Paper

Robust Computation of Optical Flow using Orientation Code Matching

Farhan Ullah* Non-member
Shun'ichi Kaneko* Member

A robust matching scheme for computing optical flow within a sequence of grayscale images is proposed. The technique employs the gradient information in textured images for extracting features in the form of orientation codes, which are then used for matching. The proposed method has been found to be robust in cases of matching under different ill-conditionings especially illumination variations. We utilize its robustness to compute optical flow in cases where illumination fluctuation is a problem and matching pixel brightness can introduce errors. Results of computation of optical flow field on real world scenes in the cases of translation, rotation and zooming have been presented and compared with other region matching techniques.

Keywords: object search, matching, robustness, orientation code, optical flow

1. Introduction

Computation of optical flow (apparent relative motion between 3-D objects and camera obtained as its projection on image plane) is an area of active research with many applications like robotics, surveillance, television coding, etc. Approximations to image motion are obtained by computation of image velocity fields by finding the optical flow vectors from consecutive images of a sequence. The computed velocity fields can then be used to extract information about 3-D motion and scene structure [1][2]. The following various methods for computing optical flow have been proposed: gradient based methods introduced by Horn and Schunck [3] and its many variations [4]~[7], frequency-based filtering methods [8][9] and correlation-based methods [10]~[12]. Detailed surveys of the various methods and issues involved are available in references [13][14]. A comparison of performance of most popularly known optical flow computation techniques is provided in Barron et al. [15].

Differential methods require algorithms to minimize some functionals for computing optical flow since their basic optical flow constraint equations are ill-posed. Correlation-based methods search for the closest matching patterns in successive images within the vicinity of the point for which the velocity vector is desired. These methods are inherently stable since the displacements for all pixels in a region which is used for computing optical flow for a certain pixel are similar and largely overlap with the region used for finding flow vectors for adjacent pixels; thus no additional constraints are necessary for smoothing the flow vectors. However, direct use of image brightness for establishing a match can cause erroneous results since brightness can vary in real images which may be attributable to many factors, e.g. illumination fluctuations due to the variations commonly encountered in the indoor lighting, shadowing effects by different objects in a cluttered environment, highlighting etc.

Although the optical flow estimates the 2-D velocity vectors in image frames separated by very small (differential) time intervals, many physical systems may exist for which it may be desirable to increase the interval from the order of milliseconds to the order of minutes. For example, in processing a pre-recorded video for tracking objects, the interval between image frames to be used for computing optical flow may be selected based on a-priori information about expected object speed. There may be cases when the apparent speed of certain object is too slow to justify the image capture rate at very high frequencies such as tracking a marine vessel from shore or the tracking of plant growth [16]. For such situations, the larger intervals between two successive frames may sometimes cause significant brightness variations between the two images.

In indoor environments, the movement of some glossy object at some instant may cause some part of it to be highlighted resulting from reflection of the indoor illumination. Similarly, a fast moving light source may cause a shiny object to illuminate suddenly at different points during a sequence capture. Yet another source of sudden variation in illumination can be the indoor fluorescent light which involves flickering due to periodic ionization of the gas inside the tube. This flickering, although not perceptible by human eye, can be captured by cameras sensitive enough to such flickering resulting in brightness variations between frames.

Some pre-filtering has been proposed for computing optical flow by matching-based methods such as sum of squared difference (SSD) or the sum of absolute differences (SAD), like construction of Laplacian pyramid [11]

*Manufacturing Environment Engineering, Hokkaido University,
Kita 13, Nishi 8, Kita-ku Sapporo 060-8628

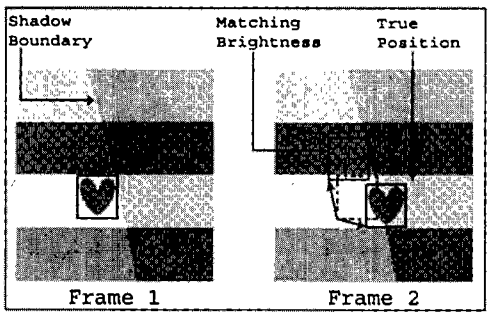


Fig. 1. Illustration of matching with occurrence of shadow.

to make them more stable in these situations.

In this study, we propose the use of a robust matching scheme called the orientation code matching (OCM) [17] for estimating displacement vectors within a sequence of images. This method has been found to be robust in cases of matching under different ill-conditionings like varying illumination and partial occlusion. We utilize the robustness of this method to compute the optical flow vectors in cases where illumination fluctuation is a problem and matching pixel brightness can introduce errors. The problem of occlusion is not attempted in this study since we use small subtemplates for finding their closest matches in the next frame, while for handling the problem of occlusion, relatively large templates are required which is not suitable for computing optical flow field.

The paper is organized as follows: Section 2 describes the use of matching algorithms for finding optical flow and formalizes the definition of OCM and its robustness to brightness variations. A simple algorithm for computing flow vectors and implementation are given in Section 3. Experimental results are given in Section 4 and the conclusions are given in Section 5.

2. Optical Flow Estimation by Region Matching

Region-based matching methods like SSD use small patches of predefined size from one frame to search its best possible match within the vicinity of the same point in the next frame in order to estimate the optical flow. In general, for the case of rigid body motion nearly parallel to the image plane, this method is quite realistic since motion vectors will be similar for all the pixels constituting the part of an object [18]. However, in real environments, variations in image brightness may be caused by shadows or environmental lighting fluctuations and as a result, the computed vector field may not be well-posed. An example is illustrated in Fig. 1 in which a region from the first frame is searched in the second frame for the closest match; a shadowing effect causes the brightness of some part of the target to change as it moves across the shadow boundary. Computation of optical flow vectors using brightness matching only can lead to erroneous results. For handling such problems of matching under varying illuminations, we propose the use of orientation code matching (OCM).

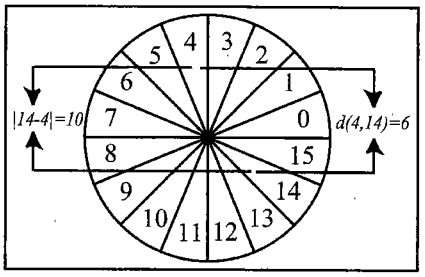


Fig. 2. Illustration of orientation codes and the definition of error function $d(\cdot)$.

2.1 Definition of OCM

OCM is defined as matching of orientation codes of two images which are to be compared for similarity evaluation. For discrete images, the orientation codes are obtained as quantized values of gradient angle around each pixel by applying some operator for computing horizontal and vertical derivatives like Sobel operator, and then using the tan^{-1} function on their ratio to obtain the gradient angle. The orientation code for a pixel location (i,j), letting $\theta_{i,j}$ be a gradient angle, for a preset sector size of Δ_θ is given as:

$$c_{i,j} = \begin{cases} \left[\frac{\theta_{i,j}}{\Delta_\theta}\right] & : \ |\nabla I_x| + |\nabla I_y| > \Gamma \\ L & : \ \text{otherwise} \end{cases}$$

A separate code L is assigned for low contrast regions (defined by the threshold level Γ), for which it is not possible to compute the gradient angles. For all the experiments, we used 16 orientation codes, which was found to be the best through experimental evaluation. The threshold value Γ plays an important role in suppressing the effects of noise and has to be selected according to the problem at hand; too large values can cause the suppression of texture information. We used a small value of 10 for Γ which was good for most of our experiments; however, for the images containing uniform stationary regions involving illumination variations, better results were noted with larger values.

A dissimilarity measure is defined as the summation of the difference between the orientation codes of the corresponding pixels of the two regions being matched. The cyclic property of orientation codes is used for finding the difference. If $O1$ and $O2$ represent the orientation code images of the two regions, then the dissimilarity function between them is given by:

$$S = \frac{1}{M} \sum d(O1(i,j), O2(i,j))$$

where M is the total number of pixels used in the match and $d(\cdot)$ is the error function based on an absolute difference criterion.

Let $\mathbf{C} = \{0, 1, \cdots, N-1\}$ be the set of orientation codes for N orientation codes, then the error function can be written as:

$$d(a,b) = \begin{cases} min\{|a-b|, N-|a-b|\}: & (a,b) \in \mathbf{C} \\ \frac{N}{4} & : a = L \text{ xor } b = L \\ 0 & : \ a = b = L \end{cases}$$

For all of the experiments, we used 16 orientation codes

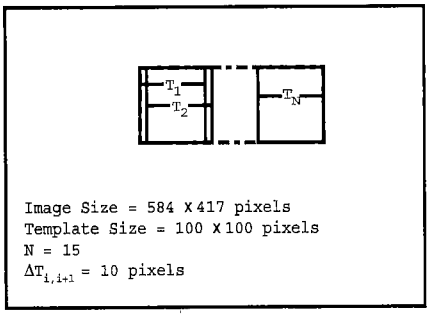
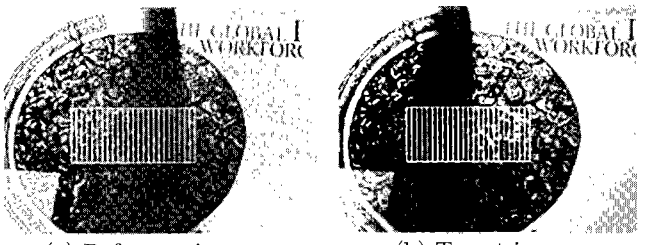


Fig. 3. Illustration of reference extraction.

corresponding to a sector width Δ_θ of $\frac{\pi}{8}$. Illustration of orientation codes is shown in Fig. 2. Cyclic property of the codes utilized for defining the error function $d(\cdot)$ is noted in contrast with the absolute difference.

2.2 **Robustness of OCM to Brightness Changes** Any similarity measure simply based on aggregation of brightness difference between a reference image and object images, such as SSD or SAD, is not invariant for even slight change in illumination. Normalized cross correlation (NCC), however, is known to be invariant against global change in illumination by a constant magnifier and an offset, but it is also known to be quite sensitive to local fluctuation in illumination that is unevenly distributed in the scene. We show the robustness of OCM against both the global and local illumination changes. This is because the gradient operation, such as Sobel operator, is defined in local domain, say a 3×3 region, maintaining phase angles unchanged even under non-uniform illumination. Let a represent the magnification and the constant offset be represented by b, then any observed brightness g corresponding to the actual pixel brightness f can be presented as: $g = a \cdot f + b$. A fraction of f_y/f_x is transformed to the phase angle θ through the relation, $\theta = \tan^{-1}(f_y/f_x)$. From the above eqution, $g_y = a \cdot f_y$ and $g_x = a \cdot f_x$, and consequently the fraction with respect to observed brightness g_y/g_x is the same as f_y/f_x.

2.3 **Verification of Invariance of OCM** For verification of the above formalization, we performed experiments involving a pair of images of a magazine cover. A number of overlapping square templates were extracted from a reference image as illustrated in Fig. 3. These templates were matched in the same position in the target image which is identical to the reference image except for a shifted shadow cast over the scene. The template-target pairs had varying extents of areas covered by the shading. The pair of images is shown in Fig. 4 with marking with squares showing the extracted templates in the first image and the corresponding targets in the second one. Values of evaluation functions were noted for OCM, SSD and NCC and the plots of dissimilarity values for the three are shown in Fig. 5; (for the sake of consistency, we changed the correlation curve of NCC into dissimilarity curve as well). As can be seen from the plots, variation in dissimilarity function was minimal in the case of OCM compared with both



(a) Reference image. (b) Target image.

Fig. 4. Matching in shifted shading.

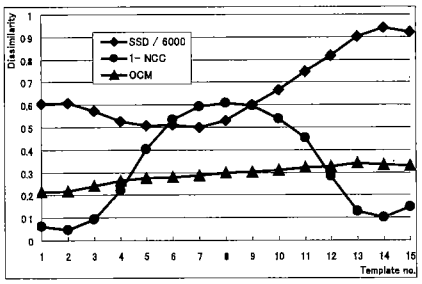


Fig. 5. Plots of variations in dissimilarity for Fig. 4. The scaling factor 1/6000 in SSD plots is introduced so that the maximal SSD value is just under 1 and the plot corresponding to NCC show the values subtracted from 1.0 to make it a dissimilarity curve like the other two.

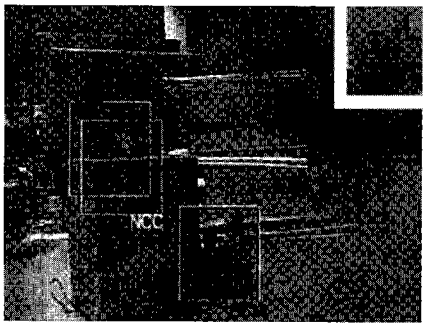
SSD and NCC. Correlation values of NCC are rather large (represented by smaller values on the dissimilarity curve) for the pairs that are located in the right and the left regions and consist of the images of overall shading in either component. But they are degraded at around the center region where each component image has local fluctuation in brightness. For the case of SSD, the curve has the tendency similar to that of NCC. In contrast with them, error values of OCM are rather low and their variations are prominently small and vary between 0.2 and around 0.35. This variation can be attributed to additive and multiplicative noise in the images.

As another measure of stability in the cases of illumination fluctuations, each extracted template from reference image was searched globally within the second one. OCM was successful for all 15 cases, while NCC and SSD reported 1 and 8 mismatches, respectively.

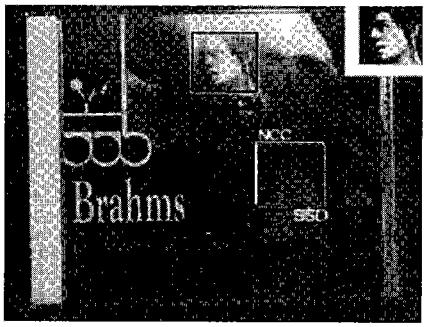
Matching results from some more real image data are shown in Fig. 6, where the template objects (shown as insets at the upper right corners on the corresponding scene images) undergo illumination variations due to shading by a nearby object (Fig. 6(a)) or appear highlighted due to surface reflection (Fig. 6(b)). Matching results (marked by rectangles as the best matches for each method) show the inherent stability of OCM in comparison with SSD or NCC.

3. OCM for Optical Flow Estimation

The optical flow vector (u, v) for a pixel is computed by selecting a small subtemplate around the position, say (m, n), from the first image and then searching its closest match in the second image inside a window of



(a) Searching for toy under shadow.



(b) Highlighted CD jacket.

Fig. 6. Matching result for ill-conditioned images.

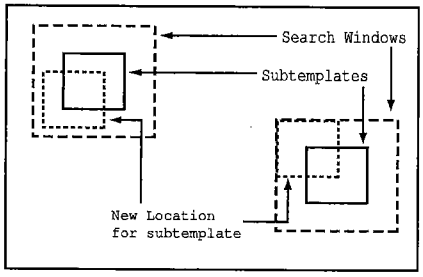


Fig. 7. Illustration of region-matching based optical flow computation.

pre-defined size within the neighbourhood of the same location. An illustration is given in Fig. 7 for the above-mentioned procedure.

If $O1_{m,n}$ and $O2_{m,n}$ are the corresponding orientation code images for the images involved in the computation, then the computation of the optical flow vector at the pixel position (m, n) can be expressed as:

$$(u, v)_{m,n} = \arg\min_{x,y} S_{m,n}(x, y)$$

$$S_{m,n}(x, y) = \frac{1}{M} \sum_{i,j} d(O1_{m,n}(i, j), O2_{m+x,n+y}(i, j))$$

$$\forall (x, y) \in W$$

where W is a window, centered at the origin, whose size is pre-selected to accommodate the maximum expected displacement of regions within image frames.

3.1 **Implementation** In order to compute the optical flow efficiently using the OCM based search, a table can be constructed for reference at run time instead of time-consuming mathematical and logical operations. If we set L (the code assigned in place of the orientation code to pixels corresponding to low contrast regions) equal to N (maximum number of orientation codes) then the error function $d(\cdot)$ can be expressed in the form of an $(N+1) \times (N+1)$ table based on the definition given in Section 2.1. The table, shown as Table 1, is like a symmetric matrix and all diagonal elements are zero. The last row and column (except the last element) represent the match in low contrast region. As a result, the computation of the error function reduces to table reference as $d(a, b) = OCTable[a, b]$.

Table 1. Orientation code differences.

	0	1	· ·	N/2 - 1	N/2	N/2 + 1	· · ·	N-1	N
0	0	1	· · ·	N/2 - 1	N/2	N/2 - 1	· · ·	1	N/4
1	1	0	· · ·	N/2 - 2	N/2 - 1	N/2	· · ·	2	N/4
2	2	1	· · ·	N/2 - 3	N/2 - 2	N/2 - 1	· · ·	3	N/4
· · ·				· · ·				· · ·	
N-2	2	3	· · ·	N/2	N/2 - 1	N/2 - 2	· · ·	1	N/4
N-1	1	2	· · ·	N/2 - 1	N/2	N/2 - 1	· · ·	0	N/4
N	N/4	N/4	.....					N/4	0

4. Experiments

Some experiments were performed for checking the effectiveness of OCM based computation of the optical flow.

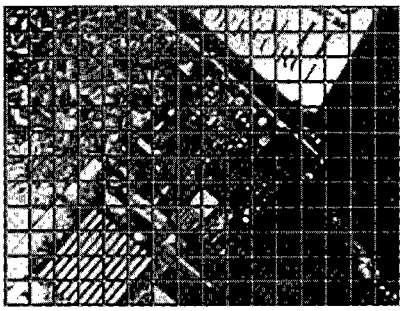
This study focuses on the computation of optical flow for the cases when there is a possibility of illumination variation between the frames involved as can be expected in many situations like those mentioned earlier in Section 1. For achieving such effects, time interval between images used for the experiments was not fixed and a few frames were skipped after capturing the image sequence before performing actual computation in order to increase the time interval between the two images. This helped us in performing better qualitative evaluation of the OCM based optical flow computation in comparison with other commonly used matching techniques viz. SDD and NCC. For all the experimental images shown here, a grid has been superimposed to increase the visibility of the difference between the two images. Details of the experimental setup and computer setup are given in Table 2

4.1 **Translation** In the experiment described here, a camera was set to view a scene containing some magazines and an object (a video card). A normal electric lamp was used for providing the lighting conditions in the scene. The camera was translated parallel to the plane containing objects of interest causing the dominant apparent motion to be along the axis of camera displacement. A shadow was cast on the scene at the same time to provide effects of illumination variations.

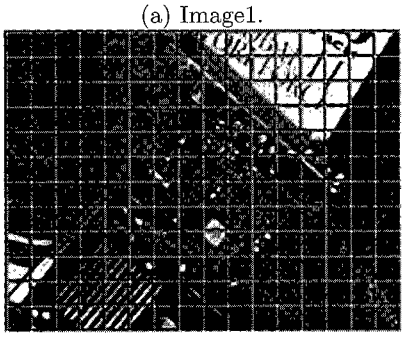
The results obtained by using OCM were compared with those obtained by using SSD and NCC. The im-

Table 2. Experimental setup.

Cameras	JAI CV-M10BX CCD Progressive Scan Monochrome (lens = 8mm focal length) & Victor GR-DVL7 digital video
Capture Boards	Leutron Picport Framegrabber & Adaptec AHA-8945
Image type	8-bit gray scale
CPU	Pentium-III, 1.0 GHz
OS	Windows 2000
Frame size	320 × 240 pixels
Subtemplate size	15 × 15 pixels



(a) Image1.

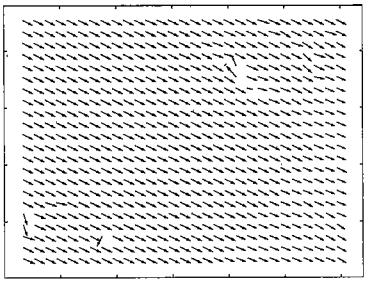


(b) Image2.

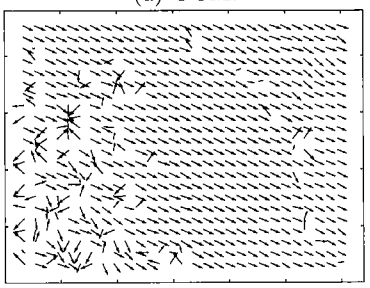
Fig. 8. Pair of images used in the experiment [19] (shading was added in image2 along with motion).

ages used in the experiment are shown in Fig. 8 and the resultant flow vectors are shown in Fig. 9 for the three methods used for comparison. Since the displacement vectors are a function of depth of objects in the scene, the ground truths are not easy to compare. We use the distribution of angles obtained for all the pixels for comparing the actual results since it is generally independent of depth information in the case of camera translations. The three distributions are shown in Fig. 10. The correct value can be assumed to correspond to the peak value, which was about -26.5 degrees in this case.

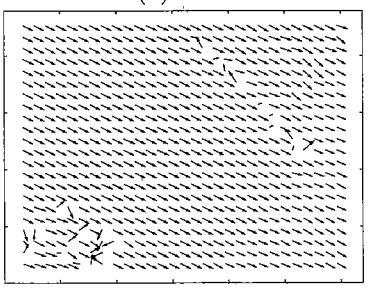
Since the optical flow vectors were computed to pixel accuracy only, the results are rounded to the nearest pixel position for the flow vectors. As can be seen from the results in Fig. 9, OCM had a rather smooth velocity field corresponding to the regions containing sufficient texture, whereas SSD-based computation had many errors in the regions affected by the lighting; unaffected regions have the same results as those obtained by OCM. NCC had better results in comparison with SSD, but it also suffered from some errors in the affected region.



(a) OCM.



(b) SSD.

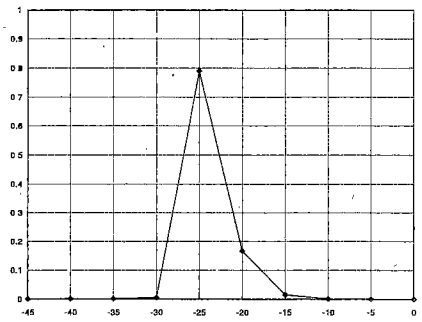


(c) NCC.

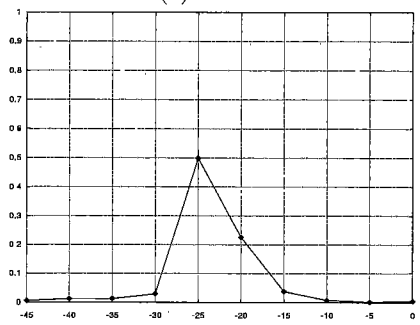
Fig. 9. 2-D flow vectors as obtained by using OCM, SSD and NCC.

4.2 **Rotation** While region matching techniques are not very robust for searching the objects rotated by arbitrary angles, there are some tolerance limits for angle of rotation within which the object may be located even if matched directly without causing rotation of one to align with the other. Also, if the range of search is confined to within the vicinity of the expected location of occurrence of the object, the chances of locating the correct position improve; the case of computation of optical flow is similar, where a small region from one frame is searched inside a window within the neighbourhood of the previous location. OCM can also be used for finding the correct position of the target object if the relative angle difference between the template and the target is not too large to be outside the tolerance limits for OCM. This characteristic of the region matching techniques can be used for finding the optical flow field for the cases of rotating flow fields as as well.

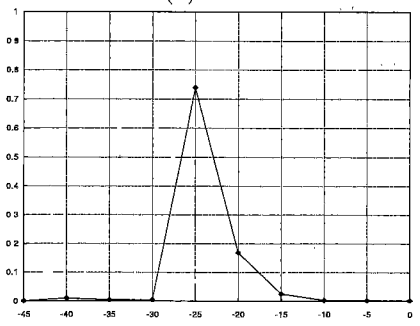
For the experiment involving computation of optical flow for rotated objects, a glossy magazine cover was rotated freely to create the effect of rotation. The motion was confined to 2-D rotation; however, since the movement was manual, a small perspective tilt during the rotation caused some part of the magazine cover to reflect the illumination, thereby causing a highlighting effect. The image pair is shown in Fig. 11 and the re-



(a) OCM.

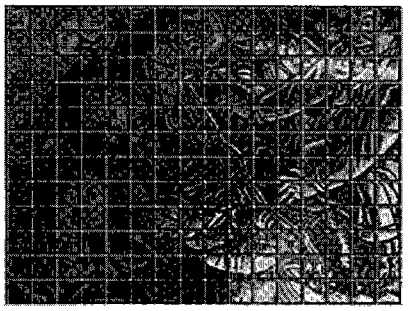


(b) SSD.

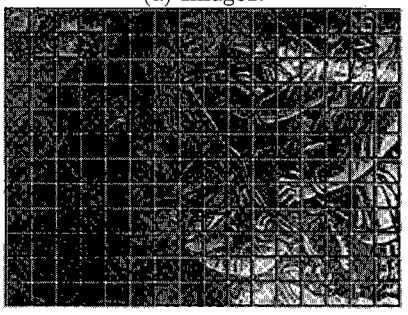


(c) NCC.

Fig. 10. Normalized histograms of angles obtained.



(a) Image1.



(b) Image2.

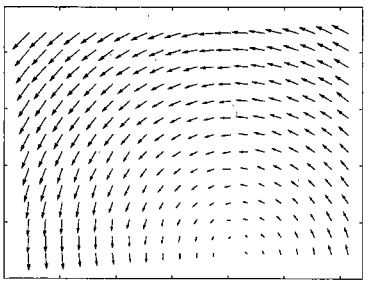
Fig. 11. Pair of images used in the rotation experiment.

sults by using the three methods are shown in Fig. 12. As can be seen from Fig. 12(b), SSD based computation had many errors in the region affected by highlighting.

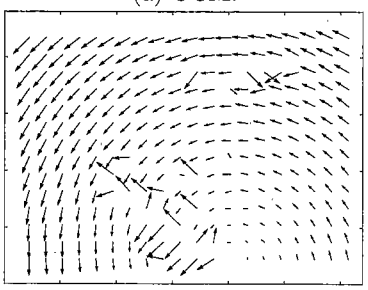
4.3 **Zooming** For the case of zooming, similar reasoning as in the case of rotation can be used for assuming that minor variation in scale can also be handled by region matching techniques, especially if the search is within the neighbourhood of the original subtemplate location.

In the experiment, an object was moved along the optical axis to create the zooming effect; the image pair is shown in Fig. 13. Results by using the three methods are shown in Fig. 14. Small variation in the brightness between the two frames caused the SSD based computation to make some errors during the computation around the regions where brightness variation is more conspicuous. OCM had better overall results except for the regions in the first frame which undergo occlusion in the second due to the looming motion of the object.

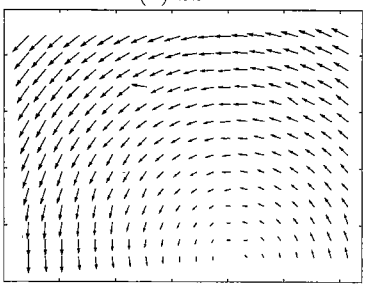
4.4 **Computation Time** The overall computation time depends on the sizes of the subtemplate and the search window used for computing the flow field. We noted the total processing time for the three method used here. For the sake of efficiency, we constructed tables for performing squares of differences, multiplication



(a) OCM.

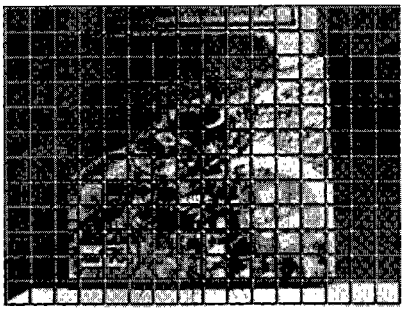


(b) SSD.

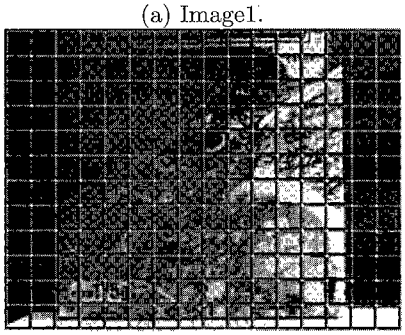


(c) NCC.

Fig. 12. 2-D flow vectors as obtained by using OCM, SSD and NCC for Fig. 11.

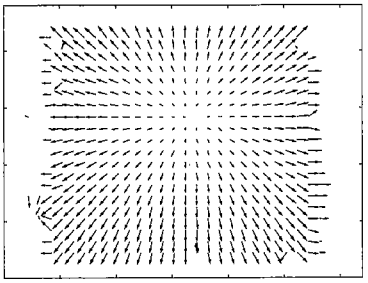


(a) Image1.

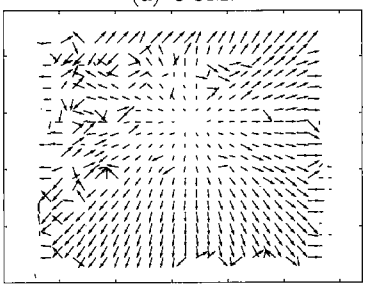


(b) Image2.

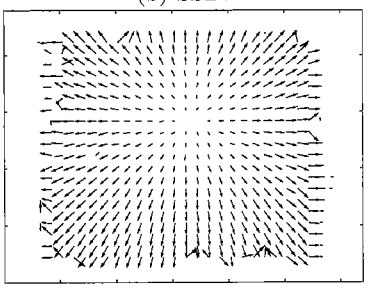
Fig. 13. Pair of images used in the zooming experiment.



(a) OCM.



(b) SSD.



(c) NCC.

Fig. 14. 2-D flow vectors as obtained by using OCM, SSD and NCC for Fig. 13.

and squares for all the gray values in a fashion similar to Table 1 in order to speed up the computation for SSD and NCC as well; however the computation for NCC involved floating point operations, which made the search slower in comparison with both SSD and OCM. For the images used in the experiment in Fig 8, the total computation time for dense flow field for 300×220 region and the search window size of 35×35 was found to be 136.8, 146.7 and 288.7 seconds by using OCM, SSD and NCC respectively. As can be noted, the computation time for OCM was almost same as SSD; the preprocessing involved for computation of the two OC images took about 80 milliseconds in total. The small difference in favor of the OCM may be attributed to the smaller size of lookup table which made the access time shorter for the computer system used in the experiment.

5. Conclusions

Application of OCM for computing optical flow has been introduced for situations where lighting conditions change rapidly, thereby causing erroneous result in the conventional matching. Such problems may arise in situations when the time interval between two successive frames is large enough to make the significant brightness variation more likely. Comparative analysis of the results obtained by OCM on the real world images show its reliability for computing optical flow for the cases of translation, rotation and zooming. For the case of translation, it was possible to compare the obtained results with the ground truths since the angle for the whole optical flow field was almost same. In general, results by OCM and NCC were very close but in terms of computation time, OCM has an advantage with almost half the time compared to NCC.

Low contrast regions are one of the source of problem in any optical flow computation strategy in general; however, for the region matching based methods, the size of the subtemplates can be varied optimally to increase the likelihood of including some texture which may be helpful in discrimination. The increase in subtemplate size may increase the processing time required for computing the optical flow vectors and at the same time, the effective domain of computable vectors may be reduced due to the exclusion of some boundary points necessary for accommodating the search window of the subtemplates.

From the point of view of real time implementation, OCM based computation involves an additional step of pre-processing for computing the orientation codes from gradient angles; however, with the availability of hardware implementations of gradient operators like Sobel, computing optical flow by OCM can also be implemented efficiently in real time as well. The preprocessing time can be compensated by the faster search during the computation of optical flow field due to the shorter lookup table required for computing the error function.

(Manuscript received February 28, 2002, revised July 5, 2002)

References

(1) K. Prazdny: "Motion and Structure from Optical Flow", *Proc. of the 6th IJCAI*, pp. 702-704, Tokyo, Japan, (1979)

(2) G. Adiv: "Determining Three-Dimensional Motion And Structure From Optical Flow Generated By Several Moving Objects", *IEEE Trans. on PAMI*, Vol. 7,pp. 384-401 (1985)

(3) B.K.P. Horn and B.G. Schunck: "Determining Optical Flow", *Artif. Intell.*, Vol. 17, pp. 185-204 (1981)

(4) B.G. Schunck: "Image Flow Segmentation and Estimation by Constraint Line Clustering", *IEEE Trans. on PAMI*, Vol. 11, No. 10, pp. 1010-1027 (1989)

(5) B.D. Lucas and T. Kanade: "An Iterative Image Registration Technique with an Application to Stereo Vision", *IJCAI*, pp. 674-679 (1981)

(6) F. Glazer: "Hierarchical Gradient-Based Motion Detection", *IUW*, Vol. 87, pp. 733-748 (1987)

(7) H.H. Nagel: "On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results", *Artif. Intell.*, Vol. 33, pp. 299-324 (1987)

(8) D.J. Fleet and A.D. Jepson: "Computation of Component Image Velocity from Local Phase Information", *IJCV*, Vol. 5, No. 1, pp.77-104 (1990)

(9) D.J. Heeger: "Optical Flow using Spatiotemporal filters", *IJCV*, Vol. 1, (1988)

(10) S.T. Barnard and W.B. Tompson: "Disparity Analysis of Images", *IEEE Trans. on PAMI*, Vol. 2, No. 4, pp.330-340 (1980)

(11) P. Anandan: "A Computational Framework and an Algorithm for the Measurement of Visual Motion", *IJCV*, Vol. 2, pp. 283-310 (1989)

(12) A. Singh: "An Estimation-Theoretic Framework for Image flow Computation", *Proc. 3rd ICCV, Osaka, Japan*, pp. 168-177 (1990)

(13) S.S. Beauchemin and J.L. Barron: "The Computation of Optical Flow", *ACM Computing Surveys*, Vol. 27, No. 3, pp. 433-467 (1995)

(14) A. Mitiche and P. Bouthemy: "Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods", *IJCV*, Vol. 19, No. 1, pp. 29-55 (1996)

(15) J.L. Barron , D.J. Fleet, and S.S. Beauchemin: "Performance of Optical Flow Techniques", *IJCV*, Vol. 12, No. 1, pp. 43-77 (1994)

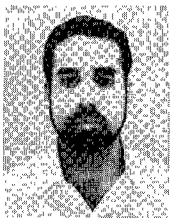
(16) J.L. Barron and A. Liptay: "Optical Flow to Measure Minute Increments in Plant Growth", *Bioimaging*, Vol. 2, No. 1, pp. 57-61 (1994)

(17) F. Ullah, S. Kaneko, and S. Igarahi: "Orientation Code Matching for Robust Object Search", *IEICE Trans. on Inf. & Sys.*, Vol. E84-D. No. 8, pp. 999-1006 (2001)

(18) T. Camus: "Real-Time Quantized Optical Flow", *Proc. IEEE Computer Architectures for Machine Perception*, Como, Italy, (1995)

(19) http://mee.coin.eng.hokudai.ac.jp/~farhan/

Farhan Ullah (Non-member) received B.E. in Computer Systems Engineering from NED University of Engineering and Technology, Karachi and M.Sc. (Systems Engineering) from Quaid-i-Azam University, Islamabad in 1990 and 1992 respectively. He is currently a Ph.D. student in the department of Systems and Information Engineering in Hokkaido University, Sapporo. His research interests include image processing, pattern recognition and robotic vision. He is a student member of IEEE Computer Society and Association of Computing Machinery.



Shun'ichi Kaneko (Member) received the B.S. degree in precision engineering and the M.S. degree in information engineering from Hokkaido University, Japan, in 1978 and 1980, respectively, and then the Ph.D. degree in systems engineering from the University of Tokyo, Japan, in 1990. He was an associate professor of the Department of Electronic Engineering since 1991 to 1996, in Tokyo University of Agriculture and Technology, Japan. He is an associate professor of Hokkaido University from 1996. He received the Best Paper Award in 1990, the Society Award in 1998, respectively, from JSPE. His research interests include machine vision, image sensing and understanding, robust image registration. He is a member of JSPE, IEICE, IPSJ and IEEE.