Subpixel Motion Estimation without Interpolation with Two Blocking Methods(Full Search, Log Search)

Ji Dung, Lo, Rui Qi, Wang

Abstract

In our report, we will first review a new method for fast subpixel motion estimation focusing on motion deblurring, compared to traditional motion estimation algorithms which are often complex and require interpolation, the proposed method combines block matching and simplified version of optical flow. It provides a speed advantage without the need for motion-compensated frames and achieves comparable accuracy to more computationally intensive methods. Secondly, we will compare the proposed method with 2 classical methods, which are Full Search and Log Search. Lastly, we will analyze each method's performance through MSE, PSNR, and run time based on the experimental results to demonstrate its effectiveness.

1. Literature Review

1.1. Introduction

Accurately estimating the PSF is essential for reducing motion blur and restoring image clarity. Without a good estimation, it will be challenging to get closer to the true object. In this section, we are going to introduce PSF and detail why motion estimation is important.

Motion blur is a common occurrence in most real-world video systems. To restore a video effectively, it is essential to obtain an accurate estimate of the motion blur point spread function (PSF), which in turn requires determining the motion vector. As a result, motion estimation plays an important role in modeling blur kernel (PSF). The PSF describes how pixel intensities spread due to motion blur. Without motion estimation, we wouldn't know the blur direction, shape, or magnitude. The motion blurred image formation process is shown in Fig 1 and can be mathematically represented by a convolution equation:

$$g(x,y) = f(x,y) * PSF(x,y)$$

where $*$ denotes convolution, f(x,y) is the original image, and g(x,y) is the blurred image.

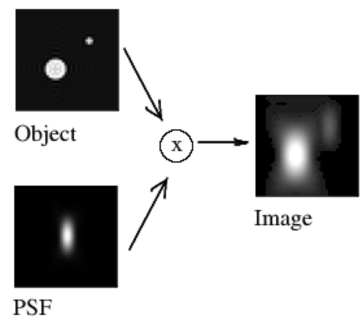

Fig. 1. Illustration of PSF

To deblur an image, we take the acquired image and PSF to perform deconvolution as illustrated in Fig 2. Therefore, we conclude that the more accurate the estimated PSF is, the closer we get to the true object during deconvolution.

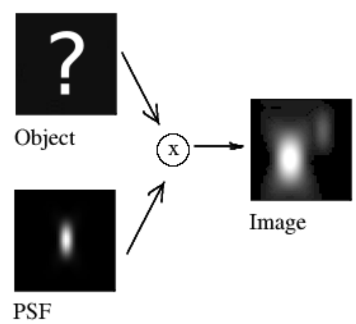

Fig. 2. Deconvolution using acquired image and PSF to find the true object

1.2. Problem Statement

In this section, we will explore two issues with traditional motion estimation methods designed for motion deblur. First, traditional motion estimation tends to be complex and wasteful for motion deblurring because motion compensation is irreverent to estimation of motion blur point. PSF does not need to predict future frames like in video compression and only estimates how pixel intensities spread due to motion within a single frame.Thus, motion compensated frames are not needed because we focus on image deblurring, not video compression.

Second, non-iterative motion estimation, commonly used in video compression, is ineffective due to no refinement being performed after the initial estimation. A more effective approach is an iterative process, where motion estimation and deblurring refine each other. An initial motion estimate is improved as deblurring sharpens the image, leading to progressively better motion vectors and a clearer result.

Third, achieving subpixel accuracy in motion estimation typically requires image interpolation, which accordingly increases computational cost. Since motion-compensated frames are not needed, the new algorithm avoids interpolation entirely, thereby reducing complexity.

Thus, the author proposed a new hybrid algorithm combining block matching algorithm and optical flow, which does not require interpolation. The method first determines a coarse motion vector using block matching algorithm, and refines itself using local approximation. It can be implemented with any block matching algorithm without interpolation. The experimental result from the author shows that the new method can find a more accurate motion vector than other methods at faster speed. We will detail our result in section 4.

1.3. Classical Methods Review

1.3.1 Block Matching Algorithms

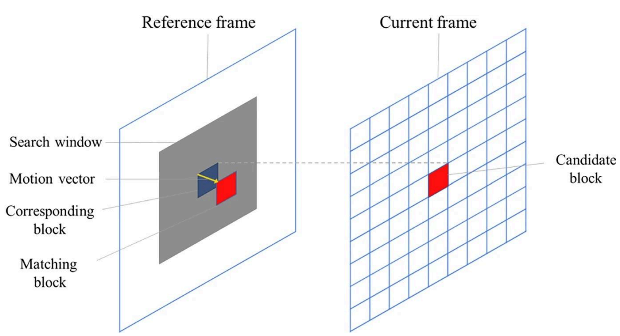

Fig. 3. Illustration of block matching algorithm

Block matching is one of the go-to techniques for motion estimation. It works by dividing an image into blocks and searching for the best-matching block in a reference frame. As shown in Fig 3, if we want to find the best matching block in the reference frame, full search has to search all the possible positions in the search range, which is computationally expensive. To reduce the number of search points, we can use fast search methods, such as 2-D logarithmic search.

Optical flow

The author used a simplified version of the classical optical flow. The mathematical formation equations are comprehensively explained in [1], we do not probe into the details of the process. In short, given two consecutive frames, we can express as:

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

Using the first-order Taylor series approximation, it approximates:

$$g(x, y) \approx f(x, y) + \Delta x\,(\partial f/\partial x) + \Delta y(\,\partial f/\partial y)$$

To determine the optimal displacement (Δx,Δy), we minimize the objective function:

$$\Phi(\Delta x, \Delta y) = \sum(g(x, y) - f(x, y) - \Delta x(\partial f/\partial x) - \Delta y(\partial f/\partial y))^2$$

Solving $\frac{\partial \Phi}{\partial \Delta x} = 0$ and $\frac{\partial \Phi}{\partial \Delta y} = 0$ yields a system of linear equations, solved via least squares. This method is effective for small shifts, using finite differences for efficiency without interpolation.

2. Study of Proposed Method

While being effective, block matching methods often generate motion-compensated frames, which isn't necessary for deblurring. In addition, to achieve subpixel accuracy, interpolation is required and thereby increase the computational cost. Simplified optical flow can make up for this subpixel accuracy by merely taking the first order of Taylor expansion since the pixel displacement at refinement step is smaller than 1, which significantly decreases the computational cost.

The new method aims to combine both as shown in Fig 4. The proposed fast subpixel motion estimation method follows this idea, integrating block matching and optical flow to create a more efficient and accurate solution for motion deblurring. Here's how it works:

Step 1: **Coarse Motion Estimation with Block Matching**. This quickly determines integer pixel displacements $\Delta\bar{x}$ and $\Delta\bar{y}$. We then shift the image block by $\Delta\bar{x}$ and $\Delta\bar{y}$. Since the shift is an integer factor, no interpolation is needed.

Step 2: **Refinement the search with Taylor series Approximation**. Fine-tunes the motion vector with Taylor series approximation. 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 only needs to take the first order.

Hence, the overall displacement can be determined as follow:

$$\Delta x = \Delta\bar{x} + \delta x \quad and\ \Delta y = \Delta\bar{y} + \delta y$$

where the first step can be implemented with any block matching algorithm, such as full search and log search. Unlike traditional methods, this approach skips the expensive interpolation step, significantly speeding things up.

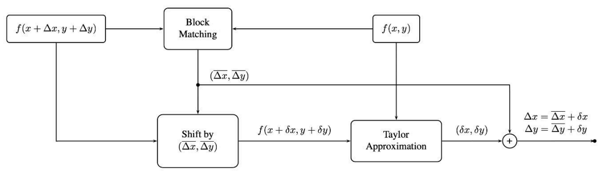

Fig. 4. Combination of block matching and Taylor expansion(Simplified optical flow)

3. Research Design, Implementation, Evaluation

To evaluate the effectiveness of this hybrid method, we will compare it against classical motion estimation algorithms with interpolation. The comparison will focus on computational efficiency and accuracy in estimating motion vectors.

Implementation Plan:

1. **Implementation of Classical Methods**. Implement 1 block matching motion estimation: full search with interpolation as baseline methods.
2. **New proposed Method Implementation**. Code the proposed approach that combines block matching and optical flow refinement.
3. **Performance Metrics**. Measure execution time, accuracy of estimated motion vectors (MSE and PSNR).
4. **Experimental Comparison**. Run all methods on the dataset and compare their outputs.
5. **Analysis and Visualization**. Present results through performance graphs and visual comparisons of motion vector accuracy.

4. Experimental Results

4.1. Full Search with Linear Interpolation

In the traditional method, we first use full search to find the coarse pair of integer motion vectors, and then refine the subpixel accuracy with linear interpolation (see Fig 5). We evaluate the result in Fig 18. with the run time in motion estimation and motion compensation, and compute the MSE (Mean Squared Error) and PSNR (Peak Signal-to-Noise Ratio).

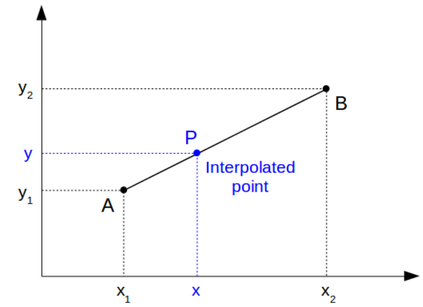

Fig. 5. Illustration of linear interpolation

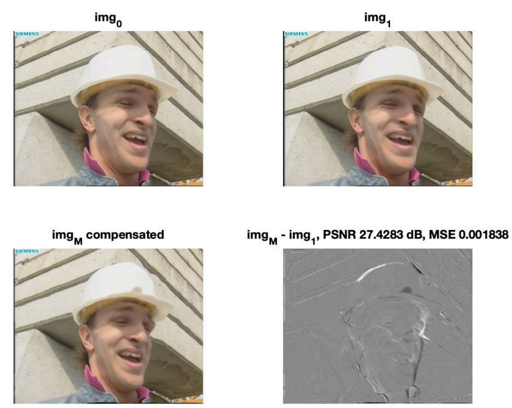

Fig. 6. Result of Full Search with linear interpolation

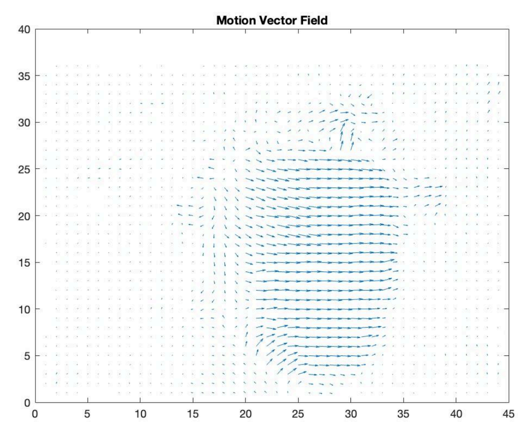

Fig. 7. Motion vector field of Full Search with linear interpolation

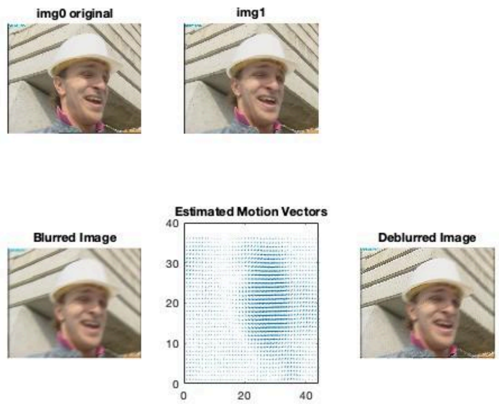

Fig. 8. Deblurred image using Full Search with interpolation

4.2. Full Search with Taylor Series Approximation

In the proposed method, we again use full search to find the coarse pair of integer motion vectors, and then refine the subpixel accuracy with Taylor Approximation. We evaluate the result in Fig 18 with run time in motion estimation and motion compensation, and compute the MSE (Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio).

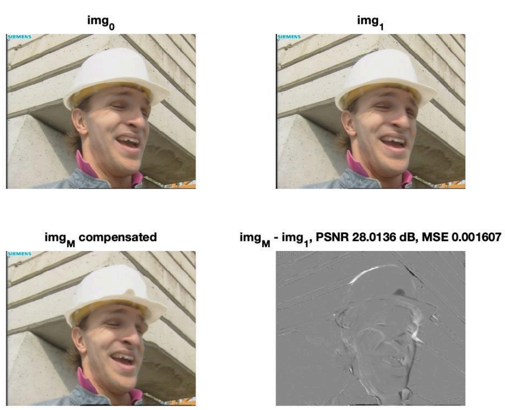

Fig 9. Result of Full Search with Taylor Approximation

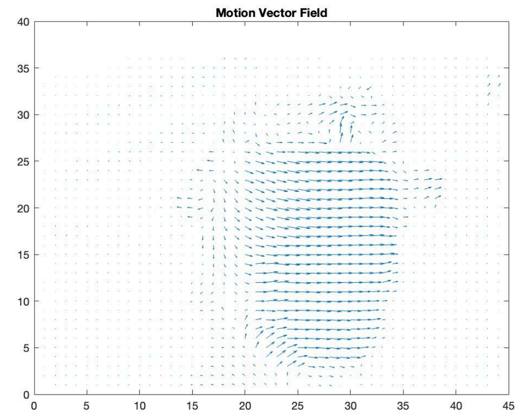

Fig 10. Motion vector field of Full Search with Taylor Approximation

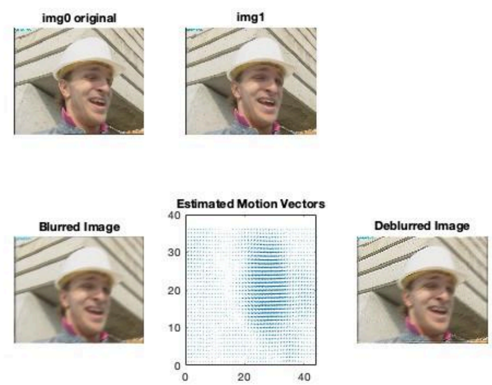

Fig. 11. Deblurred image using Full Search with Taylor Approximation

4.3. LogSearch with Linear Interpolation

We first use Log search to find the coarse pair of integer motion vectors, and then refine the subpixel accuracy with linear interpolation (see Fig 5). We evaluate the result in Fig 18 with the run time in motion estimation and motion compensation, and compute the MSE (Mean Squared Error) and PSNR (Peak Signal-to-Noise Ratio).

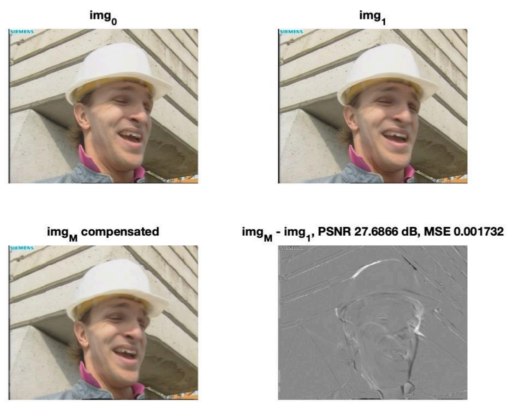

Fig. 12. Result of LogSearch with linear interpolation

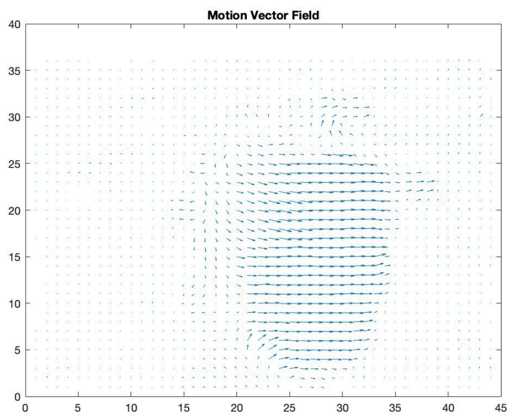

Fig 13. *Motion vector field of LogSearch with linear interpolation refinement*

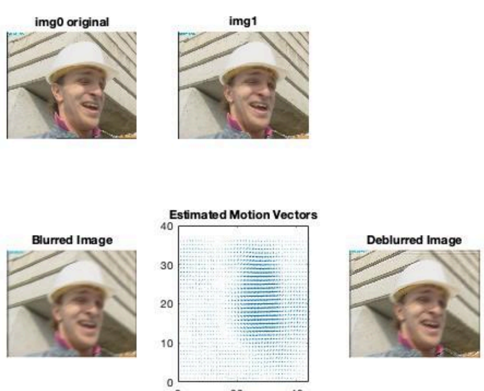

Fig. 14. Deblurred image using LogSearch with interpolation

4.4. LogSearch with Taylor Approximation

In the proposed method, we again use Log search to find the coarse pair of integer motion vectors, and then refine the subpixel accuracy with Taylor Approximation. We evaluate the result in Fig 18 with run time in motion estimation and motion compensation, and compute the MSE (Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio).

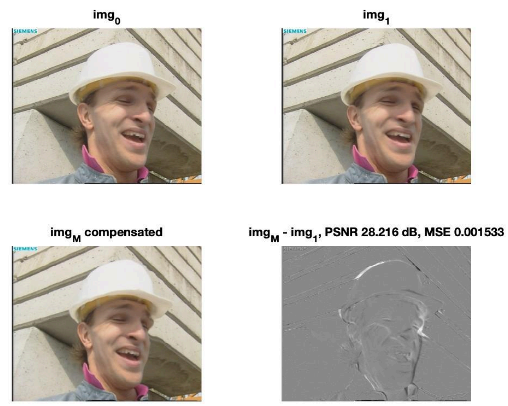

Fig. 15. *Result of LogSearch with Taylor Approximation*

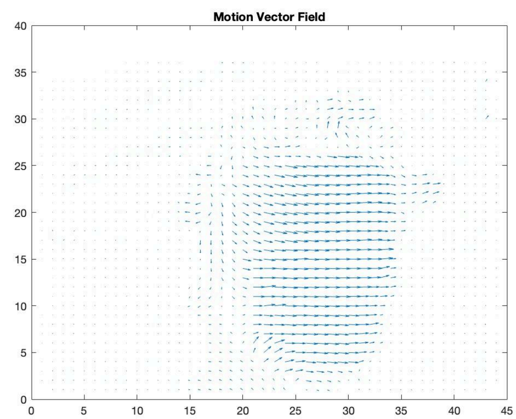

Fig 16. Motion vector field of LogSearch with Taylor Approximation

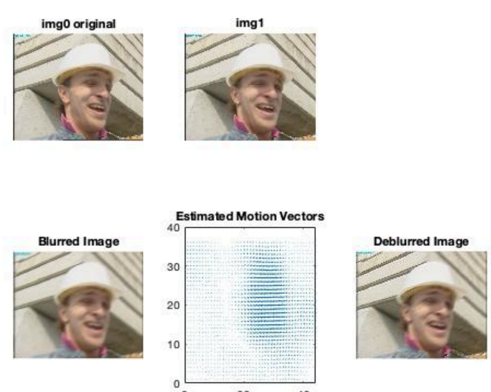

Fig. 17. Deblurred image using LogSearch with Taylor Approximation

4.5. Result Comparison

	PSNR (dB)	MSE	Motion Estimation Run Time (sec)	Motion Compensation Run Time (sec)
FS-Interpolation	27.4283	0.001838	1.293936	0.027356
FS-Taylor	28.0136	0.001607	2.13767	0.029503
LogSearch-Interpolation	27.6866	0.001732	0.385722	0.026424
LogSearch-Taylor	28.216	0.001533	1.060385	0.032306

Fig 18. Comparison result of 4 different methods

Our experiment results show that Log Search is more efficient than Full Search (FS) while maintaining comparable PSNR. Among all methods, LogSearch-Taylor achieves the best PSNR (28.216 dB) and lowest MSE (0.001533), offering the highest motion estimation accuracy while being nearly twice as fast as FullSearch-Taylor. In contrast, FullSearch methods are computationally expensive, making them impractical for real-time applications despite their high PSNR.

For applications requiring a balance between speed and PSNR, LogSearch-Taylor is the optimal choice, offering high PSNR with minimum computational cost. If efficiency is the primary concern, LogSearch-Interpolation is the fastest method, achieving not too bad PSNR but very fast runtime compared to FS methods. Overall, LogSearch proves to be a more practical approach as the first step compared to FullSearch, particularly when computational efficiency is a priority. The code used to compute our results is available at https://drive.google.com/drive/folders/1mxlky0FqXftOxTltOALglIFeF007AL7B?usp=share_link.

5. References

[1] Chan, Stanley H., Dũng T. Võ, and Truong Q. Nguyen. "Subpixel motion estimation without interpolation." *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010.https://ieeexplore.ieee.org/abstract/document/5495054

[2] Point Spread Function (PSF) | Scientific Volume Imaging. (n.d.). Scientific Volume Imaging. https://svi.nl/Point-Spread-Function-%28PSF%29#contentimaging_depth-1

[3] Mirjalili, M., & Mousavinia, A. (2024). Efficient block matching motion estimation using Variable-Size blocks and predictive tools. Circuits Systems and Signal Processing, 43(11), 7224–7248. https://doi.org/10.1007/s00034-024-02790-3

[4] x-engineer.org. (n.d.). Linear interpolation and extrapolation with calculator – x-engineer.org. https://x-engineer.org/linear-interpolation-extrapolation-calculator/