

# A robust optical flow estimation technique using gradient orientations

Pramuk Boonsieng\*, Toshiaki Kondo, Waree Kongprawechnon

School of Information,  
Computer and Communication Technology Sirindhorn International Institute of Technology,  
Thammasat University

\*Corresponding author, e-mail: b.pramuk@gmail.com

Received 7 Dec 2012

Accepted 29 Jul 2013

**ABSTRACT:** The gradient method is a classical technique for calculating optical flows in image sequences. In spite of its serious limitations, including the extreme sensitivity to changing illuminations, it is still popular and widely used in various image processing and computer vision applications because of its efficiency in computation. We previously proposed an improved version of the gradient method based on gradient orientations instead of intensities. Since gradient orientations are known to be stable to global changes of intensities, the proposed technique can perform optical flow estimation regardless of irregular lighting. We refer to this method the gradient orientation based gradient method (GOGM). This paper is a further study of the GOGM. New simulation and experimental results show that this technique works at a video rate of 16 Hz and outperforms the traditional gradient method where the aperture problem and illumination changes are involved.

**KEYWORDS:** motion estimation, gradient method, the Lukas-Kanade algorithm

## INTRODUCTION

Motion estimation is an important task in the fields of image sequence processing and computer vision. It finds a variety of applications, including object-based video coding (e.g., MPEG-4), object detection and tracking, scene change detection for video indexing, image stabilization for digital cameras and camcorders, and dynamic 3-D scene analysis for autonomous navigation. Motion estimation techniques in the spatial domain may be classified into two groups: gradient-based methods and correlation-based methods. This paper is concerned with the former group, the gradient-based approaches, which is more suitable for computing a dense optical flow field. The spatiotemporal gradient method is one of the gradient-based techniques and it is often more simply referred to as gradient method (GM)<sup>1–4</sup>. This technique assumes that the image intensity is constant along the motion trajectory over time. Because of the assumption of brightness constancy, it is extremely sensitive to varying image intensities that are often caused by irregular illuminations. The approach, however, is still widely used for research purposes, partly because one can easily obtain the sophisticated code that can execute motion estimation at a video rate such as the OpenCV library for free<sup>5</sup>. Meanwhile, various approaches have been also proposed to cope with

the varying lighting conditions. One approach is to model the intensity changes, for example, linearly that can handle both multiplicative and additive change of image intensities. The model is incorporated in the process of the computation for motion estimation<sup>6</sup>. The effectiveness of this approach, however, will be limited to the case where the model is a good approximation of real intensity changes. The employment of a more robust feature, instead of image intensities, is another promising approach. The use of the gradients of image intensities is described in Ref. 7. It is certain that image gradients are a more robust feature than image intensities. This technique assumes that the image gradient is constant along the motion trajectory over time<sup>8</sup>. We call this approach the gradient vector-based method or GVGM. However, it is yet susceptible to multiplicative changes of intensities<sup>9</sup>. To cope with the irregular lighting condition problem, we previously proposed a method that employs unit gradient vectors (UGVs) in place of image intensities and gradients. This is because UGVs are known to be robust to varying lighting conditions<sup>10,11</sup>. By theory, the UGVs are invariant to both additive and multiplicative changes of intensities. We call this technique the gradient orientation based gradient method or GOGM<sup>12</sup>. This paper focuses on the implementation, analyses, and evaluations of the GOGM on both synthetic and real image sequences, and comparing with

the GM and GVGM. The rest of the paper is organized as following. In the Methods section, we first review the conventional spatiotemporal gradient method (or GM) and then describe the proposed method GOGM together with a related method GVGM. In Results and Discussion, we demonstrate the performances of the GM, GVGM, and GOGM under a constant lighting condition, those under a varying lighting condition, and those on various real image sequences. Finally, we state our conclusions and future work in the last section.

## METHODS

### Gradient method

Image flow is the velocity field in the image plane due to the motion of the observer, the motion of objects in the scene, or apparent motion which is a change in the image intensity between frames that mimics object or observer motion. Gradient-based methods are one of the schemes devised for calculating image flow. They exploit the relationship between the spatial and temporal gradients of intensity<sup>13</sup>. When image intensities are assumed to be constant along motion trajectory over time, we have

$$g(x, y, t) = g(x + dx, y + dy, t + dt), \quad (1)$$

where  $g(x, y, t)$  indicates the image intensities at pixel coordinates  $(x, y)$  at time  $t$ , and  $(dx, dy)$  denotes the image displacements over time  $dt$ . By taking the first-order Taylor series expansion of the right-hand side of (1), we obtain

$$g(x + dx, y + dy, t + dt) \approx g(x, y, t) + g_x dx + g_y dy + g_t dt, \quad (2)$$

where the subscripts denote partial derivatives. From (1) and (2), one can derive the well-known optical flow constraint equation or OFCE

$$g_x u + g_y v + g_t = 0, \quad (3)$$

where  $(u, v) = (dx/dt, dy/dt)$  denotes the optic flow (image flow), and  $(g_x, g_y, g_t) = (\partial g(x, y, t)/\partial x, \partial g(x, y, t)/\partial y, \partial g(x, y, t)/\partial t)$  express the spatiotemporal image gradients. Eq. (3) is underdetermined because it contains two unknowns,  $u$  and  $v$ . To solve the equation for the optical flow  $(u, v)$ , we may introduce either a local or global smoothness constraint. The simplest local smoothness constraint is the assumption that the motion is constant within the small area where a motion vector is estimated, though it is possible to extend the assumption to

polynomial or other parametric models<sup>2-5</sup>. The global smoothness constraint assumes that the optical flow changes smoothly over the entire image<sup>1,5,13</sup>. A comparative study between these two optimization techniques can be found in Ref. 2. The local approach is more suitable for the applications that require well-defined boundaries between areas with different motions because each motion vectors is computed within a block independently of its neighbouring blocks. On the contrary, the boundaries by the global approach are blurred because of its smoothing operation among adjacent motion vectors. In this paper, we employ the local smoothness constraint. The most popular local approach for optical flow computation is the linear differential method of Lucas and Kanade<sup>3</sup>. Under the local smoothness constraint, the optical flow  $(u, v)$  can be determined by minimizing the quadratic cost-function  $F$

$$F = \sum (g_x u + g_y v + g_t)^2, \quad (4)$$

where  $\sum$  indicates the sum over a small region or a block for computing a flow vector. The least squares solution of (4) is given by

$$\begin{cases} u = \frac{(\sum g_x g_y)(\sum g_y g_t) - (\sum g_y^2)(\sum g_x g_t)}{(\sum g_x^2)(\sum g_y^2) - (\sum g_x g_y)^2} \\ v = \frac{(\sum g_x g_y)(\sum g_x g_t) - (\sum g_x^2)(\sum g_y g_t)}{(\sum g_x^2)(\sum g_y^2) - (\sum g_x g_y)^2} \end{cases} \quad (5)$$

### Gradient orientation based gradient method

In practice, however, the assumption for the OFCE is often violated by varying lighting conditions. As a result, all the motion estimation techniques based on the OFCE are susceptible to the changes of lighting conditions. To overcome the problem, we propose to use gradient orientation information (GOI) in place of traditional image features such as intensities and gradients, because gradient orientation is known to be insensitive to varying illuminations<sup>10,11</sup>. We devise a novel technique, called the gradient orientation based gradient method or GOGM, by introducing GOI to the GM. Our previous work shows that GOI can be effectively utilized in the form of UGVs<sup>14</sup>. We first summarize the procedure of the GOGM. Let  $g(x, y)$  be the image intensities at pixel coordinates  $(x, y)$ . The gradient vectors of  $g(x, y)$  are approximated by partial derivatives

$$\begin{cases} g_x(x, y) = g(x, y) * k_x \\ g_y(x, y) = g(x, y) * k_y \end{cases}$$

where the symbol  $*$  denotes convolution, and  $k_x$  and  $k_y$  are the first-derivative operators in the  $x$  and  $y$

directions, respectively. The Sobel operators may be used as the first-derivative operators. Note that  $g_x(x, y)$  and  $g_y(x, y)$  are image gradients which are employed in GVGM. Next, GOI can be obtained as UGVs by dividing gradient vectors by their magnitudes

$$\begin{cases} n_x(x, y) = \frac{g_x(x, y)}{\sqrt{g_x(x, y)^2 + g_y(x, y)^2}} \\ n_y(x, y) = \frac{g_y(x, y)}{\sqrt{g_x(x, y)^2 + g_y(x, y)^2}} \end{cases}$$

where zeros are assigned to  $(n_x, n_y)$  if the denominator is equal to or close to zero. In this way, we extract GOI only by discarding the magnitude information of the gradient vectors.

Fig. 1a shows an 8-bit grey-scale image, Baboon, of size 256 by 256 pixels. Figs. 1b and 1c show the gradient orientation patterns  $n_x$  and  $n_y$  of the Baboon image. The patterns are scaled between 0 and 255 for visualization purposes. It should be emphasized that horizontal lines, i.e., vertical gradients are rich in  $n_x$  (Fig. 1b), while vertical lines, i.e., horizontal gradients are more dominant in  $n_y$  (Fig. 1c).

The use of  $n_x$  and  $n_y$  is more advantageous than directly using angular values  $\theta$  (radius) as GOI because there is no need to take an extra precaution when computing the derivatives of GOI (difference between two angles cannot exceed  $\pi$ )<sup>10</sup>. Fig. 1d shows part of the Baboon image of size 40 by 40 pixels. Fig. 1e plots the gradient vectors  $(g_x, g_y)$  within the subimage. Prominent vectors can be seen where intensities change rapidly, whereas no distinguished vectors are seen where intensities change gradually. Meanwhile, Fig. 1f shows the UGVs in the same subimage. By the comparison between Fig. 1e and (f), it is evident that a gradient orientation pattern carries rich spatial information even in areas where intensities change gradually.

Furthermore, Fig. 1g shows the same subimage as in Fig. 1d except that the intensity of the upper half of it is reduced by 50%. All image intensity based motion estimation techniques are inevitably ill affected by varying intensities like this. Fig. 1h shows the gradient vectors of the subimage. The gradient vectors in the upper half of the subimage are considerably different from those of Fig. 1e. This indicates a weakness of gradient vector based motion estimation techniques such as GVGM. In contrast, Fig. 1i shows GOI represented by UGVs. It is identical to that of Fig. 1f except for the boundaries between shaded and non-shaded areas. This comparison demonstrates that UGVs are remarkably invariant to uniform changes

of image intensities, and thus can maintain the information on local gradient orientations under varying illuminations.

We make use of the robust feature UGVs,  $n_x$  and  $n_y$ , in place of image intensities  $g$  as expressed in (6), which corresponds to (1) in the gradient method (GM):

$$\begin{cases} n_x(x, y) = n_x(x + dx, y + dy, t + dt) \\ n_y(x, y) = n_y(x + dx, y + dy, t + dt) \end{cases} \quad (6)$$

where  $n_x$  and  $n_y$  indicate the UGVs in the  $x$  and  $y$  directions, respectively. By taking the first-order Taylor series expansion of the right hand sides, we can get the new OFCEs as shown in (7), which corresponds to (3) in the GM:

$$\begin{cases} n_{xx} u_1 + n_{xy} v_1 + n_{xt} = 0 \\ n_{yx} u_2 + n_{yy} v_2 + n_{yt} = 0 \end{cases} \quad (7)$$

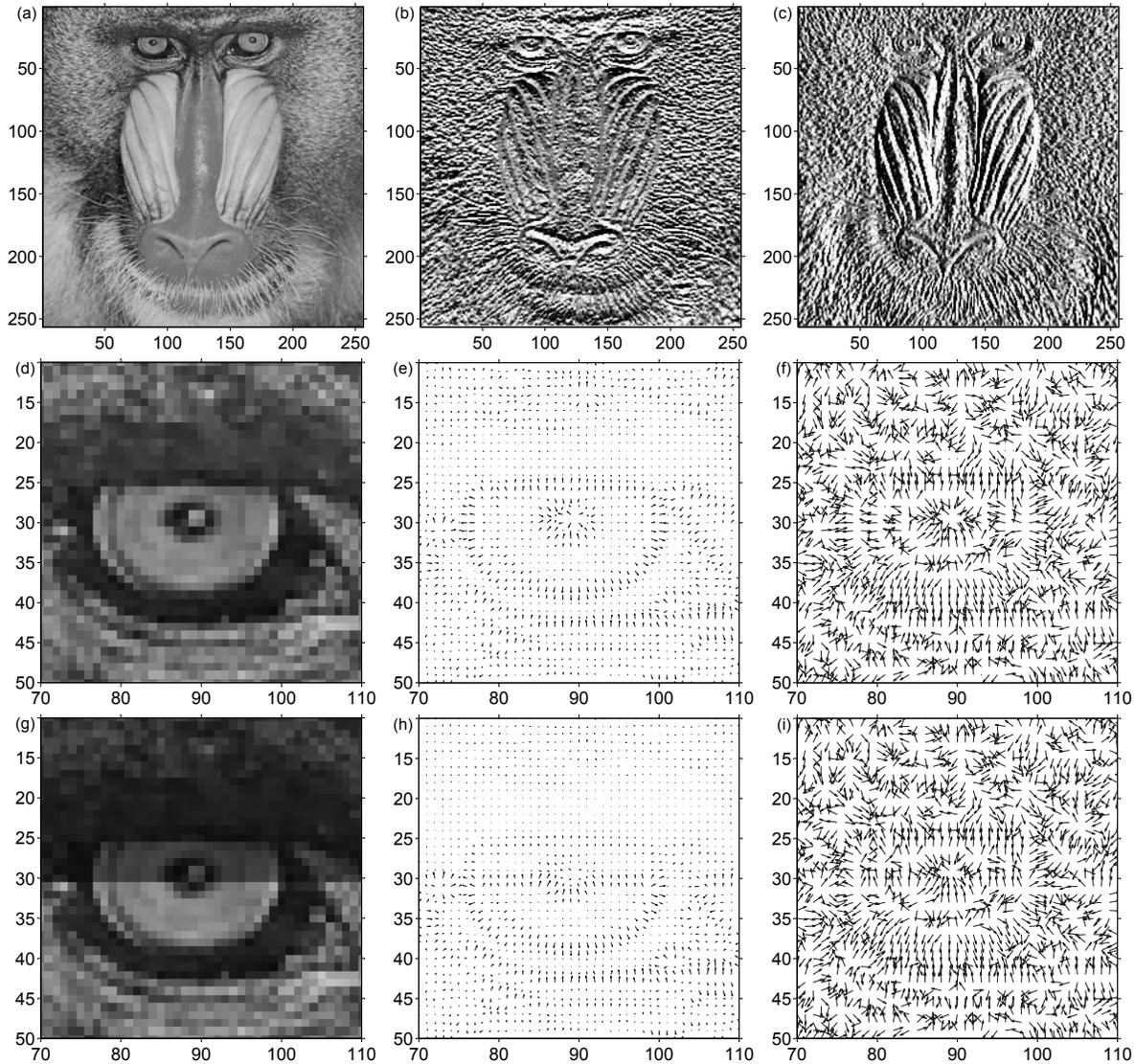
where the first subscript of each element denotes which of  $n_x$  and  $n_y$  components of the UGVs are used, and the second the direction of the partial derivative, either  $x$ ,  $y$ , or  $t$ . The subscripts 1 and 2 are used for defining the estimated optical flow on the patterns  $n_x$  and  $n_y$ , respectively. We have used the Sobel operators for computing the partial derivatives because they yield good approximations of the derivatives with minimal computation<sup>15</sup>. In place of (4), we then have the following two cost functions

$$\begin{cases} F_1 = \sum (n_{xx} u_1 + n_{xy} v_1 + n_{xt})^2 \\ F_2 = \sum (n_{yx} u_2 + n_{yy} v_2 + n_{yt})^2 \end{cases} \quad (8)$$

From (8), we obtain two motion estimates  $(u_1, v_1)$  and  $(u_2, v_2)$  using the least squares method as in the GM:

$$\begin{cases} u_1 = \frac{(\sum n_{xx} n_{xy})(\sum n_{xy} n_{xt}) - (\sum n_{xy}^2)(\sum n_{xx} n_{xt})}{(\sum n_{xx}^2)(\sum n_{xy}^2) - (\sum n_{xx} n_{xy})^2} \\ v_1 = \frac{(\sum n_{xx} n_{xy})(\sum n_{xx} n_{xt}) - (\sum n_{xx}^2)(\sum n_{xy} n_{xt})}{(\sum n_{xx}^2)(\sum n_{xy}^2) - (\sum n_{xx} n_{xy})^2} \\ u_2 = \frac{(\sum n_{yx} n_{yy})(\sum n_{yy} n_{yt}) - (\sum n_{yy}^2)(\sum n_{yx} n_{yt})}{(\sum n_{yx}^2)(\sum n_{yy}^2) - (\sum n_{yx} n_{yy})^2} \\ v_2 = \frac{(\sum n_{yx} n_{yy})(\sum n_{yx} n_{yt}) - (\sum n_{yx}^2)(\sum n_{yy} n_{yt})}{(\sum n_{yx}^2)(\sum n_{yy}^2) - (\sum n_{yx} n_{yy})^2} \end{cases}$$

It should be noted that motion estimates  $(u_1, v_1)$  are based on the pattern  $n_x$  while motion estimates  $(u_2, v_2)$  are based on the pattern  $n_y$ . Because there should be only one motion vector per block, we need to



**Fig. 1** (Right then down) (a) an 8-bit grey-scale image, gradient orientation patterns, (b)  $n_x$ , and (c)  $n_y$ ; (d) an 8-bit grey-scale subimage of (a), (e) the gradient vectors of (d), (f) the unit gradient vectors of (d); (g) partially shaded subimage, (h) the gradient vectors of (g), and (i) the unit gradient vectors of (g).

unify those two motion estimates into one. For this, we introduce a weighting factor that is dependent on the gradient orientation patterns<sup>12,16</sup>. The reliability of each motion estimate may be judged from the diversity of gradient orientations (directions) in its local area. The diversity of gradient orientations can be quantitatively measured by the two eigenvalues of the empirical estimator of the covariance matrix,  $N$ , between  $n_x$  and  $n_y$ ,

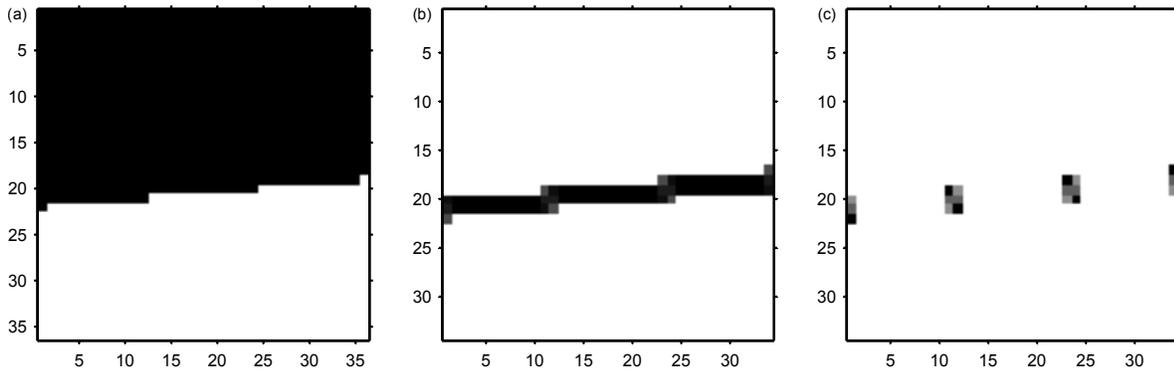
$$N = \begin{pmatrix} \sum n_x^2 & \sum n_x n_y \\ \sum n_x n_y & \sum n_y^2 \end{pmatrix}$$

where  $\sum$  indicates the integration area in a local 2-D

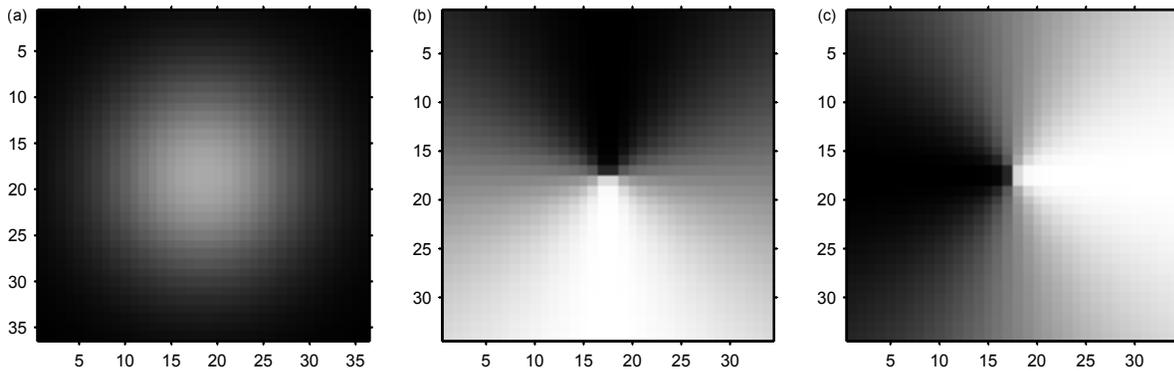
space. By applying principal component analysis to this matrix, we obtain two eigenvalues  $\lambda_1$  and  $\lambda_2$  where  $\lambda_1 \geq \lambda_2 \geq 0$ . We then use these eigenvalues as a weighting factor as in:

$$\begin{cases} u = \frac{\lambda_1}{\lambda_1 + \lambda_2} u_1 + \frac{\lambda_2}{\lambda_1 + \lambda_2} u_2 \\ v = \frac{\lambda_2}{\lambda_1 + \lambda_2} v_1 + \frac{\lambda_1}{\lambda_1 + \lambda_2} v_2 \end{cases} \quad (9)$$

Eq. (9) achieves the weighted average between the two motion estimates  $(u_1, v_1)$  and  $(u_2, v_2)$ . When the two eigenvalues hold such relationship as  $\lambda_1 \gg \lambda_2 \approx 0$ , the gradients in the block are close



**Fig. 2** (a) image block with a horizontal edge, (b) normalized vertical gradients  $n_x$ , and (c) normalized horizontal gradients  $n_y$ .



**Fig. 3** (a) image block with a 2-D Gaussian function, (b) normalized vertical gradients  $n_x$ , and (c) normalized horizontal gradients  $n_y$ .

to unidirectional and we can only estimate motion along the direction parallel to the gradients. Image blocks normally contain various directions of gradients. However, what if one direction of gradient is dominant, namely,  $\lambda_1 \gg \lambda_2 \approx 0$ . Fig. 2 shows such a pattern where the vertical gradient is dominant. Note that the estimation of the optical flow  $(u_1, v_1)$  is based on the pattern  $n_x$  shown in Fig. 2b while the estimation of the optical flow  $(u_2, v_2)$  depends on the pattern  $n_y$  in Fig. 2c. The two estimates are integrated as shown in (9).

It is obvious that the pattern  $n_x$  provides reliable information for estimating the vertical motion  $u_1$ . Thus a large weight should be assigned to the first term of the first equation. In contrast, the pattern  $n_x$  may contribute to a large error in estimating the horizontal motion  $v_1$  because of the aperture problem. Hence the first term of the second equation should be assigned a small weight. In this manner, the weight is adjusted automatically, depending on the relationship between the two eigenvalues. Meanwhile, Fig. 3a shows an

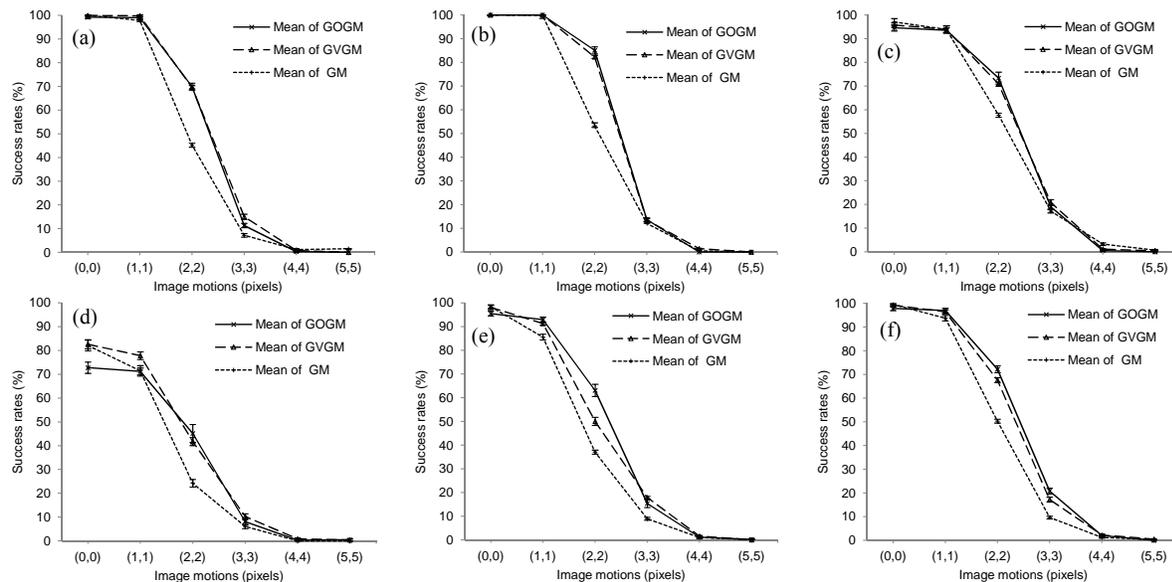
image block that has various directions of gradients, namely,  $\lambda_1 \approx \lambda_2 > 0$ . The patterns  $n_x$  (Fig. 3b) and  $n_y$  (Fig. 3c) are both useful for estimating vertical and horizontal motions. Thus we evaluate two motion estimates  $(u_1, v_1)$  and  $(u_2, v_2)$  nearly equally.

**RESULTS AND DISCUSSION**

For quantitative evaluations of the three motion estimation techniques, the GM, GVGM, and GOGM, we create six synthetic image sequences using six 8-bit grey-scale images of size 256 by 256 pixels. They are standard test images widely used in the image processing community, i.e., Lena, Girl, Peppers, House, Bridge, and Baboon images. Those images are used as the first frames and the second frames are generated by translating the first frames (test images) by a known amount of pixels.

**Under a constant lighting condition**

We first compare the performances of the GM, GVGM, and GOGM under a mildly noisy (40 dB)



**Fig. 4** (Right then down) successful motion estimation rates for mildly noisy (40 dB) (a) Baboon, (b) Bridge, (c) Girl, (d) House, (e) Lena, and (f) Pepper images under a constant lighting condition.

and constant lighting condition. The input image is partitioned to blocks of size 16 by 16 pixels. We estimate 14 by 14, i.e., 196 motion vectors within the image. It is relevant to note here that gradient-based methods are sensitive to high frequency in an image and an application of a certain low-pass filter is always necessary prior to motion estimation. Consequently, the outmost blocks are not used for motion estimation because they contain invalid pixels after the spatial filtering (convolution). The low-pass filter is a normalized Gaussian function where the sum of the coefficients in the filter is 1. The size of the filter kernel is chosen from  $(9 \times 9)$ – $(13 \times 13)$  pixels as they are found to be the appropriate sizes in our previous study<sup>12,17</sup>. For simplicity, the standard deviation of the normalized Gaussian function is set to be the half of the filter kernel.

We evaluate the rates of successful motion estimation for different amount of image motions. Estimated motions are considered successful when the deviations between the estimates and the known amount of motion (ground truth) are less than 0.5 pixels in both  $x$  and  $y$  directions. The amount of image motions is varied from (0, 0) to (5, 5) pixels. The simulation is repeated 10 times with different Gaussian noise patterns of the SNR 40dB every time. The means and standard deviations of the successful motion estimation rates are plotted against the varied image motions in Fig. 4. The GVGM and GOGM produce higher success rates on all the six image sequences, especially around

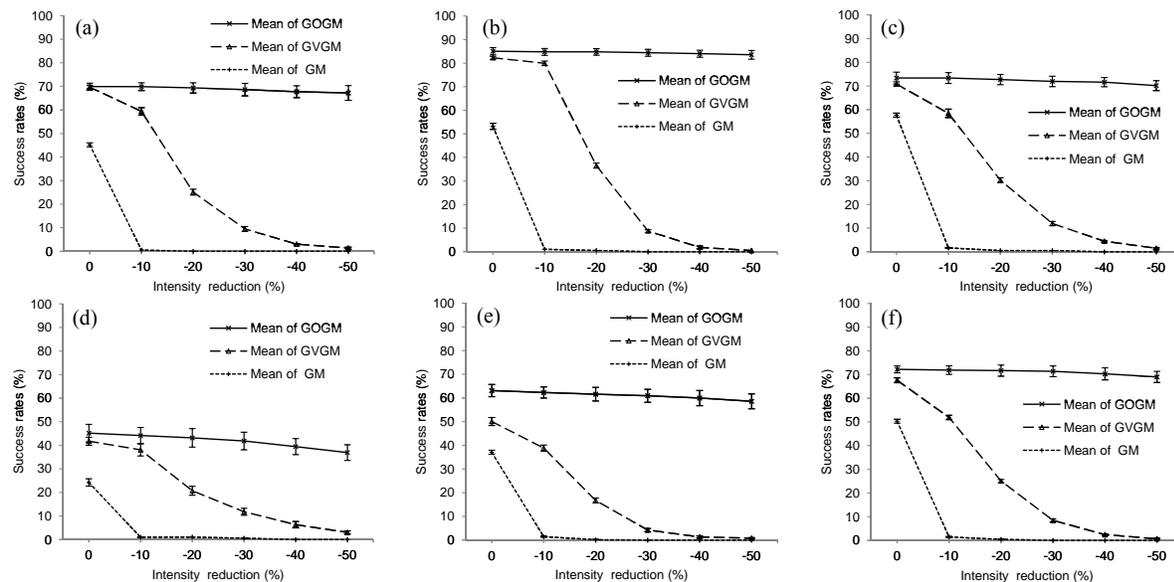
the motion (2, 2) pixels. Hence, we can state that the GVGM and GOGM outperform the GM under a mildly noisy and constant lighting condition.

#### Under a varying lighting condition

Next, the intensities of the second frames are uniformly reduced from 0–50%. Gaussian noise of the SNR 40dB is again added to every frame. The image motion is set (2, 2) pixels and the size of the low-pass filter mask is 11 by 11 pixels. The number of estimated motion vectors is the same as before, i.e., 14 by 14 vectors. Fig. 5 shows the successful motion estimation rates versus the amount of intensity reductions from 0% to 50%. It is obvious that the GM completely breaks down and the decline of the GVGM also clearly appears when the amount of multiplicative intensity reduction increases, while the GOGM performs motion estimation regardless of the variations of image intensities. This stark contrast clearly exhibits an advantage of the proposed method, GOGM.

#### Motion estimation on real image sequences

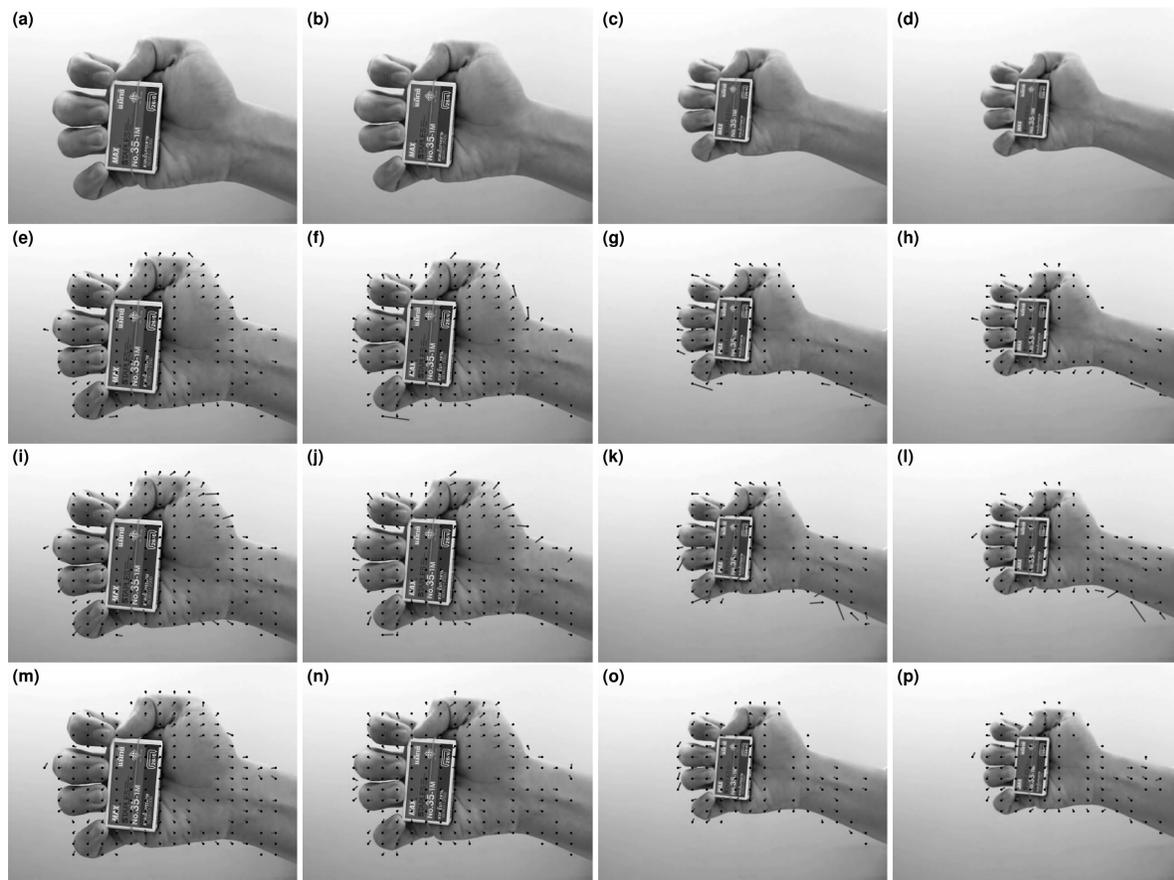
We compare the feasibility of the proposed method on real image sequences which contain local motions. The real image sequences are captured by a USB camera with resolution 480-pixel height and 640-pixel width. They are converted to 8-bit grey-scale images. The size of the block is set at  $24 \times 32$  pixels height and width, respectively. We calculate  $18 \times 18$



**Fig. 5** Successful motion estimation rates for mildly noisy (40 dB) (a) Baboon, (b) Bridge, (c) Girl, (d) House, (e) Lena, and (f) Pepper images when images intensities are varied from 0% to 50%.



**Fig. 6** (Right then down) part of the motion estimation results on a real image sequence with a horizontal translational motion under a constant lighting condition in the (a) 20th, (b) 23rd, (c) 32nd, and (d) 35th frames. Motion vectors estimated by (e-h) the GM, (i-l) the GVGM, and (m-p) the GOGM.



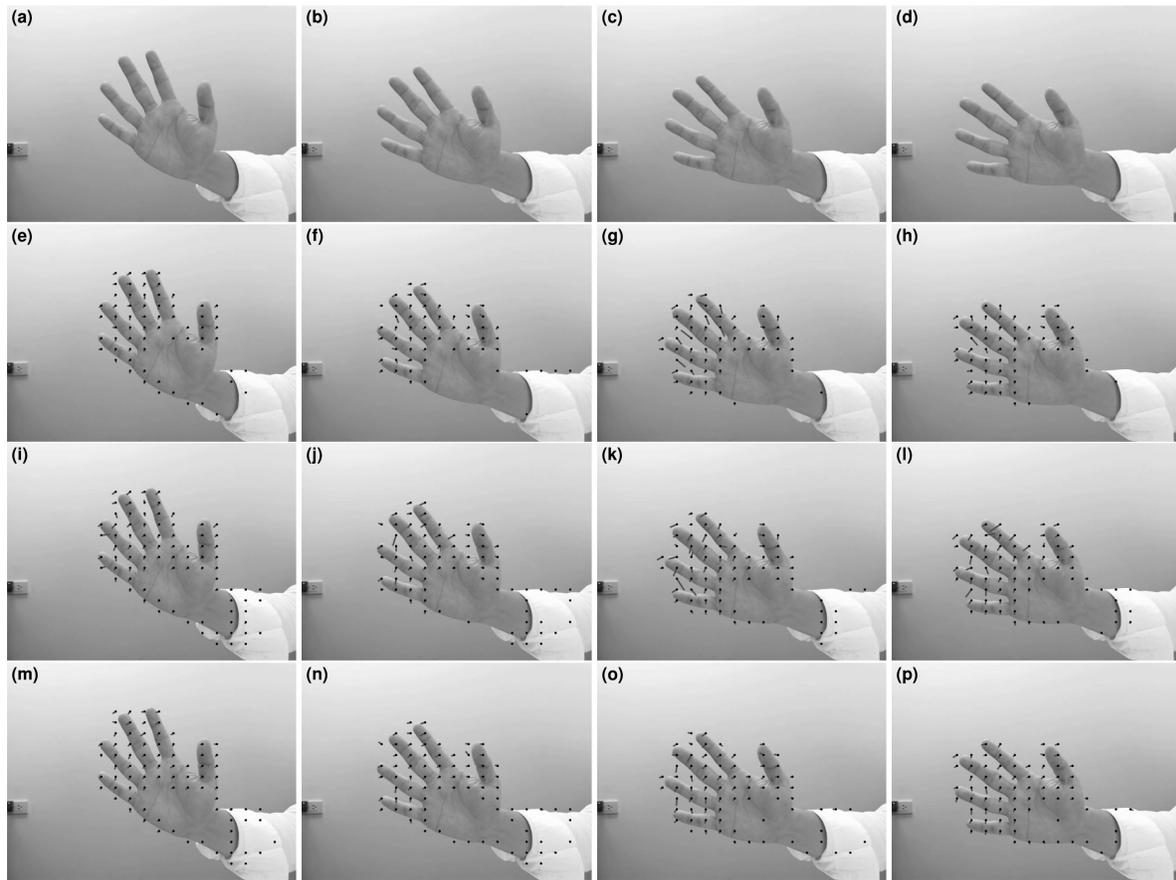
**Fig. 7** (Right then down) part of the motion estimation results of a real image sequence with zooming out under a constant lighting condition in the (a) 108th, (b) 110th, (c) 117th, and (d) 119th frames. Motion vectors estimated by (e–h) the GM, (i–l) the GVGM, and (m–p) the GOGM.

motion vectors each frame. The Gaussian low-pass filter of size  $13 \times 13$  pixels is used for smoothening input images. Figs. 6–8 show real image sequences (first row) together with the motion vectors estimated by the three techniques GM (second row), GVGM (third row), and GOGM (fourth row). Fig. 6 shows a translational motion while Fig. 7 and Fig. 8 show zooming and rotational motions under a constant lighting condition. Since there is no ground truth for the motions in these image sequences, the GOGM is compared with the GM and GVGM based on visual inspection. In this experiment, the flat areas like a wall are ignored by setting a gradient magnitude threshold, 500. From Figs. 6–8, we may state that the GM, GVGM, and GOGM are working reasonably well under constant illuminations. Close observation, however, reveals that the GM produces more erroneous motion estimations, especially around the boundaries of the human subject. These errors are due to the aperture problem when there is only one directional

gradient in the block. In this case, motion estimation is possible only along the direction perpendicular to the boundary and a large estimation error is expected in the direction tangent to the boundary. Those errors are well suppressed for the GVGM and GOGM owing to the weighted sum of the two motion estimates (9).

Meanwhile, Fig. 9 shows experimental results of motion estimating on the first real image sequence under irregular lighting. The image intensities of the odd numbers of frames in the sequence are reduced uniformly by 30%. It is obvious that the conventional approaches, the GM, cannot produce reliable motion estimates at all under the time-varying lighting condition (Fig. 9e–h). The GVGM almost produce the reliable motion estimated vectors (Fig. 9i–l) but the GOGM performs very well even under such irregular lighting condition (Fig. 9m–p).

Finally, the computation times of the GM, GVGM, and GOGM are 41.43 millisecond (ms), 57.69 ms, and 59.78 ms per frame, respectively.

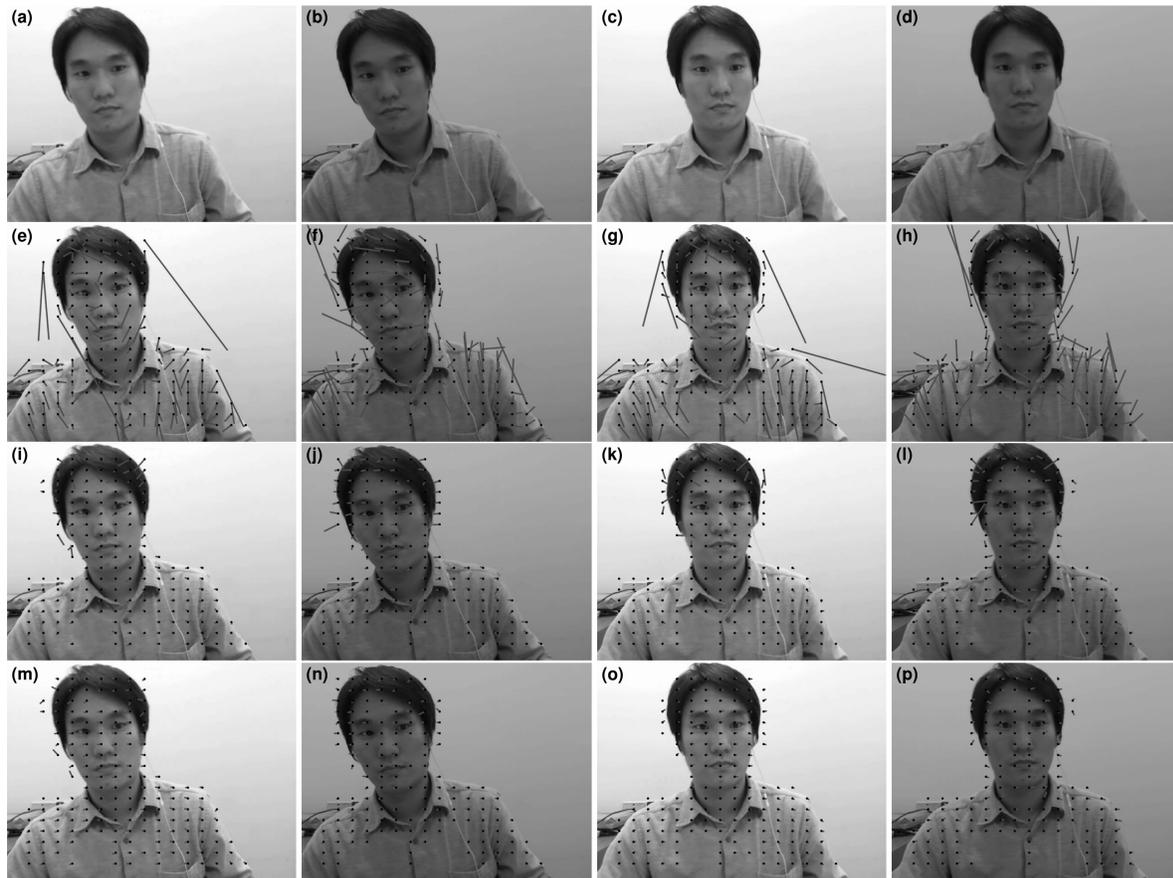


**Fig. 8** (Right then down) part of the motion estimation results of on a real image sequence with a rotational motion under a constant lighting condition in the (a) 63rd, (b) 72nd, (c) 76th, and (d) 80th frames. Motion vectors estimated by (e–h) the GM, (i–l) the GVGM, and (m–p) the GOGM.

These are the CPU times required for convoluting the Gaussian low-pass filter of size  $13 \times 13$  pixels and calculating  $18 \times 18$ , namely, 324 motion vectors within the frame of size  $480 \times 640$  pixels. Both methods are implemented in the Microsoft Visual C++ 2008 with the OpenCV library<sup>5</sup>. The CPU times are measured on a standard notebook PC with the Intel Core2 Duo CPU P8600 2.4GHz and 4 GB RAM. The longer time required for the GOGM is due to the extra computations for normalizing gradients, obtaining the eigenvalues of the gradient covariance matrices, and so on. The CPU time of the GM, 41.43 ms/frame, is equivalent to 24.14 frames per second (fps), which is nearly as fast as the PAL TV standard, 25 fps. Furthermore, the CPU time of the GVGM, 57.69 ms/frame, is equivalent to 17.33 fps. Lastly, the CPU time of the GOGM, 59.78 ms/frame, equals to 16.73 fps, which is as fast as many commercial USB cameras (15 fps). Thus we can state that both methods already work at a video rate without any spatial down-sampling.

## CONCLUSIONS

This paper presents a robust optical flow estimation technique that is based on the spatiotemporal gradient method (GM) with a local smoothness constraint. We propose to use gradient orientation information (GOI) instead of conventional image features such as intensities and gradients. Since GOI is stable under varying illuminations, the proposed method performs well regardless of varying lighting conditions. We utilize the GOI in the form of unit gradient vectors rather than directly using the gradient directions  $\theta$  for computational efficiency. An added advantage of the proposed method, gradient orientation based gradient method or GOGM, is that it is more robust to the aperture problem. Since two motion estimates are averaged with an adaptive weight that is determined by the diversity of gradients in a local region (block), the method effectively avoids producing large estimation errors that are common in the traditional gradient-



**Fig. 9** (Right then down) part of the motion estimation results on a real image sequence in the (a) 20th, (b) 23rd, (c) 32nd, and (d) 35th frames where the odd numbers of frames are uniformly darkened by 30%. Motion vectors estimated by (e–h) the GM, (i–l) the GVGM, and (m–p) the GOGM.

based methods. The GOGM can estimate as many as 324 motion vectors at the rate of 16 frames per second, though it is slower than the traditional method GM and GVGM because of the extra computation for normalizing image gradients. As future work, we plan to employ a strategy for reducing the computation time to achieve the speed of the PAL (25 fps) and NTSC (30 fps) video standards.

*Acknowledgements:* This study is supported by research grants from the Thammasat University Research Fund and the National Research University Project of Thailand Office of Higher Education Commission.

## REFERENCES

1. Horn BKP, Schunck BG (1981) Determining optical flow. *Artif Intell* **17**, 185–203.
2. Kearney JK, Thompson WB, Boley DL (1987) Optical flow estimation: an error analysis of gradient-based methods with local optimization. *IEEE Trans Pattern Anal Mach Intell* **9**, 229–44.
3. Lucas BD, Kanade T (1981) An iterative image-registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, pp 674–9.
4. Reed TR (2005) *Digital Image Sequence Processing Compression and Analysis*, CRC Press.
5. Bradski G, Kaehler A (2008) *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st edn, O'Reilly Media, Inc., Sebastopol.
6. Kim YH, Martínez AM, Kak AC (2005) Robust motion estimation under varying illumination. *Image Vis Comput* **23**, 365–75.
7. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: *Computer Vision – ECCV 2004: Proceedings of the 8th European Conference on Computer Vision*, part IV, pp 25–36.
8. Schuchert T, Aach T, Schar H (2010) Range flow

- in varying illumination: algorithms and comparisons. *IEEE Trans Pattern Anal Mach Intell* **32**, 1646–58.
9. Baker S, Roth S, Scharstein D, Black MJ, Lewis JP, Szeliski R (2007) A database and evaluation methodology for optical flow. In: *ICCV 2007: Proceedings of the IEEE 11th International Conference on Computer Vision*, pp 1–8.
  10. Burgi PY (2004) Motion estimation based on the direction of intensity gradient. *Image Vis Comput* **22**, 637–53.
  11. Kondo T, Yan H (1999) Automatic human face detection and recognition under non-uniform illumination. *Pattern Recogn* **32**, 1707–18.
  12. Boonsieng P, Kondo T, Kongprawechnon W (2011) Unit gradient vectors based motion estimation techniques. *ECTI-EEC* **9**, 246–54.
  13. Jain R, Kasturi R, Schunck BG (1995) *Machine Vision*, McGraw-Hill, Inc., Singapore.
  14. Kondo T, Boonsieng P, Kongprawechnon W (2008) Improved gradient-based methods for motion estimation in image sequences. In: *Proceedings of the SICE Annual Conference*, pp 1120–3.
  15. Gregson PH (1993) Using angular dispersion of gradient direction for detecting edge ribbons. *IEEE Trans Pattern Anal Mach Intell* **15**, 682–96.
  16. Kondo T, Kongprawechnon W (2009) Robust motion estimation methods using gradient orientation information. *Sci Asia* **35**, 196–202.
  17. Boonsieng P, Kondo T (2007) Comparative study of motion estimation techniques: the gradient method and structure tensor method. In: *Proceedings of the International Workshop on Advanced Image Technology*.