) when motion vectors estimated by the proposed method becomes more accurate, the PSNR increases; (2) at all three subpixel accuracy levels (1/2, 1/4, and 1/8), the PSNR of the motion compensated frames using proposed method is comparable to the one performed by full search; (3) the computation time of the proposed method is significantly shorter. Fig. 5 summarizes the results.

4.4. Motion Deblurring Results

Last, we apply the proposed method to LCD deblurring [6]. LCD deblurring is a particular motion blur problem where an overdriven signal has to be inversely synthesized so that it compensates the slow response (hence temporal blur) of the LCD. To solve this problem, the most important step is to determine the motion and form the point spread functions at each block of the image. This can be done using any existing ME algorithms, but these are usually slow as they require interpolations. So the proposed method is used to find the motion vectors.

After estimating the point spread functions we solve a least square problem of minimize $\|Ax - b\|_2^2$, where A is a shift varying convolution matrix characterized by the point spread functions, x is the unknown vector, and b is the target sharp image stacked in lexicographic order (See [6] for details).

Fig. 7(left) shows the perceived signal using the original image as the input to the LCD, and Fig. 7(right) shows the perceived signal using the synthesized overdriven signal. The motion vectors are computed using full search with Taylor approximation.

384 frames	384 frames
192 frames	192 frames
96 frames	96 frames
64 frames	64 frames
32 frames	32 frames

Fig. 7. LCD deblurring results. (Left) The perceived signal using original input. (Right) The perceived signal using synthesized signal, where motion vectors are determined by full search with Taylor approximation.

5. CONCLUSION

In this paper we propose a motion estimation method that only aims at providing motion vectors. A particular application of the proposed method is for motion deblurring, where no interpolation is needed. Since no interpolation is needed, the method is tremendously faster than any exiting block matching algorithms. Analytical and experimental results show that the method can provide more accurate motion vectors and comparable motion compensated frames when compared to full search.

6. REFERENCES

- [1] Yao Wang, Jorn Ostermann, and Ya-Qin Zhang, *Video Processing and Communications*, Prentice Hall, 2002.
- [2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *IEEE NTC*, 1981, pp. 531 – 534.
- [3] B. Choi, J. Han, C. Kim, and S. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 4, pp. 407–416, 2007.
- [4] Michal Irani and Shmuel Peleg, "Improving resolution by image registration," in *CVGIP: Graphical Models and Image Processing*, 1993, vol. 53, pp. 324–335.
- [5] Xin Li, "Video processing via implicit and mixture motion models," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 17, no. 8, pp. 953–963, 2007.
- [6] Stanley Chan and Truong Q. Nguyen, "Fast LCD motion deblurring by decimation and optimization," in *Proceedings of ICASSP 2009*, pp. 1201–1204.