The Study of Phase-Based Optical Flow Technique Using an Adaptive Bilateral Filter

Ju Hwan LEE†, Sung Yun PARK†, Sung Jae KIM†, Nonmembers, and Sung Min KIM†, Member

SUMMARY The purpose of this study is to propose an advanced phase-based optical flow method with improved tracking accuracy for motion flow. The proposed method is mainly based on adaptive bilateral filtering (ABF) and Gabor based spatial filtering. ABF aims to preserve the maximum boundary information of the original image, while the spatial filtering aims to accurately compute the local variations. Our method tracks the optical flow in three stages. Firstly, the input images are filtered by using ABF and a spatial filter to remove noises and to preserve the maximum contour information. The component velocities are then computed based on the phase gradient of each pixel. Secondly, irregular pixels are eliminated by using reliability verification, if the phase differences are not linear over the image frames. Lastly, the entire velocity is derived by integrating the component velocities of each pixel. In order to evaluate the tracking accuracy of the proposed method, we have examined its performance for synthetic and realistic images for which the ground truth data were known. As a result, it was observed that the proposed technique offers higher accuracy than the existing optical flow methods.

**key words:** optical flow, phase gradient, adaptive bilateral filter, tracking accuracy

1. Introduction

Motion analysis that aims to analyze sequential images has been one of the most active research areas in image processing and computer vision [1]–[3]. Motion is regarded as a fundamental component in obtaining information from an image, such as the shapes and structures of moving objects. Using the motion information from an image sequence, it is possible to derive only the optical flow, which is an estimate of the variation in brightness patterns [4]–[6]. Since the optical flow technique was first introduced by Horn & Schunck [7] and Lucas & Kanade [8] in 1981, it has become a very useful method for analyzing sequential images in medical imaging and vehicle recognition.

Optical flow techniques are generally subdivided into four different methods, namely differential, region-based, energy-based and phase-based methods [9], [10]. In a review paper, Barron et al. have tested these algorithms on several image sequences, both synthetic and realistic, for which the ground truth data were known. One of their conclusions was that the phase-based and differential techniques showed more accurate performances overall [9], [10]. Phase-based optical flow techniques track moving objects according to the variances of phase information from the local areas [11], [12]. Fleet and Jepson adopted a spatiotemporal velocity tuned filter to extract the phase information. They confirmed that constant phase information provides a better approximation to the local velocity than constant amplitude [11]. They also showed that phase contours are more robust with regard to smooth shading and light variations, and are more stable with the small variations of moving objects [13]. Previous techniques, however, revealed tracking instability from local areas due to the problem of aperture interruption. Gautama et al. introduced a different approach to the estimation of the optical flow, which is based on the spatial filtering method to solve the aperture problem [14]. They presented a more accurate performance by using this spatial filtering method. It was however observed that the spatial filtering yields low tracking accuracy for the contour areas, as the boundary information is not preserved correctly.

In this paper, we present an advanced phase-based optical flow tracking method to solve the pre-mentioned problems and improve the tracking accuracy. The proposed technique adopts adaptive bilateral filtering (ABF) and Gabor based spatial filtering instead of the spatiotemporal filtering method as was used by Fleet and Jepson [11]. In order to evaluate the tracking accuracy of our method, we examined its performance for synthetic and realistic images for which the ground truth data were known. We also tested existing image filtering methods to evaluate whether ABF increases the tracking capacity. The rest of the paper is organized as follows: Section 2 outlines the proposed tracking technique and quantitative analysis method. Section 3 gives the tracking results derived from the developed method. We then assess the tracking accuracy by analyzing the results quantitatively in Sect. 4. Finally, we conclude our findings in Sect. 5.

2. Methods

In this paper, we adopted ABF and Gabor based spatial filtering to improve the tracking accuracy of the phase-based optical flow method. The proposed technique tracks the optical flow in three stages. In the first stage, the input images are filtered by using ABF and a spatial filter to remove noises and to preserve the maximum contour information. The component velocities are then computed based on the phase gradient of each pixel. In the second stage, irregular pixels are eliminated by using reliability verification, if
the derived phase differences are not linear over the image frames. Finally, the entire velocity is derived by integrating the component velocities of each pixel.

2.1 Adaptive Bilateral Filter

Traditional linear spatial filters usually handle noise by smoothing the original signal. These filters are appropriate for low signal variations, whereas unexpectedly blurring occurs when the signal variation is relatively high [15]. The bilateral filter is the one of the most suitable solutions to cope this problem [16]. Bilateral filtering is defined as a non-linear filtering technique which utilizes both spatial and amplitudinal distances to better preserve the boundary information [15], [17], [18].

In this study, the ABF described previously was carried out using a modified version of Wong’s method [15]. This filter is able to improve the performance of the noise suppression as well as preserving contours even in low signal to noise ratio conditions. In addition, ABF is rarely influenced by magnitude variations caused by illumination changes in the original image.

Using the bilateral filter, the brightness value of each pixel in an image is replaced by a weighted average of noise ratio conditions. In addition, ABF is rarely influenced by magnitude variations caused by illumination changes in the original image.

Using the bilateral filter, the brightness value of each pixel in an image is replaced by a weighted average of brightness values from neighboring pixels. The bilateral filter performs filtering by combining domain and range filtering, in a similar process to the domain filtering method.

Firstly, a low-pass domain filter applied to input image \( f(x) \) produces an output image defined by the following Eq. (1):

\[
h(x) = k_d^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi)c(\xi, x)d\xi
\]

where \( c(\xi, x) \) indicates the geometric closeness between the neighborhood center \( x \) and a nearby point \( \xi \), and \( f \) and \( h \) represent input and output signals, respectively. If low-pass filtering preserves the mean signals (DC component), then Eq. (1) can be generalized as Eq. (2).

\[
k_d(x) = \int_{-\infty}^{\infty} c(\xi, x)d\xi
\]

If the filter is shift-invariant, \( c(\xi, x) \) is regarded as a function of the vector difference between \( \xi \) and \( x \), and \( k_d \) is constant. Range filtering can be similarly defined by the Eq. (3)

\[
h(x) = k_c^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi)s(f(\xi), x)d\xi
\]

where \( s(f(\xi), x) \) represents the photometric similarity between the pixel at the neighborhood center \( x \) and that at a nearby point \( \xi \). Thus, the similarity function \( s \) operates in the range of the image function, whereas the closeness function \( c \) operates in the domain of \( f \).

\[
k_c(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(f(\xi), x)d\xi
\]

The appropriate solution for a filter that considers both domain and range is to combine these filters, thereby enforcing both geometric closeness and photometric similarity [17]. A bilateral filter is represented by this combined form of domain and range filtering, as described in Eq. (5) and generalized in Eq. (6):

\[
h(x) = k_d^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi)c(\xi, x)s(f(\xi), f(x))d\xi
\]

\[
k(x) = \int_{-\infty}^{\infty} c(\xi, x)s(f(\xi), f(x))d\xi
\]

The bilateral filter, in this way, can be denoted as a combined form of domain and range filtering. It replaces the pixel value at \( x \) with an average of similar and nearby pixel values. This ensures that averaging is performed along the boundary area and is significantly decreased in the gradient direction. For this reason, the bilateral filter can suppress the noise while at the same time preserving contour areas [18]. Consequently, the bilateral filter performs as a standard domain filter, and eliminates the weakly correlated differences between pixel values caused by noise.

The output image of the ABF is optimized by spatial filtering. We used Gabor based spatial filtering, which is characterized by the center frequency, \( (f_x, f_y) \). In general, a band pass filter is required to extract the local phase information from an image. A Gabor filter is the one of the most commonly used types of band pass filter. This filter can achieve the theoretical minimum product of spatial width and bandwidth for any complex valued filter [19]. A Gabor filter also presents a more accurate computation of variations and better localization for the space and frequency domains [20]. We thus used a Gabor filter with a designated wavelength and bandwidth of 12 pixels and 0.6 octaves, respectively.

2.2 Phase-Based Optical Flow Technique

In this study, we developed our algorithm based on the existing phase-based optical flow technique introduced by Fleet and Jepson [11], and analyzed whether the proposed method increases the tracking performance. The phase-based method defines the component velocity in terms of the instantaneous motion normal to level phase contours in the output of the spatial filter. The output of the Gabor filter is complex valued and can be represented by Eq. (7)

\[
R(x, t) = \rho(x, t)e^{i\phi(x, t)}
\]

where \( \rho(x, t) \) and \( \phi(x, t) \) are the amplitude and phase components of input signals, and can be represented by Eqs. (8) and (9), respectively.

\[
\rho(x, t) = |R(x, t)| = \sqrt{\text{Re}[R(x, t)]^2 + \text{Im}[R(x, t)]^2}
\]

\[
\phi(x, t) = \arg[R(x, t)] = \text{Im}[\log_e R(x, t)] \in (-\pi, \pi)
\]

If the pixel \( x \) is located at a point of constant phase over the image frame, the point \( x \) on such a contour satisfies \( \phi(x, t) = C \). Thus, phase variation on the image can be defined as follows:
Table 1 Comparative result for the synthetic image sequence.

|                | Sinusoidal Image | Square Image | Yosemite Image |
|----------------|------------------|--------------|----------------|
|                | AAE (°)  | D (%)     | AAE (°)  | D (%)     | AAE (°)  | D (%)     |
| Horn and Schunck| 4.19±0.59 | 100       | 47.21±14.60 | 100           | 32.43±30.28 | 100       |
| Lucas and Kanade (λ≥1.0)| 2.47±0.16 | 100       | 0.21±0.16 | 7.9          | 5.20±9.45 | 35.1       |
| Uras et al. (Unthresholded) | 2.59±0.71 | 100       | 0.15±0.10 | 26.1         | 10.44±15.00 | 100       |
| Singh (Step 1, n=2, w=2) | 91.71±0.04 | 100       | 49.03±21.38 | 100           | 18.24±17.02 | 100       |
| Anandan        | 30.80±5.45 | 100       | 31.46±18.31 | 100           | 15.84±13.46 | 100       |
| Nagel          | 2.55±0.93  | 100       | 34.57±14.38 | 100           | 11.71±10.59 | 100       |
| Fleet and Jepson (r=1.25) | 0.03±0.01 | 100       | 0.18±0.13  | 12.6         | 4.95±12.39 | 30.6       |
| Developed Method | 5.75±5.56 | 80.23     | 3.00±1.49  | 95.50        | 4.09±3.80  | 62.44      |

* AAE: Average Angular Error, D: Density
* Data: Mean ± SD (Standard Deviation)

\[ \phi_t = -(v \cdot \phi) = -||\phi_x||\|proj_{\phi_x}(v) \] (10)

where \( \phi \) indicates the normalized vector of \( \phi \), and \( v \) represents the full velocity. The phase component \( \phi \) can also be replaced by the frequency vector \((2\pi f_x, 2\pi f_y)\), and the component velocity can be denoted as \( V_c \), and computed as

\[ V_c = proj_{\phi_x}(v)\phi_x^p = -\frac{\phi_x}{2\pi(f_x^2 + f_y^2)}(f_x, f_y) \] (11)

This phase-based technique shows a more robust performance than other tracking methods due to the utilization of phase signals from the band-pass filter. Phase information, however, can become unstable because of the phase singularities. We thus applied reliability verification to reject irregular and nonlinear component velocities by applying constraints to the frequency and amplitude of each pixel. The reliability of each pixel is evaluated by the Mean Squared Error (MSE) method as shown in Eq. (12) [14]:

\[ \Delta \phi(x, t) = (a + \Psi(x)(x) - \phi(x, t) \] (12)

We determined that the component velocity shows higher reliability, if it is larger than the threshold value derived from the MSE method. Otherwise, the component velocity was excluded from the motion tracking process. Finally, the full velocity was calculated by integrating the component velocities of each pixel.

2.3 Error Measurement

In order to evaluate the performance of the proposed technique, we selected the average angular error (AAE) and the density as evaluation indices [11], and quantitatively analyzed the results.

Velocity can be written as displacement per frame unit with \( v = (u, v) \) pixel/frame, or \( v = (u, v, 1) \) in the units of (pixel, pixel, frame). If the ground truth data are known, it is possible to calculate the angular error between estimated and correct velocity vectors. Thus, if the velocity \( v \) is represented by \( v = (v_1, v_2)^T \), the 3-D velocity vector can be written as \( \vec{v} = \frac{1}{\sqrt{u^2 + v^2 + 1}}(u, v, 1)^T \). Therefore, the angular error \( (\psi_E) \) between the correct velocity \( V_c \) and the estimated velocity \( V_e \) can be represented as

\[ \psi_E = \arccos(\vec{V}_c \cdot \vec{V}_e) \] (13)

Since the AAE presents the angular error between the estimated and the actual velocity vectors, a higher tracking accuracy is obtained as its value approaches zero.

Density is defined as the ratio of the number of pixels from which the component velocity is derived to the number of pixels in the entire image. Density is proportional to the AAE in the phase-based optical flow technique. The AAE increases as the density increases in the image, since the unreliable and nonlinear component velocities also increase simultaneously with each other. Thus, the density is utilized as an indirect evaluation index to assess the tracking performance of optical flow [14], [21].

2.4 Data Analysis

To evaluate the performance of the proposed technique, we examined the tracking performance for synthetic and realistic images for which the ground truth data were known. The synthetic images consisted of Sinusoidal, Square and Yosemite images, and the realistic images consisted of NASA, Rubik and Taxi images [9]. We analyzed the experimental results produced by developed and existing methods to evaluate the tracking performance. We also tested existing image filtering methods to evaluate whether ABF increases the tracking capacity. Linear and nonlinear filters were selected as existing image filtering methods (Table 1). The average filter and the Gaussian filter were employed for the linear filter group, while the median filter and the wiener filter were selected for the nonlinear filter group. Lastly, in order to analyze the variation of performance in accordance with kernel size, kernel sizes of 3x3 and 5x5 were adopted for the linear and nonlinear filter groups.
3. Results

3.1 Comparison of Experimental Results between Proposed and Existing Methods

3.1.1 Synthetic Images

a) Sinusoidal Image
In the Sinusoidal image (Fig. 1 (a)), two superimposed sinusoids move with speeds of 1.63 pixels/frame at 54 degrees and 1.02 pixels/frame at −27 degrees, and lattice patterns are translated with a perceived velocity of (1.585, 0.863) pixels/frame (Fig. 1 (b)). The Singh method yielded the largest AAE of 91.71 ± 0.04°, while the Fleet & Jepson (F&J) method yielded the smallest error of 0.03 ± 0.01° (Table 1). The existing methods also revealed a density of 100% on the Sinusoidal image. Namely, the F&J method derived the most accurate flow result (Fig. 1 (c)), while the Singh method produced the poorest accuracy under the same experimental conditions. On the other hand, the proposed method showed an AAE of 5.75 ± 5.56° and a density of 82.23% for the Sinusoidal image (Fig. 1 (d)).

b) Square Image
The Square image consists of a white square 40 pixels wide on a black background, and the square translates with a velocity of $v = (1, 1)$ pixels/frame (Fig. 2 (a), (b)). The Singh method showed the largest AAE of 49.03 ± 21.38°, while the Uras method showed the smallest error of 0.15 ± 0.10° for the Square image (Table 1). The Horn and Schunck (H&S), Singh, Anandan and Nagel methods produced the largest density of 100%, while the lowest value of 7.9% was revealed by the Lucas and Kanade (L&K) method. The existing methods were unable to produce high tracking performance, since both the AAE and density were shown to be large or small at the same time. In particular, the F&J method yielded poor performance due to the small density of 12.6%, although the AAE was calculated to be 0.18 ± 0.13° (Fig. 2 (c)). On the other hand, the proposed method showed reasonable tracking results for the Square image as the AAE and density were derived to be 3.00 ± 1.49° and 95.50%, respectively (Fig. 2 (d)).

c) Yosemite Image
The Yosemite image is a more complex test case compared to the Sinusoidal and Square images. The motion of the clouds is 2 pixels/frame to the right, while the rest of the flow is divergent with a speed of approximately 5 pixels/frame in the lower left corner (Fig. 3 (a), (b)). The H&S

Fig. 1 Vector flow of the sinusoidal image.

Fig. 2 Vector flow of the square image.

Fig. 3 Vector flow of the yosemite image.
method produced the largest AAE of 32.42 ± 30.28°, while the smallest error of 4.95 ± 3.80° was revealed by the F&J method (Table 1). The existing methods showed poor tracking results with large AAEs over 10°, although the density was shown to be 100%. In particular, most of the existing methods failed to extract the clouds in the upper area as shown in Fig. 3 (c). Our method, however, tracked the upper clouds more precisely than all other employed methods (Fig. 3 (d)). Due to this improvement, our method increased the density by a factor of two and decreased AAE by about 1° compared to the F&J method. Consequently, the proposed method showed the most accurate performance for the Yosemite image.

3.1.2 Realistic Images

a) NASA Image
The NASA image shows primarily dilational motions. The camera moves along its line of sight toward the Coke can, and the velocities are typically less than one pixel/frame (Fig. 4 (a)). The F&J method underestimated the motion flow, since the main object and background areas were not separated distinctly (Fig. 4 (b)). This method also yielded large motion errors due to uncertain extraction of the boundary information. In the H&S method, outliers that exist in the input image remained in the output signal (Fig. 4 (c)). On the other hand, the proposed method differentiated between the main object and the background areas. Our method also preserved the boundary information effectively, and thereby prevented underestimation of the motion flow (Fig. 4 (d)).

(b) Rubik Image
In the Rubik sequence, a cube is rotating counter-clockwise on a turntable. The motion field of the cube shows that the velocities are typically less than two pixels/frame. Velocities on the turntable range from 1.2 to 1.4 pixels/frame, and those of the cube are between 0.2 and 0.5 pixels/frame (Fig. 5 (a)). Although the F&J method partly extracted the rotating motion of the turntable area, the motion fields of the cube and the white turntable were underestimated due to the unclear segmentation (Fig. 5 (b)). The H&S method struggled to extract the correct motion fields due to the low density, nevertheless the motion flows of the cube and the turntable were partially tracked (Fig. 5 (c)). The proposed method however was able to track the rotating motions of the cube and the turntable, and calculated the rotation velocities for the entire image. Our method also succeeded in extracting the motion flows of the three sides of the cube as well as those of the white turntable (Fig. 5 (d)).

(c) Taxi Image
In the Taxi sequence, there are four moving objects: a) a taxi turning the corner, b) a car in the upper left, c) a car driving from left to right, and d) a van in the lower right driving from right to left. The image speeds of the four moving objects are approximately 1.0, 3.0, 3.0 and 0.3 pixels/frame, respectively (Fig. 6 (a)). The F&J method produced numerous tracking errors due to the uncertain separation of the motion fields (Fig. 6 (b)). The H&S method extracted only a) the taxi turning the corner out of the total four moving areas. This method also produced numerous errors in all of the image regions (Fig. 6 (c)). The proposed method however removed the outliers and separated the main objects from the background area. Consequently, our method tracked the four moving objects precisely, and improved the tracking accuracy (Fig. 6 (d)).
Table 2  Result of the different filtering method (synthetic image).

|                  | Sinusoidal Image | Square Image | Yosemite Image |
|------------------|------------------|--------------|---------------|
|                  | AAE (°)          | D (%)        | AAE (°)       | D (%)          | AAE (°)       | D (%)          |
| **Linear Filter**|                  |              |               |                |               |                |
| Average (3x3)    | 9.15±5.53        | 31.56        | 3.58±1.70     | 49.75          | 4.38±3.99     | 32.68          |
| Sigma (3x3)      | 6.50±6.22        | 49.33        | 3.27±1.80     | 95.00          | 4.14±3.87     | 32.23          |
| Sigma (5x5)      | 6.51±6.22        | 49.22        | 3.28±1.80     | 95.10          | 4.15±3.87     | 32.62          |
| Median (3x3)     | 32.97±15.30      | 28.89        | 5.00±3.70     | 95.25          | 4.54±4.86     | 14.05          |
| **Nonlinear Filter** |                |              |               |                |               |                |
| Average (5x5)    | 63.47±10.70      | 21.56        | 5.78±0.85     | 94.75          | 5.42±3.81     | 9.04           |
| Wiener (3x3)     | 9.10±7.35        | 32.11        | 3.01±1.48     | 96.50          | 4.28±3.97     | 30.37          |
| Wiener (5x5)     | 10.34±5.89       | 9.78         | 3.19±1.76     | 96.00          | 4.45±4.08     | 30.32          |
| Proposed Method  | 5.75±5.56        | 80.23        | 3.00±1.49     | 95.50          | 4.09±3.80     | 62.44          |

* AAE: Average Angular Error, D: Density
* Data: Mean ± SD (Standard Deviation)

In particular, the median filter revealed the largest variance of tracking performance as the kernel size increased from 3x3 to 5x5. On the other hand, the kernel size did not cause large variance for the Gaussian filter as the changes of AAE and density revealed very small.

b) Square Image
Our method produced the smallest AAE of 3.00 ± 1.49°, whereas the median filter (5x5) produced the largest error of 5.78 ± 4.85° (Table 2). The average filter (5x5) and the wiener filter (3x3) produced the largest density of 96.50%, while the smallest densities were derived from the average filter (3x3) and the median filter (5x5). The median filter (5x5) revealed distorted motion flow on each corner of the white square object. Hence, it caused significant tracking errors, and thereby showed the poorest result among the employed methods. Additionally, we confirmed that the 3x3 kernel showed a much smaller AAE than the 5x5 kernel for all linear and nonlinear filters. The density, however, was shown to be larger with the 5x5 kernel than the 3x3 kernel for the linear filter, while the 3x3 kernel showed a larger density for the nonlinear filters. We also observed that the average filter produced the largest variance of AAE and density on the different kernel sizes. On the other hand, the kernel size did not lead to a large variance of parameters for the Gaussian filter, similar to the result obtained for the Sinusoidal image.

c) Yosemite Image
The proposed method yielded the most accurate result with the smallest AAE of 4.09 ± 3.80° and the largest density of 62.44% (Table 2). The existing methods showed reasonable AAE, but the density was almost 50% less than that of the proposed method (Fig. 7 (a), (b)). The median filter (5x5) showed the poorest result for the Yosemite image. This filter failed to track the motion flows of the main objects due to excessive blurring (Fig. 7 (c)). Particularly, the clouds in the upper area were rarely extracted, which led to a decrease of the density. We also confirmed that the 3x3 kernel produced more robust performance than the 5x5 kernel for all employed filters. In particular, the median filter revealed the largest variance of the evaluation parameters between the
different kernels. However, the kernel size did not cause any changes for the Gaussian filter, as was the case for the pre-mentioned synthetic images. Consequently, our method showed the best performance among the employed filtering methods (Fig. 7 (d)).

3.2.2 Realistic Images

a) NASA Image
The median filter (5x5) yielded the smallest density of 38.61%, while the average filter (5x5) showed the largest value of 54.88% (Table 3). The median filter (5x5) was unable to extract the motion flows of the corner areas, which degraded the visibility. This filter also produced the largest density variation according to the size of the kernel, while the Gaussian filters produced the same density of 52.51% for each size. In addition, regular changes did not occur on the NASA image, unlike the synthetic images. The 5×5 kernel produced a larger density for the average and wiener filters, whereas the median filter produced the larger value for the 3x3 kernel. Consequently, the kernel size did not show any correlation with the density variation for the NASA image.

b) Rubik Image
The median filter (5x5) produced the smallest density of 48.00%, and the wiener filter (5x5) yielded the largest density of 60.30% (Table 3). The median filter (5x5) was unsuitable for extracting the motion fields of the edges of the cube, since the boundary information was not preserved. The largest density variance according to the kernel size was also revealed on the median filter. The Gaussian filter however showed the same density of 58.37% for each size. Additionally, no correlation was found between the kernel size and the density. The average and the median filters revealed much larger densities with the 3x3 kernel, whereas the 5x5 kernel showed larger densities for the wiener filter. Namely, the kernel size did not influence the changes of density for the Rubik image, in agreement with the results for NASA image.

c) Taxi Image
The average filter (5x5) showed the largest density of 19.85%, while the median filter (3x3) presented the smallest density of 17.27% (Table 3). The average and the Gaussian filters did not present large tracking errors, and reasonably extracted the four moving objects (Fig. 8 (a), (b)). The median filter, however, showed minor outliers in the central area, and failed to track the correct motion flow of b) the car in the upper left area (Fig. 8 (c)). This filter also yielded the largest density variance of 1.06% with respect to the different kernel sizes. However, no variation occurred for the Gaussian filter, which produced the same density of 18.71% for each size. In addition, the 5x5 kernel yielded much larger densities than that of the 3x3 kernel for all linear and nonlinear filters, except for the Gaussian filter. Hence, it was confirmed that the kernel size did not cause any changes of density for any of the realistic images.

| Filtering Method | NASA | RUBIK | TAXI |
|------------------|------|-------|------|
| Average (3x3)    | 52.96% | 58.81% | 18.86% |
| Average (5x5)    | 54.88% | 58.52% | 19.85% |
| Gaussian (3x3)   | 52.51% | 58.37% | 18.71% |
| Gaussian (5x5)   | 52.51% | 58.37% | 18.71% |
| Median (3x3)     | 43.93% | 56.59% | 17.27% |
| Median (5x5)     | 38.61% | 48.00% | 18.33% |
| Wiener (3x3)     | 52.07% | 59.11% | 18.71% |
| Wiener (5x5)     | 53.40% | 60.30% | 18.79% |
| Proposed Method  | 52.51% | 58.81% | 18.79% |
4. Discussion

The earlier phase-based method produced a reasonable result by using spatiotemporal filtering. The earlier method, however, revealed performance instability in local areas. We adopted several highly loaded techniques instead of spatiotemporal filtering to solve the pre-mentioned problems. Despite the high computational load, we managed to reduce the computation times for all test images compared to the existing methods. The F&J method yielded computation times of 3.44 s and 2.94 s for the Yosemite and RUBIK images, while the proposed method only needed 2.45 s and 2.13 s, respectively. This result may come from the better localization of the spatial and frequency domains caused by the employed filters. It also led to rapid computation of the component velocities. Computations were performed on a standard personal computer equipped with a 1.87 GHz Intel core 2 CPU, 2 GB of main memory and a RADEON X800GT graphic card.

The proposed method showed reasonable tracking results for all synthetic images except for the sinusoidal image. Firstly, our method revealed relatively low tracking performance compared to existing method on the Sinusoidal image. It is considered that the inaccurate separation of outer areas from the input image caused the low tracking performance. As mentioned in Sect. 3, in this image the lattice patterns move in the diagonal direction with constant velocity. The proposed method, however, extracted an over-estimated motion from the left and right border areas (Fig. 1 (d)). It thus degraded the tracking accuracy for the Sinusoidal image.

The existing methods such as L&K, Uras and F&J yielded much smaller AAEs than that of the proposed method. The density however was less than 30% of that of the proposed method. In particular, the L&K method showed poor tracking performance for the Square image due to the small density of 7.9%, even though a small AAE of $0.21 \pm 0.16^\circ$ was shown. AAE is generally regarded as the most accurate evaluation index to quantitatively analyze the tracking performance. A small AAE however does not necessarily indicate a robust tracking accuracy. In other words, it is possible to determine that the tracking method shows a low performance when the density is small, even when a significantly small AAE is shown (Fig. 2 (c), Table 1). The proposed method produced high tracking performance for the Square image by extracting a small displacement from the large objects efficiently. Our method however failed to correctly separate the central areas that occupy 4.5% of the entire region (Fig. 2 (d)). This result may occur because our technique tracked only the outer areas of the square object, whereas the inner areas were not extracted from the original image.

The proposed method also revealed high tracking accuracy for the Yosemite image, as the AAE and density were derived to be $4.09 \pm 3.80^\circ$ and 62.44%, respectively. Most of the existing methods produced a density of 100%, whereas the AAE was derived to be larger than $10.00^\circ$ due to the nonlinear motion in the original image. The F&J method removed the nonlinear pixels by applying constraint conditions to each pixel. The F&J method however yielded a small density of $30.60\%$ and a large standard deviation of $12.39^\circ$ as nonlinear motions were removed from the tracking process. On the other hand, our method resulted in a density twice as large as that of the F&J method, and decreased the AAE by approximately $1^\circ$ simultaneously. The main differences between the F&J method and the proposed method are the ABF and the Gabor based spatial filter. Namely, the F&J method which is based on the spatiotemporal filter, computed the component velocity from a small spatial neighborhood. This method also revealed an unsolved aperture problem, since additional conditions were not employed except for the constraint of the original filter. On the other hand, our method derived the component velocity for each single spatial location by using a spatial filter. In addition, the proposed method minimized the influence of the magnitude variations caused by illumination changes, and thereby produced improved results.

The proposed method also showed high tracking performance for realistic images such as the NASA, Rubik and Taxi images. The F&J method yielded a small density for realistic images due to the limitation of the constraint condition. In particular, this method only performed a simple classification due to the excessive application of the constraint, and failed to extract the three sides of the cube and the upper areas of the turntable. Although the H&S method revealed a larger density than that of the F&J method, moving regions were not extracted correctly due to the uncontrolled outliers in the original image as shown in Fig. 5 (c). On the other hand, the proposed method produced high tracking performance for all realistic images. In particular, a total of four moving areas were distinctly separated from the background in the Taxi image. Our method also extracted the three sides of the rotating cube motion correctly in the Rubik image. The reason for this improvement is mainly because the bilateral filter replaces the brightness value of each pixel with the weighted average from neighboring pixels. The existing methods were unable to track the white turntable area in the Rubik image due to the remaining aperture problem (Fig. 5 (b), (c)). We used ABF and the intersection of constraints (IOC) approach to overcome the aperture problem. The IOC method uses only the minimum local motions of two edge parts to derive the global motion flow by finding the intersection of all possible global motions [22]. It is possible to improve the performance of the IOC method by better preservation of the edge information. We therefore used the ABF to preserve the maximum boundary information, and thereby efficiently solved the aperture problem. Consequently, we succeeded in extracting the motion flows of the white turntable area (Fig. 5 (d)), which led to the improvement of the tracking performance.

The median filter showed the lowest tracking performance on all synthetic and realistic images. Particularly, this filter produced an AAE 11.04 times larger than that of
the proposed method for the Sinusoidal image. The density, at the same time, was only 78.44% of that of the proposed method. The median filter also produced a density of 72.13% of that of the proposed method for the Yosemite image. Despite the effectiveness of the median filter in removing the outlier brightness pixels, this filter modifies the contour information of the original images. Namely, the images that contain small deviations and various contours such as the Sinusoidal and Yosemite images tend to be influenced by the filtering process. Thus, the median filter showed the lowest tracking performance for all test images.

Finally, the linear and nonlinear filters showed robust tracking performance for all synthetic images, when the 3x3 kernel was applied to the filtering method. The median filter produced the largest variance of AAE and density according to the kernel size, while the Gaussian filter produced the smallest variances for all synthetic and realistic images. As the kernel size increases, the original pixels tend to be changed since the number of neighboring pixels that influence the brightness value of each pixel also increases. The contours can also be unclear as the kernel size increases. Thus, the 3x3 kernel size delivered a more accurate performance than the 5x5 size.

5. Conclusions

In this paper, we proposed an advanced phase-based optical flow method to improve the tracking accuracy for motion flow. The proposed method was mainly based on ABF and Gabor based spatial filtering. In order to evaluate the tracking accuracy, we adopted AAE and density as the evaluation indices, and examined the performance for synthetic and realistic images. Consequently, it was confirmed that the proposed method offers more robust performance than the existing methods.

It is considered that the proposed technique could have various applications, such as security systems and vehicle recognition. It is also possible to use our algorithm in medical imaging including CT (Computed Tomography), ultrasound and MRI (Magnetic Resonance Imaging). In order to pursue these applications of our technique, our future work will focus on developing the algorithm implementation to optimize the tracking performance based on pre- and post-processing methods. Through this process, it is expected that the proposed method will provide a diagnostic function by tracking the contrast media in medical images.

Acknowledgments

This work was supported by Industrial Source Technology Development Program (10033726) funded by the Ministry of Knowledge Economy (MKE), Korea.

References

[1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” Int. J. Comput. Vis., vol.47, no.1-3, pp.7–42, 2002.
[2] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.4, pp.594–611, 2006.
[3] A. Bruhn, T. Brox, S. Didas, and J. Weickert, “Highly accurate optic flow computation with theoretically justified warping,” Int. J. Comput. Vis., vol.67, no.2, pp.141–158, 2006.
[4] J.W. Lee, S. You, and U. Neumann, “Large motion estimation for omnidirectional vision,” Proc. IEEE Workshop on Omnidirectional Vision, pp.161–168, 2000.
[5] J.W. Suh and Y.S. Ho, “Error concealment based on motion vector recovery using optical flow fields,” IEICE Trans. Commun., vol.E86-B, no.4, pp.1383–1390, April 2003.
[6] A. Bruhn and J. Weickert, “A multigrid platform for real-time motion computation with discontinuity-preserving variational methods,” Int. J. Comput. Vis., vol.70, no.3, pp.257–277, 2006.
[7] B.K.P. Horn and B.G. Schunck, “Determining optical flow,” Artif. Intell., vol.17, pp.185–203, 1983.
[8] B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” Proc. 1981 DARPA Imaging Underst. Workshop, pp.121–130, 1981.
[9] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkkitt, “Performance of optical flow techniques,” Proc. IEEE Comput. Vis. and Pattern Recognit., pp.236–242, 1992.
[10] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, “Performance of optical flow technique,” Int. J. Comput. Vis., vol.12, no.1, pp.43–77, 1994.
[11] D.J. Fleet and A.D. Jepson, “Computation of component image velocity from local phase information,” Int. J. Comput. Vis., vol.5, no.1, pp.77–104, 1990.
[12] M. Tomasi, M. Vanegas, F. Barranco, J. Diaz, and E. Ros, “High performance optical flow architecture based on a multi scale, multi orientation phase-based model,” IEEE Trans. Circuits Syst. Video Technol., vol.20, no.12, pp.1797–1807, 2010.
[13] D.J. Fleet and A.D. Jepson, “Stability of phase information,” IEEE Trans. Pattern Anal. Mach. Intell., vol.15, no.12, pp.1253–1268, 1993.
[14] T. Gautama and M.M. van Hulle, “A phase-based approach to the estimation of the optical flow field using spatial filtering,” IEEE Trans. Neural Netw., vol.13, no.5, pp.1127–1136, 2002.
[15] A. Wong, “Adaptive bilateral filtering of image signals using local phase characteristics,” Signal Process., vol.88, no.6, pp.1615–1619, 2008.
[16] M. Elad, “On the origin of the bilateral filter and ways to improve it,” IEEE Trans. Image Process., vol.11, no.10, pp.1141–1151, 2002.
[17] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” Proc. IEEE Int. Conf. on Comput. Vis., pp.839–846, 1998.
[18] B. Zhang and J.P. Allebach, “Adaptive bilateral filter for sharpness enhancement and noise removal,” IEEE Trans. Image Process., vol.17, no.5, pp.664–678, 2008.
[19] H. Arora, A.M. Namboodiri, and C.V. Jawahar, “Accurate image registration from local phase information,” Proc. 13th Natl. Conf. on Commun., pp.37–41, 2007.
[20] M.E. Spetsakis, “An optical flow estimation algorithm that uses Gabor filters and affine model for flow,” Technical Report CS-94–06, York University, 1994.
[21] S.H. Lai, “Computation of optical flow under non-uniform brightness variations,” Pattern Recognit. Lett., vol.25, no.8, pp.885–892, 2004.
[22] E.H. Adelson and J.A. Movshon, “Phenomenal coherence of moving visual patterns,” Nature, vol.300, no.5892, pp.523–525, 1982.
Ju Hwan Lee received the B.S. degree in Biomedical Engineering from Konkuk University, Korea in 2009. Since 2009, he has been a Ph. D candidate in the Dongguk university, Seoul, Korea. His research interest ultrasound image analysis, motion analysis and pattern recognition.

Sung Yun Park received the B.S. and M.S. degrees in Biomedical Engineering from Konkuk University, Korea in 2005 and 2007, respectively. Since 2010, he has been a Ph. D candidate in the Dongguk University, Seoul, Korea. His research interest neural network, pattern recognition and fluid dynamics.

Sung Jae Kim received the B.S. degree in Mechanical Engineering from Seoul National University of Science & Technology in 1996, and the M.S. degree in Biomedical Engineering from Konkuk University in 2005. Since 2011, he has been a Ph. D candidate in the Dongguk University. His research interest medical image analysis, pattern recognition and simulation.

Sung Min Kim received the B.S. degree in Electrical Engineering from Yonsei University in 1985, and the M.S degree in Biomedical Engineering from University of Iowa in 1991. He also received the Ph. D degree in Biomedical Engineering from same university in 1995. During 2002–2009, he was an associate professor in Konkuk university, Korea. Since 2009, he has been an associate professor in Dongguk university, Seoul, Korea.