

# Motion Detection Techniques Using Optical Flow

A. A. Shafie, Fadhlan Hafiz, and M. H. Ali

**Abstract**—Motion detection is very important in image processing. One way of detecting motion is using optical flow. Optical flow cannot be computed locally, since only one independent measurement is available from the image sequence at a point, while the flow velocity has two components. A second constraint is needed. The method used for finding the optical flow in this project is assuming that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. This technique is later used in developing software for motion detection which has the capability to carry out four types of motion detection. The motion detection software presented in this project also can highlight motion region, count motion level as well as counting object numbers. Many objects such as vehicles and human from video streams can be recognized by applying optical flow technique.

**Keywords**—Background modeling, Motion detection, Optical flow, Velocity smoothness constant, motion trajectories.

## I. INTRODUCTION

OPTICAL flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer [1, 2]. Consequently, optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement [3]. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects [4].

## II. OPTICAL FLOW IN MOTION ANALYSIS

Optical flow gives a description of motion and can be a valuable contribution to image interpretation even if no quantitative parameters are obtained from motion analysis. Optical flow can be used to study a large variety of motions—moving observer and static objects, static observer and moving objects, or both moving. Optical flow analysis does not result in motion trajectories instead, more general motion properties are detected that can significantly increase the reliability of complex dynamic image analysis [5]. Motion, as it appears in dynamic images, is usually some combination of four basic elements:

- i) Translation at constant distance from the observer.
- ii) Translation in depth relative to the observer.
- iii) Rotation at constant distance about the view axis.
- vi) Rotation of a planar object perpendicular to the view axis.

Optical-flow based motion analysis can recognize these basic elements by applying a few relatively simple operators to the flow [6]. Motion form recognition is based on the following facts:

- a) Translation at constant distance is represented as a set of parallel motion vectors.
- b) Translation in depth forms a set of vectors having a common focus of expansion.
- c) Rotation at constant distance results in a set of concentric motion vectors.
- d) Rotation perpendicular to the view axis forms one or more sets of vectors starting from straight line segments. Exact determination of rotation axes and translation trajectories can be computed, but with a significant increase in difficulty of analysis.

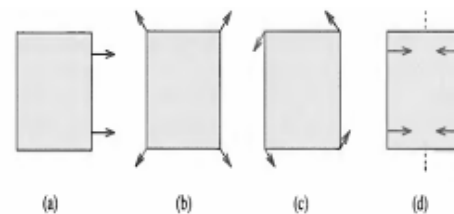


Fig. 1 Motion form recognition, (a) Translation at constant distance, (b) Translation in depth, (c) Rotation at constant distance, (d) Planar object rotation perpendicular to the view

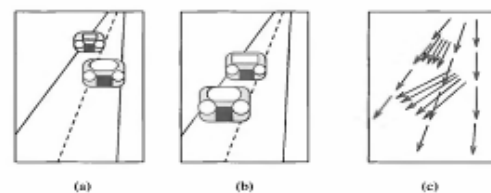


Fig. 2 Focus of expansion, (a) Time t1 (b) Time t2. (c) Optical flow

## III. OPTICAL FLOW COMPUTATION

Optical flow computation is based on two assumptions:

1. The observed brightness of any object point is constant over time.
2. Nearby points in the image plane move in a similar manner (the velocity smoothness constraint). Suppose we have a continuous image;  $f\{x,y,t\}$  refers to the gray-level of  $(x,y)$  at time  $t$ .

Representing a dynamic image as a function of position and time permits it to be expressed.

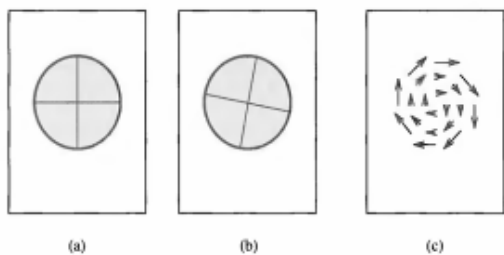


Fig. 3 Optical flow, (a) Time t1. (b) Time t2 (c) Optical flow

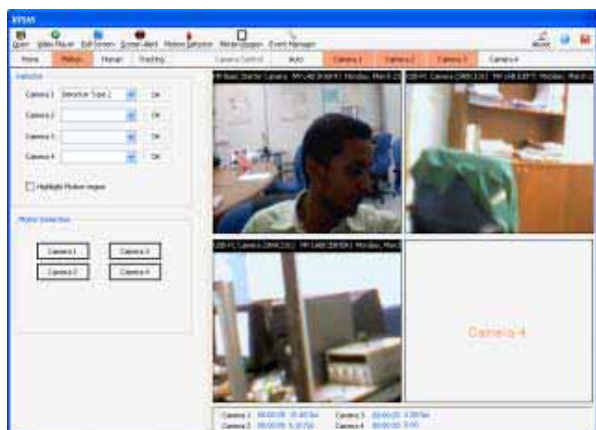
As refer to (1):

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + f_x dx + f_y dy + f_t dt + O(d2) \tag{1}$$

where  $f_x$ ,  $f_y$ ,  $f_t$  denote the partial derivatives of  $f$ . We can assume that the immediate neighborhood of  $(x, y)$  is translated some small distance  $(dx, dy)$  during the interval  $dt$ ; that is, we can find  $dx$ ,  $dy$ ,  $dt$  as refer to (2) :

$$f(x + dx, y + dy, t + dt) = f(x, y, t) \tag{2}$$

#### IV. METHODS AND MATERIALS



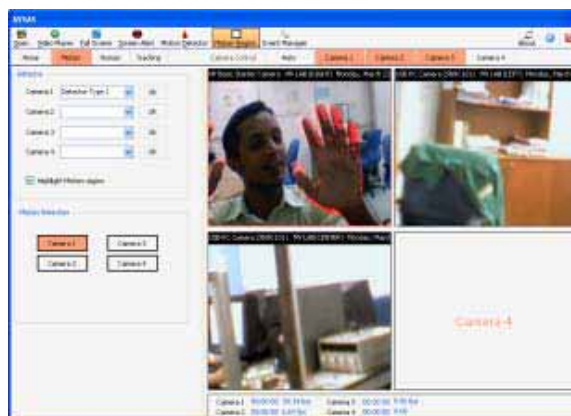
##### A. Functionalities

This software has capability to detect motion detection based on optical flow and gives certain level of motion detection which can be used as a threshold. Analyzing motion level and comparing it with predefined threshold allows raising alarm, when detected motion level is greater then the level which is considered to be safe. In addition to motion level detection, there are four types of motion detectors and all of them support highlighting of detected motion regions (can be turned on/off).

##### B. Types of Detection used in the Software

###### 1) Two frames difference motion detector

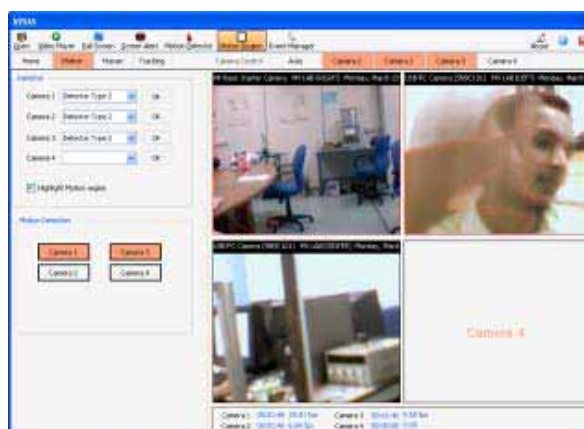
This type of motion detector is the simplest one and the quickest one. The idea of this detector is based on finding amount of difference in two consequent frames of video stream. The greater is difference, the greater is motion level. As it can be seen from the picture below, it does not suite very well those tasks, where it is required to precisely highlight moving object. However it has recommended itself very well for those tasks, which just require motion detection.



###### 2) Motion detectors based on background modeling

In contrast to the above motion detector, these motion detectors are based on finding difference between current video frame and a frame representing background. These motion detectors try to use simple techniques of modeling scene's background and updating it through time to get into account scene's changes. The background modeling feature of these motion detectors gives the ability of more precise highlighting of motion regions. Below are demonstrated outputs of two versions of motion detectors based on background modeling. One does more precise highlight of moving objects' borders, but consumes more computational resources. Another one does less precise objects' highlight in the cost of requiring much less computational resources.

###### i) Low-Precision Background Modeling



## ii) High-Precision Background Modeling



## 3) Counting Motion Detectors

The counting motion detector is based on the same idea of background modeling as the above motion detectors. However after that it does additional processing and different object's highlighting. Once motion regions are identified, this detector uses blob counting algorithm to find rectangles of each detected moving object. This gives the ability to report about amount of detected objects, as well as position and size of each detected object. **The sizes can later being used in human recognition.**



## V. RESULT

According to the figures given above we found that “two frames difference” method is suitable for simple motion detection only it can not highlight specific region of moving objects. From the two types of background modeling we observed that the high precision method is good for detecting whole moving object precisely but it requires complex computational method. On the other hand Low precision method requires less calculation but it gives less precise result. We also found that the Counting motion method is good to count the object and get their specific size.

## VI. DISCUSSION

Optical flow reflects the image changes due to motion during a time interval  $dt$  which must be short enough to guarantee small inter-frame motion changes. The optical flow field is the velocity field that represents the three-dimensional motion of object points across a two-dimensional image.

Optical flow computation is based on two assumptions:

- 1) The observed brightness of any object point is constant over time.
- 2) Nearby points in the image plane move in a similar manner (the velocity smoothness constraint).

## VII. CONCLUSION

Optical flow computation will be in error if the constant brightness and velocity smoothness assumptions are violated. In real imagery, their violation is quite common. Typically, the optical flow changes dramatically in highly textured regions, around moving boundaries, at depth discontinuities, etc. Resulting errors propagate across the entire optical flow solution.

## REFERENCES

- [1] Gibson. I.I.. The Perception of the Visual World (Riverside Press, Cambridge. 1950)
- [2] Gibson. I.I.. The Senses Considered as Perceptual Systems (Houghton-Mi Win, Boston. MA, 1966).
- [3] Gibson, J.J., On the analysis of change in the optic array. Scandinavian I. Psychol. 18 (1977)161-163.
- [4] Nakayama. K. and Loomis. J.M.. Optical velocity patterns. Velocity-sensitive neurons and space perception. Perception 3 (1974) 63-80.
- [5] Thompson et al., 1985; Kearney et al., 1987; Aggarwal and Martin, 1988.
- [6] Thompson et al., 1984; Mutch and Thompson, 1984.
- [7] B.H. Sonka, Image Processing, Analysis, and Machine Vision [1999], pp. 757-759, 808.