Determination of Object Directions Using Optical Flow for Crowd Monitoring

Aravinda S. Rao[1], Jayavardhana Gubbi[1], Slaven Marusic[1], Andrew Maher[2], and Marimuthu Palaniswami[1]

[1] ISSNIP, Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC - 3010, Australia
aravinda@student.unimelb.edu.au, {jgl, slaven, palani}@unimelb.edu.au
[2] ARUP, Melbourne, VIC - 3000, Australia
Andrew.Maher@arup.com

Abstract. Determination of object direction in a multi-camera tracking system is critical. The absence of object direction from other cameras pose challenges if the object is along the optical axis. The problem of determining object direction worsens further if the cameras in the existing infrastructure are improperly placed and are uncontrollable. To determine the direction of an object in such situations, three methods based on optical flow (OF) are presented. The first method uses centroids of optical flow vector magnitudes and Kalman filter for tracking and is suitable for less crowded scenarios. The second method uses geometric moments to evaluate the flow vector distribution and to ascertain the direction in case of crowded scenarios by partitioning the scene and then applying moments to individual partitions independently. The third method is appropriate for small-sized objects near vanishing points where global object motion is less. During surveillance, whether multi-object, single-object or crowded scenarios, the aforementioned methods are applicable accordingly. The results show that the object directions can be accurately inferred from three methods for different scenarios.

1 Introduction

Crowd tracking is an important application in computer vision. In a networked camera setting, within a camera sensor network, three scenarios of object identification and tracking are encountered: (1) overlapping field of view, (2) partially overlapping field of view, and (3) non-overlapping field of view. In case of partially overlapping and non-overlapping field of view, object tracking and determining directions are significantly critical for co-operative tracking. Determining the correct direction of motion is important and challenging, specifically, when fixed existing camera infrastructure is used. For instance, if we have an object approaching towards the camera or moving way from the camera along the optical axis, it is essential for us to determine the direction of the object. However, obtaining direction information in the presence of multiple objects and when the size of the objects are small becomes challenging.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

Background subtraction [1] and optical flow (OF) analyses are the two main methods popularly used for extracting motion information from a region of interest [2]. While the background subtraction models use the variations of the pixels, OF uses irradiance constancy and smoothness in determining the pixel displacement [3]. Subtracting background model would help to provide region of movements, but not direction when object is small and along optical axis. The subtracted model would still highlight the same region without any additional directional information. The optical vectors provide magnitude and direction information along the X and Y axes of the image plane, but meager information along the Z (depth of field of view) axis. For instance, when a camera is installed along the corridors of a large venue, we often see people movement along the optical axis and near vanishing point. Moreover, the placement of the camera (with respect to height) is variable due to varying ceiling heights causing mismatched size of the objects as seen in different cameras.

Using optical flow, in [4], the apparent motion of the observer and the actual optical flow vectors were separated and were mapped using a rotating observer. In [5], one-dimensional optical flow vectors were queued for each direction and the queue that had the maximum positive value was considered as the moving direction. Shibata *et.al.* [6] have used the prominent direction indicated by the feasible vectors. In determining the direction of the object, all of these works inherently depend on the dominant vector directions. Others have proposed head-and-face detection [7], walking direction [8] and gait action [9], where the primary aim was to distinguish among different positional body angles.

Most of the CCTV systems will have vertical FoV (VFoV) of up to 45° from the ceiling such as overhead cameras [10] [11] and tilted [12]. The data that is being used in our work has VFoV up to 30°. Because of this vertical FoV, the objects at the far end of the perspective projection appear along the optical axis. In this paper, we focus on resolving the issues that arise when using motion information obtained from OF. We aim to determine direction of objects only from motion information. Three methods have been proposed to address the direction issue primarily using optical flow. The first method is applicable for situations where objects are clearly separated. The method uses magnitudes of the flow vectors, and their corresponding centroids to track the object direction. The second method uses geometric moments of optical flow distributions in a smaller search space and when the scene is cluttered to determine the collective direction of objects. The third method analyzes the directions obtained by flow vectors of the neighboring pixels of identified object region. The third method is suited for crowded and small-sized objects when the objects appear to be moving along optical axis near vanishing points where motion along X and Y axes are limited.

2 Methodology

Horn-Schunck OF method [3] based on brightness constancy is used in this work. The OF vector matrix O consisting of horizontal (x) and vertical (y) velocities

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

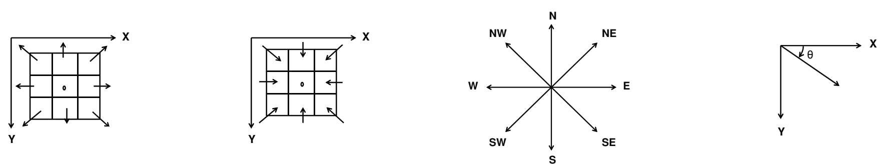
(Eq. 1) is used to calculate the magnitude (mag) and direction (dir) of the vectors as given by Eq. 2 and Eq. 3 respectively. Considering most of the surveillance cameras come with short focal length, and consequently wide angle of view, the imaging of the scene falls under the category of perspective projection. The magnification factor $m = \frac{f'}{-z}$, where z is the distance from the camera to the object point in the scene and f is the focal length of the camera [13]. As the distance between the object and the camera decreases (i.e. z decreases) along the optical axis, the magnification of the object increases and also the area (m^2), associated with it [13]. The OF pattern for a 3×3 object region moving along the optical axis is as shown in Fig. 1.

$$O := \{x + iy : x, y \in \mathbb{R}\} \in \mathbb{C}^{m \times n} \tag{1}$$

where $m, n \in \mathbb{R}$ and $i = \sqrt{-1}$.

$$\text{mag} := \{(x^2 + y^2)^{\frac{1}{2}}\} \in \mathbb{R}^{m \times n} \tag{2}$$

$$\text{dir} := \{\tan^{-1}(\frac{y}{x})\} \in \mathbb{R}^{m \times n} \tag{3}$$



(a) (b) (c) (d)

Fig. 1: Optical flow pattern for an object along the optical axis (a) approaching the camera, (b) moving away from the camera, (c) and (d) depict eight directions juxtaposed against X,Y directions.

2.1 Direction of an object using flow vector magnitudes

Here we consider less-dense case for inferring directions from multiple objects. In case of less dense scenarios, background subtraction approach provides rich information about the scene by subtracting the background of a scene. In order to reduce the noise present in the video and to handle the crowded scenes, preprocessing, segmentation and morphological operations were applied to the raw video (24-bit RGB) by frame differencing combined with RGB channel operations $G^2 - B$ and $(G^2 - B)^{-1}$. All the pixels with OF magnitude greater than zero are labeled as 1 and others as 0. Next, the binarized matrix is relabeled by performing 8-connected-component analysis. Later, the centroids the relabeled matrices are stored for tracking. If an object is approaching towards the

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

camera, then the centroid would move in positive y direction. Furthermore, the Kalman filter was implemented for tracking the objects with centroid of each objects as the current position in the state space model. However, since there is no information about the object directions, a separate routine was maintained to derive directions from the updated equation of the Kalman filter. Based on the previous locations and trajectories for up to $n = 5$ time periods, we deduce object directions.

2.2 Direction of an object using geometric moments

In Section 2.1, the problem to determine directions was a global one, where we considered the entire scene. The drawback of this approach is that because of nonuniform illumination, shadows and noise, global approaches sacrifice certain information to maximize the efficiency. In order to overcome loss of information, we analyzed geometric moments to infer directions from the scene. The analysis was conducted for a single-object moving in cardinal and inter-cardinal directions are considered. Moments and functions of moments indicate invariant pattern features [14] and are separately calculated based on horizontal and vertical velocities obtained from OF. Table 2a (refer to the last page of the paper) summarizes the interpretations based on the real (horizontal) and imaginary (vertical) components of the flow vectors. Fig. 1 shows the convention of the x and y axes along with eight directions. The moments for real values are given by equations (4)– (7). Likewise, we also compute the geometrical moments ($\overline{o}_I, \sigma_I, S_I$ and K_I) for imaginary values of the OF matrix. Our hypothesis is that the same results can be applied to multiple objects to obtain collective object directions by partitioning the scene into different windows based on centroids and apply the same analysis within each window corresponding to an object of interest. As a preliminary result, in this work we have presented the OF distributions for a single object case. From Table 2b it is evident that Kurtosis can be used to infer whether the object is approaching or moving away from the camera.

$$\text{Mean}_R = \overline{o}_R = \frac{1}{N} \sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [O_R(k,l)] \tag{4}$$

$$\text{Variance}_R = \sigma_R = \frac{1}{N} \times \sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k,l) - \overline{o}_R]^2 \tag{5}$$

$$\text{Skewness}_R = S_R = \frac{1}{N} \times \frac{\sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k,l) - \overline{o}_R]^3}{\sigma_R^3} \tag{6}$$

$$\text{Kurtosis}_R = K_R = \frac{1}{N} \times \frac{\sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k,l) - \overline{o}_R]^4}{\sigma_R^4} \tag{7}$$

where $N = m \times n$.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

2.3 Direction of an object based on flow directions

When the objects are small and near vanishing points, their size will not convey much information about direction because the size almost remains same. Because of this there will be ambiguity whether the object is moving away or approaching. As mentioned before, when the depth increases, the magnification factor decreases. Therefore, in contrast to the above two methods, the objective of this method is to extract information from the direction matrix (Eq. 3) for an object's region and make a decision as to whether the object is approaching or moving away from the camera especially near the vanishing points. We In order to determine the direction of the object, a template mask, T, is moved over the direction matrix with the center of the 3×3 matrix being the pixel under consideration. This pixel is assumed to be at the center of the circle as shown in Fig. 2. A score is assigned to determine as to how much the neighboring pixel's direction vector is indicating that it is pointing towards the center pixel is calculated. The scores are calculated from $n = 0$ to $n = 7$. This method is equivalent to finding convergence (sink) or divergence (source) of flow field in a given vector field.

$$T := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} t_5 & t_6 & t_7 \\ t_4 & 0 & t_0 \\ t_3 & t_2 & t_1 \end{bmatrix} \tag{8}$$

A function $f : (\mathrm{dir}(k,l), n) \mapsto c(k,l)$, maps OF vectors' direction matrix to a real number $[-1, 1]$ considering the neighborhood of $c(i,j)$. For a pixel at the center of T, the eight neighboring pixels are considered. In each direction as shown in Fig. 2, the $c(i,j)$ is computed as given by: $\mathrm{c}(k,l) = \frac{1}{8}\sum_{n=0}^{n=7} I_n$, where where,

$$I_n = \begin{cases} +1, \; |r| \leq (\pm\frac{\pi}{8}) \\ 0 \quad, \pm\frac{\pi}{8} < |r| \leq \pm\frac{7\pi}{8} \\ -1, \; |r| > (\pm\frac{7\pi}{8}) \end{cases} \tag{9}$$

and

$$r = (n \times \frac{\pi}{4}) - t_n \tag{10}$$

where n indicates the position of the current pixel being analyzed from center and t_n is equal to the $dir(k,l)$ along that direction in the neighborhood. The value $\frac{\pi}{4}$ is chosen as a threshold such that r of the center pixel determines whether the neighboring pixel is within $\pm\frac{\pi}{8}$ radians from the center and I_n assigns scores based of deviation from the center pixel's value. The score is incremented or decremented (by 1) based on whether the pattern agrees with the indented direction (within $\pm\frac{\pi}{8}$) or in the opposite direction, and left unchanged for any other directions. This is then summed for all the eight directions and normalized. For this score to apply, the magnitude of center pixel must be zero. Either Fig. 1-(a) or Fig. 1-(b) must be considered and scores must be applied. For instance, considering Fig. 1-(a), if an element $c(k,l)$ yields a score of 1, it implies that there is an object moving away from the camera and a score of -1 for $c(k,l)$ indicates that the object is approaching the camera.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

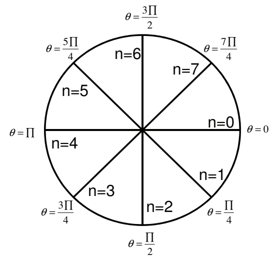
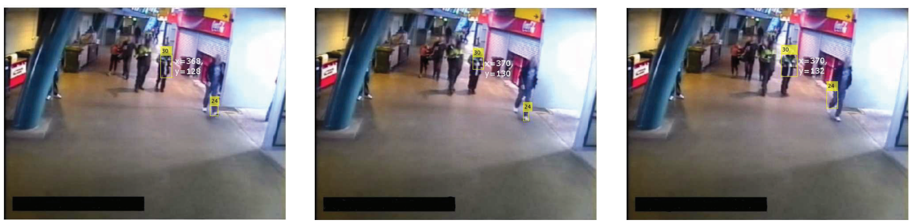


Fig. 2: Depicts the θ value in radians for n

3 Results and discussion



(a) (b) (c)

Fig. 3: Tracking centroid of object using Kalman Filter for object id 30 - the value of index y is increasing by 2 as object is approaching the camera - video collected from a major sporting venue.

All implementations were carried out in MATLAB 8.0 using Computer Vision System Toolbox on Windows XP-SP2, Intel i7 $-$ 2600, running at 3.4 GHz on a 32-bit computer utilizing 512 MB ATI Radeon$^{\text{TM}}$HD 5450 graphics card.

The result of determining the direction using the centroids is given in Fig. 3, where Fig. 3-(a) shows the centroid at location $(368, 128)$, Fig. 3-(b) shows the centroid at location $(370, 130)$ and Fig. 3-(c) at location $(370, 132)$ for a video collected from a a major sporting venue corridor camera. Kalman filter was used to keep track of the object and its location and the result for three different frames are shown in Fig. 3. By keeping track of centroid locations, we estimate the trajectory along optical axis.

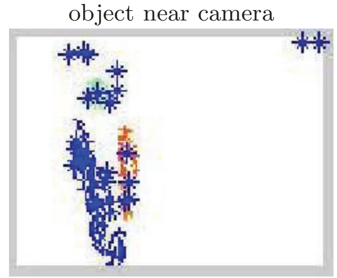
For the second method, based on the rules in Table 2b, the features in Table 1 were extracted using the OF magnitude distributions of the scene. It is evident that kurtosis can be used to determine whether the object is approaching or moving away. Additionally, mean values of horizontal and vertical velocities provide movements in X and Y directions. Skewness can be used to measure the object's movements along diagonal directions. Fig. 4 shows two cases for a video that was filmed in our lab specifically for calculating the moments.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

(a) Flow vectors for object near camera - frame #10

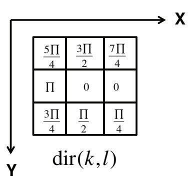
(b) Flow vectors' distribution for object near camera

(c) Flow vectors for object far from camera - frame #160

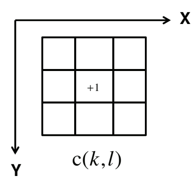


(d) Flow vectors' distribution for object far from camera

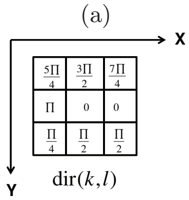
Fig. 4: Geometric moments are calculated based on OF distributions in (b) and (d).



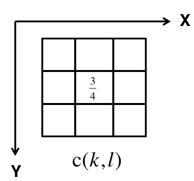
(a)



(b)



(c)



(d)

Fig. 5: (a) shows the ideal $\mathrm{dir}(k, l)$ for an object approaching the camera and (b) the corresponding $c(i, j) = +1$. (c) shows the nonideal condition for an object approaching the camera and (d) the corresponding $c(k, l) = \frac{3}{4} < 1$.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

The result for the third result was simulated and shows that if the flow vectors' direction for an object approaching the camera agrees with the Fig. 1-(a), then $c(k,l) = 1$. On the other hand, if the flow vectors' directions are not agreeing with either Fig. 1-(a) or Fig. 1(b), the method would detect this by providing score between 1 and -1 to the region being analyzed. Pragmatically, encountering the intended pattern exactly would be minimal. Hence, once can relax the tolerance in each directions while calculating the scores. For the same reason, we have shown only the simulated results. Furthermore, one can extend the 3×3 matrix to any $p \times p$ such that $p \leq \min(m, n)$.

In a surveillance system we often come across single and multiple objects. In case of multi-object scenarios, when the objects are larger in size (because of less depth along optical axis) the first method provides object direction along optical axis. Kalman filter is required to keep track of individual object's centroids and velocities to estimate the direction. Further during surveillance, we can separate multiple objects into single objects and then apply the geometric moments on flow vectors (this requires less processing cycles since the OF results are already available) to find out the directions. It is to be noted that when multiple objects are present, the OF directions do not convey meaningful information. In order to make sense of the OF vectors, the distribution of geometric moments are used. When the size of the objects becomes smaller, the first two methods would not provide accurate results. Therefore, we concentrate on the optical flow vectors of identified small region of the frame and apply the third method for direction information (diverging or converging).

Moments	**Vector coefficients**	**Feature**
Mean	Real, Imaginary	X,Y directions
Skewness	Real	Diagonal movements
Kurtosis	Real, Imaginary	Object closeness

Table 1: Features extracted from moments.

4 Conclusion

Determination of object direction using optical flow along the optical axis was presented. Three methods were proposed to calculate the direction based on OF centroids, geometric moments, and direction flow pattern at any given instance. In terms of suitability, the first method is applicable for situations where objects are clearly separated. The method uses optical flow and background subtraction to obtain centroids. Further, the Kalman filter is used for tracking and a subroutine to deduce directions based on previous location trajectories. The second method is devised for obtaining directions when the scene is cluttered. In this case the scene is divided into partitions and geometric moments are calculated to infer collective group directions. The third method is suited for crowded and small-sized objects when the objects appear not to be moving along optical axis. The results show that object direction along the optical axis can be deduced from the three methods for different scenarios.

References

1. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE. Volume 2. (1999) 246–252
2. Lepisk, A.: The use of optic flow within background subtraction. Master's thesis, Stockholm, Sweden: Numerisk analys och datalogi (NADA) (2005)
3. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
4. Kinoshita, K., Enokidani, M., Izumida, M., Murakami, K.: Tracking of a moving object using one-dimensional optical flow with a rotating observer. In: 9th International Conference on Control, Automation, Robotics and Vision. IEEE. (2006) 1–6
5. Kinoshita, K., Murakami, K.: Moving object tracking via one-dimensional optical flow using queue. In: 10th International Conference on Control, Automation, Robotics and Vision. IEEE. (2008) 2326–2331
6. Shibata, M., Makino, T., Ito, M.: Target distance measurement based on camera moving direction estimated with optical flow. In: 10th IEEE International Workshop on Advanced Motion Control. IEEE. (2008) 62–67
7. Ishii, Y., Hongo, H., Yamamoto, K., Niwa, Y.: Real-time face and head detection using four directional features. In: Proceedingof the Sixth IEEE International Conference onAutomatic Face and Gesture Recognition, IEEE. (2004) 403–408
8. Lertniphonphan, K., Aramvith, S., Chalidabhongse, T.H.: Human action recognition using direction histograms of optical flow. In: 2011 11th International Symposium on Communications and Information Technologies (ISCIT). IEEE. (2011) 574–579
9. Lijia, W., Songmin, J., Xiuzhi, L., Shuang, W.: Human gait recognition based on gait flow image considering walking direction. In: 2012 International Conference on Mechatronics and Automation (ICMA). IEEE. (2012) 1990–1995
10. Barandiaran, J., Murguia, B., Boto, F.: Real-time people counting using multiple lines. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services. IEEE. (2008) 159–162
11. Rizzon, L., Massari, N., Gottardi, M., Gasparini, L.: A low-power people counting system based on a. In: IEEE International Symposium on Circuits and Systems. IEEE. (2009) 786–786
12. Garcia, J., Gardel, A., Bravo, I., Lazaro, J., Martinez, M., Rodriguez, D.: Directional people counter based on heads tracking. IEEE Transactions on Industrial Electronics. IEEE **PP** (2012) 1–10
13. Horn, B.K.P.: Robot vision. MIT Press, Cambridge, MA, USA (1986)
14. Teh, C.H., Chin, R.T.: On image analysis by the methods of moments. IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988) 496–513

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-41939-3_60

Moments	Vector coefficients	Absence of object	Forward (to N)	Backward (to S)	Diagonal (SW to NE)	Diagonal (SE to NW)	Object Near Camera	Object away from Camera
Mean	Real	0	constant	constant	positive	negative	N/A	N/A
	Imaginary	0	negative	positive	negative	negative		
Variance	Real	0	nominal	nominal	nominal	N/A	N/A	N/A
	Imaginary	0	nominal	nominal	nominal	N/A		
Skewness	Real	0	unchanged	unchanged	positive	negative	N/A	N/A
	Imaginary	0	unchanged	unchanged	N/A	N/A		
Kurtosis	Real	0	Low to High	High to Low	High to Low	High to Low	Low	High
	Imaginary	0						

Table 2a: Interpretation of geometric moments for an object moving in eight directions.

Moments	Vector coefficients	Object near camera	Minimum	Maximum	Object moving away from camera	Minimum	Maximum
Mean	Real	−0.331786	−0.3625	0	−0.003019	−0.3625	0.4703
	Imaginary	−0.059070	−0.0624	0	0.014716	−0.2593	0.0367
Variance	Real	18.215907	0	18.2159	1.048159	0	25.8190
	Imaginary	15.030494	0	15.0305	1.014135	0	21.7704
Skewness	Real	−4.220454	−10.8099	0	1.779557	−11.8754	17.9986
	Imaginary	−0.546849	−9.7723	0.3561	10.219373	−18.2828	10.2194
Kurtosis	Real	65.825336	0	332.0975	425.208734	0	1051.20
	Imaginary	65.519053	0	328.4689	551.674870	0	1209.00

Table 2b: Geometric moments calculated for an object near camera and moving away from camera as shown in Fig. 4.