SIMULATION STUDIES ON HORN AND SCHUNCK OPTICAL FLOW ALGORITHM AND FOCUS OF EXPANSION FOR AUTONOMOUS NAVIGATION OF UNMANNED VEHICLES

S. Santhosh Kumar[1], Preetham Shankpal[2], K. R. Prashanth[3], Saima Mohan[4], Narasimha Reddy[5], Govind R. Kadambi[6], S. R. Shankapal[7]

M. S. Ramaiah School of Advanced Studies, Bangalore- 58.

Abstract

Vision based obstacle detection and collision avoidance is a major challenge in unmanned vehicle navigation. The gradient based Horn and Schunck algorithm has been optimized for computation of Optical Flow (OF) vectors for both virtual and real images. Based on extensive and systematic simulation studies, the desirable values of important parameters such as Compensation parameter, number of iterations and resolution of images that determine the accuracy and efficacy of Horn and Schunck Algorithms have been suggested.

A novel and computationally efficient Cluster based method is proposed for the determination of FOE. The proposed Cluster based method overcomes the earlier drawback of rapid variations in the co ordinates of FOE when successive frames are considered for simulation. Separate decision logics appropriate for UGV and MAV navigation have been identified.

The simulation studies reported in this paper generally treat both the synthetic as well as real images.

Keywords: Micro Air Vehicle [MAV], Unmanned Ground Vehicle [UGV], Optical Flow [OF], Focus Of Expansion (FOE), Time To Contact (TTC)

1. INTRODUCTION

Computer vision [1] is a discipline that builds on the theory for developing artificial systems to obtain information from images. The image data can be a video sequence, views from multiple cameras, or multi-dimensional data from a medical scanner. Computer vision studies and describes artificial vision systems that are implemented in software and/or hardware. There are many motivations to develop autonomous artificial systems like robots, MAVs and UGVs. Artificial systems can replace humans in various situations like microscopic visual inspection, video recording in tense situations like wars and similar situations for scene reconstruction, event detection, tracking, object recognition, learning, indexing, motion estimation, and image restoration.

Computer vision [1] is a field of robotics in which programs attempt to identify objects represented in digitized images provided by video camera, thus enabling robots the power of vision. Research is being carried out on stereo vision as an aid for object identification and location within a three-dimensional field of view. Recognition of objects in real time, as would be needed for active robots in complex environments, usually requires the state-of-art technology for computing. A typical small computer vision system consists of a camera, a frame grabber suite that plugs into a personal computer to capture images, and a suite of software that allows the user to experiment using various image processing/video processing operations and develop application systems.

An artificial robotic system that has true mobility with the help of computer vision must be able to sense its surroundings and other objects in its surroundings. To sense the surrounding the system must have visual perception. The visual input should obviously be through a camera. Once the system has a visual input, the system must have capability to differentiate and analyze to facilitate recognition or motion of obstacles. This is accomplished by the computer vision technology. One of the techniques that can be used in computer vision is Optical Flow [OF]. OF [2] is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer and the scene. Optical flow algorithms provide mapping of 3D velocities on 2D image space. OF can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. Discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects.

At times of a terror attack or for homeland security surveillance, reconnaissance, bomb damage assessment, or search-and-rescue within an unfamiliar territory is dangerous and also requires a large, diverse task force. Unmanned robotic vehicles could assist in such missions by providing situational awareness without risking the lives of soldiers, first responders, or other personnel. Currently Sonar and Laser methods are used for obstacle detection but researchers are now looking towards an important sensory system that many biological creatures use everyday − vision. Vision is a very powerful sensor providing numerous types of information useful in obstacle detection. One of the techniques that can be used for computer vision is OF [3]. The focus of this paper includes both analytical and simulation studies on Horn and Schunck method which is one of the OF algorithms.

2. Optical Flow

OF is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer and the scene. OF algorithms provide mapping of 3D velocities on 2D image space. OF gives important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement.

Discontinuities in the OF can help in segmenting images into regions that correspond to different objects. The success of an optical flow algorithm is gauged on three basic requirements of robotic vision such as the robustness, computational speed and the accuracy. There are various optical flow algorithms but they do not satisfy all the above mentioned requirements. Even if there is a real time implementation of one such algorithm, it requires a high end workstation. As a result robotic vision researchers find it difficult to obtain reliable OF estimates in practical scenario.

Gradient based OF vector computation on image sequences has been used for motion estimation, Focus of Expansion [FOE] and Time To Contact [TTC] calculation and also provides decision logic for obstacle avoidance. Different pyramid levels of the images have been used to reduce the computation time. Higher the pyramid, lower the resolution of the image. Figure 1 explains the overview of visual navigation system based on OF vectors.

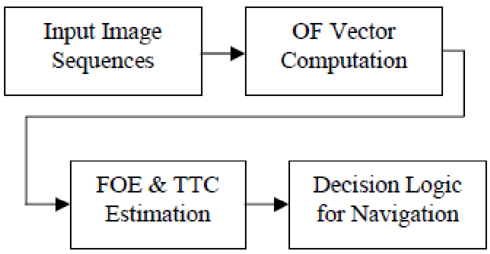


Fig. 1 Optical Flow based Visual Navigation System

3. HORN AND SCHUNCK ALGORITHM

The technique proposed by Horn and Schunck [4] come under the differential or gradient based method. Gradient-based methods use spatial and temporal partial derivatives to estimate OF at every position in the image. Probably the most well known work on OF is by Horn and Schunk [4] that is one of the foremost of the gradient-based algorithms.

Differential techniques compute image velocity from spatio-temporal derivatives of image intensities. The image domain is therefore assumed to be continuous (or differentiable) in space and time. Global and local methods based on the equation below, can be used to compute OF.

$$E_X u + E_Y v + E_t = 0 \text{ ----------- } (1)$$

Where E_X, E_Y and E_t are the partial derivatives of image brightness with respect to **x, y** and **t** respectively.
u and v are derivatives of x and y with respect to time

The brightness at a point **(x, y)** in the image plane at a time **t** is denoted by **E(x, y, t)**. When the pattern moves, the brightness of the particular point is constant, so that

$$\frac{dE}{dt} = 0 \text{ ------------- } (2)$$

Considering a patch of brightness pattern that is displaced a distance δ**x**, in the **x** direction and δ**y** in the **y** direction in time δ**t** the brightness of the patch is assumed to remain constant so that [4]

$$\mathbf{E(x, y, t) = E(x+\delta x, y+\delta y, t+\delta t)} \text{ ------------}(3)$$

Expanding the right hand side of the equation (3) about the point **(x, y, t)** we get,

$$E(x,y,t) = E(x,y,t) + \delta x\frac{\partial E}{\partial x} + \delta y\frac{\partial E}{\partial y} + \delta t\frac{\partial E}{\partial t} + e \text{ ..}(4)$$

In equation (4) **e** contains the second and the higher order terms in δ**x**, δ**y** and δ**t**. After subtracting **E(x, y, t)** from both sides and dividing through by δ**t** we get the following equation,

$$\frac{\delta x}{\delta t}\frac{\partial E}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} + \vartheta(\delta t) = 0 \text{ .... } (5)$$

Where $\vartheta(\delta t)$ is a term of the order δ**t**, and we assume that δ**x** and δ**y** vary as δ**t**. In the limit as δ**t** → **0** the equation (5) becomes,

$$\frac{\delta x}{\delta t}\frac{\partial E}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0 \text{ ----------------- } (6)$$

Denoting **u** =*dx/dt and* **v** =*dy/dt* , we have a linear equation having two unknowns **u** and **v** [4].

$$E_X u + E_Y v + E_t = 0 \text{ -------------- } (7)$$

Where E_X, E_Y and E_t are the partial derivatives of image brightness with respect to **x, y** and **t** respectively. The constraint on the local velocity expressed in the following figure 2. The equation can be expressed in another form

$$(E_x, E_y).(u,v) = -E_t \text{ --------------- } (8)$$

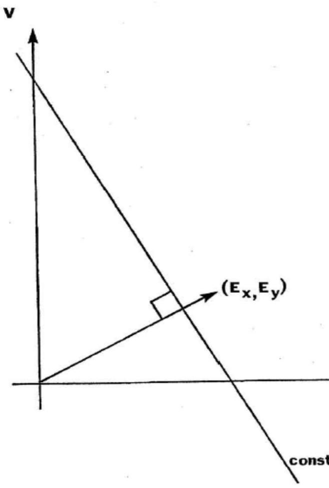


Fig. 2 Velocity (u,v) brightness gradient vector [4]

The basic rate of change of the image brightness equation constraints the optical flow velocity. The velocity **(u, v)** has to lie along a certain line perpendicular to the brightness gradient vector (E_x, E_y) in the velocity space [4]. It can be determined that the component of the motion in the direction of the gradient (E_x, E_y) is

$$\frac{E_t}{\sqrt{E_x^2 + E_y^2}} \qquad \text{--------------- (9)}$$

As a consequence, the flow velocity is computed using the smoothness constraint or the spatial coherence constraint. A way to express this additional limitation is to minimize the square of the gradient of the speed of the OF. Another measure of the smoothness of the field of OF is the sum of the squares of the Laplacians of the velocity components. The Laplacians of **u** and **v** are defined as:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \qquad \nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \qquad \text{----- (10)}$$

In normal situations both Laplacians are zero (when the observer moves parallel to a flat object, rotations on a perpendicular line to the surface, motions orthogonal to the surface); in all these cases the second derivatives of **u** and **v** disappear.

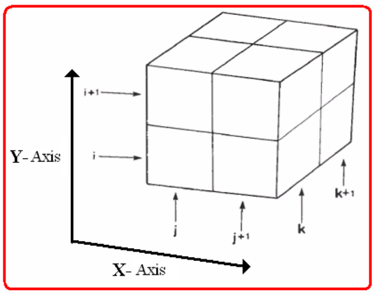


Fig. 3 Three partial derivatives of images brightness at the center of the cube [4]

To estimate u and v from the intensity in the available images, it is important that the estimations of E_x, E_y and E_t be consistent. The method used determines E_x, E_y and E_t at the centre of a cube (Figure 3) by eight measurements. Horn and Schunck estimated the partial derivatives using the following set of equations,

$$E_x = \frac{1}{4}\{E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1}\} \qquad \text{-- (11)}$$

$$E_y = \frac{1}{4}\{E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1}\} \qquad \text{--- (12)}$$

$$E_t = \frac{1}{4}\{E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k}\} \qquad \text{--- (13)}$$

Approach for the calculation of the Laplacians is given below in the following equations

$$\nabla^2 u = L(\bar{u}_{i,j,k} - u_{i,j,k}) \qquad \nabla^2 v = L(\bar{v}_{i,j,k} - v_{i,j,k}) \qquad \text{----- (14)}$$

The value of the constant **L** in the above equation (14) is assumed. The local averages $\bar{u}$ and $\bar{v}$ were given by

$$\bar{u}_{i,j,k} = \frac{1}{6}\{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\} + \frac{1}{12}\{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\} \qquad \text{------- (15)}$$

$$\bar{v}_{i,j,k} = \frac{1}{6}\{v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\} + \frac{1}{12}\{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\} \qquad \text{--------- (16)}$$

To minimize the error in the estimated OF, the following equations (17) and (18) are used

$$\Phi_b = E_x u + E_y v + E_t \qquad \text{------------------ (17)}$$

$$\Phi_c^2 = (\bar{u} - u)^2 + (\bar{v} - v)^2 \qquad \text{------------------ (18)}$$

The measurements of the image intensity can be altered by the quantification error and the noise, because of that, it cannot be ensured that Φ_b is zero. This quantity will spread to have an error of magnitude proportional to the noise of the measurements. The total noise is determined by the following equation:

$$\Phi^2 = \alpha^2 \Phi_c^2 + \Phi_b^2 \quad \text{---------------} (19)$$

Equation (19) introduces an appropriate compensation factor, called α. This value to be calculated for every iteration and is given by the following equation [4].

$$\alpha = \frac{E_x \bar{u}_{ij}^n + E_y \bar{v}_{ij}^n + E_t}{1 + L(E_x^2 + E_y^2)} \quad \text{-----------} (20)$$

Equations 21 and 22 are used to update the values of ***u*** and ***v*** for every iteration.

$$u_{ij}^{n+1} = \bar{u}_{ij}^n - \alpha E_x \quad \text{----------------} (21)$$

$$v_{ij}^{n+1} = \bar{v}_{ij}^n - \alpha E_y \quad \text{----------------} (22)$$

In equations 21 and 22, n denotes the iteration number. The new value of ***(u, v)*** at a point is equal to the average of surrounding values minus an adjustment in the direction of the brightness gradient. This will modify the restriction line or the constraint line illustrated in Figure 2. The value of the flow velocity **(u, v)** which minimized the error **(Φ)** lies on a line drawn from the local average of the flow velocity $(\bar{u}, \bar{v})$ perpendicular to the constraint line

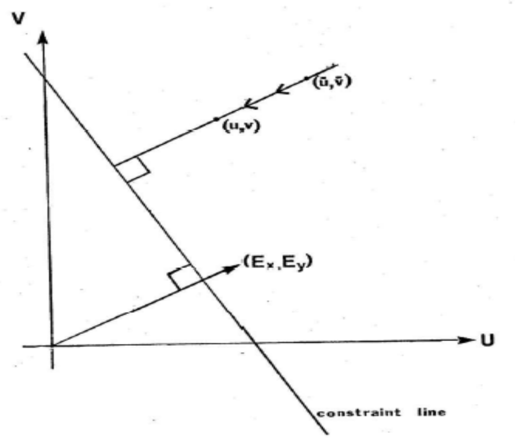


Fig. 4 Constraint line after the smoothing assumption

Horn and Schunck assumed that the when opaque objects of finite size are undergoing rigid motion, the neighboring points on the objects have similar velocities and the velocity fields of the brightness patterns in the image varies smoothly almost everywhere. This will surely be violated in the regions of motion discontinuities.

One of the most important criteria in the gradient-based methods is the number of iterations. According to Horn and Shunck, the number of iterations should be larger than the cross-section of the biggest region that must be filled in. If the

Sizes of such regions are not known in advance one may use the cross-section of the whole image as a conservative estimate. Practically one has a choice of how the iterations are to be interlaced with the time steps. On the one hand, one could iterate until the solution has stabilized before advancing to the next image frame. On the contrary, given a good initial guess one may need only one iteration per time-step. A good initial guess for the OF velocities is usually available from the previous time-step. The advantages of the latter approach include an ability to deal with more images per unit time and better estimates of OF velocities in certain regions. Areas in which the brightness gradient is small lead to uncertain, noisy estimates obtained partly by filling in from the surroundings. These estimates are improved by considering further images. The noise in measurements of' the images will be independent and tend to cancel out. Perhaps more importantly, different parts of the pattern will drift by a given point in the image [4].

Results of Horn and Schunck Algorithm:

Figure 4 shows images taken from a fixed camera. From theory we know that *for images from a still camera the optical vectors should be towards the object*. Figure 6 shows motion of OF vectors using Horn and Schunck for the case of three vehicles in different directions.

Fig. 5 Taxi sequence: Two consecutive frames showing movement of three different cars

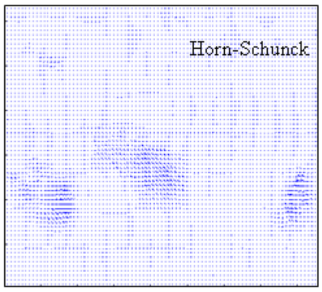


Fig. 6 OF vectors of Taxi Sequence using Horn and Schunck Algorithm

In this section results of the Horn and Schunck algorithm are discussed in detail. The influence of compensation/weight factor **(α)**, number of iterations **(n)** and resolution of the Images (Pyramidal levels) on the OF vectors are analyzed.

Variation of Compensation/ Weight Factor (α)

Cameras of various resolutions are available for the acquisition of image data. In Horn and Schunck algorithm, the weight factor (α) will vary for images obtained from different cameras of different resolution. The only drawback of Horn and Schunck algorithm is that the weight factor has to be optimized for a particular camera. From different trials it is concluded that α of 100 is good for virtual images and 30 for real images of a camera of resolution 288x352.

Figure 7 shows two consecutive images obtained from a moving camera. From the theory it is known that, the pattern of OF vectors obtained from the image flow of a moving camera approaching a stationary obstacle will exhibit a pattern of divergence. One such example of divergence of OF vectors on the image flow taken from a virtual video is as shown in Figure 8.

Fig. 7 Two consecutive frames from moving camera

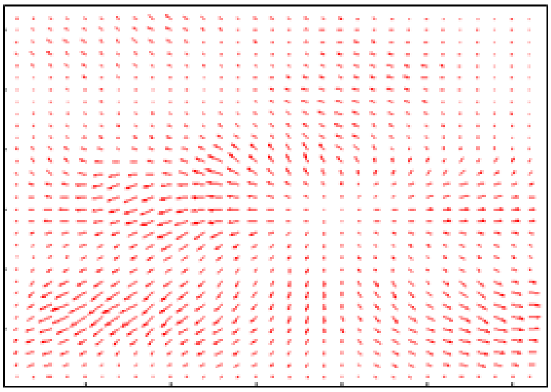
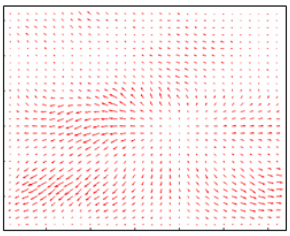
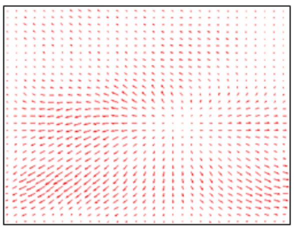


Fig. 8 OF vectors for images shown in Figure 6

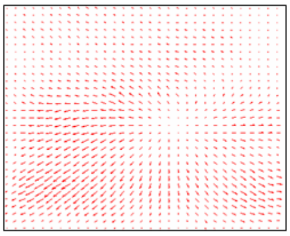
Figure 9 shows the variation in results of OF vectors for a constant iteration value (n) of 100 and weight factors (α) varying from 100 to 400. The key observations made during various trials are - as the value of α is increased, the plot of vectors get denser and as the number of iterations (n) is increased, the computation time for derivation of OF vectors also increases, which is not quite appreciable in real time applications.Thus as already mentioned for virtual images α = 100 and iteration number n = 100 are adequate for reasonable accuracy. For real images obtained from camera of 288 x 352 resolution, α = 30 and iteration number =100 seem to be appropriate for the same accuracy.



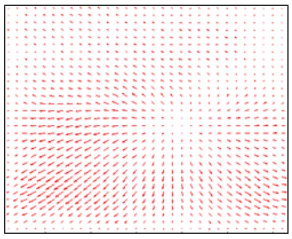
α =100 and n =100



α =200 and n =100



α =300 and n =100

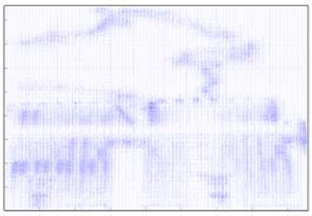


α =400 and n =100

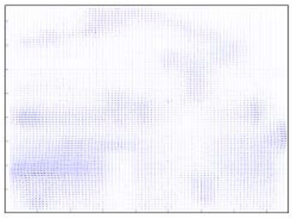
Fig. 9 Variation of OF vectors with change in Weight/Compensation Factors

Variation of Resolution

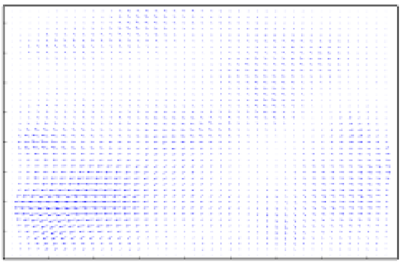
In the computation of OF vectors, resolution of input image plays an important role. This section discusses the results obtained for same images but of different resolution. Pyramid level means something which reduces the resolution of an image by half. For example, if the given image is of resolution 512 x 512, pyramid level I reduces the resolution to 256 x 256 and pyramid level II reduces the resolution to 128 x 128. Figure 10 shows the OF vectors for the images of Figure 7 but for different resolutions.



Original resolution – 512X512



Pyramid level 1 – 256X256



Pyramid level 2 – 128X128

Fig. 10 Variation of OF vectors with change of resolution of input Images

From the results of Figure 10, it can be clearly observed that OF vectors of original resolution is very dense. The vectors in pyramid level I is still denser compared to that in pyramid level II. Moredenser the vectors, more difficult will be to detect the exact point of divergence. The concept of Focus Of expansion (FOE) is based on the determination of the point of divergence. FOE is discussed in the next section. For example, the optical vectors of the original resolution are very dense and will have multiple divergence points, which will mislead in determining the FOE and will result in incorrect results. From the extensive simulation results, it is reasonable to infer that pyramid level 2 can yield OF vectors of reasonable accuracy.

4. FOCUS OF EXPANSION

Method1: Match filter based FOE

The Focus Of Expansion (FOE) [5] represents the point in the image plane where the OF vector starts diverging and has zero or minimum magnitude. This point corresponds to the intersection of the three dimensional (3D) velocity vectors describing the camera movement and the projection plane. The FOE plays an important role in many vision applications such as three-dimensional reconstruction, range estimation, Time-To-Contact (TTC) computation and obstacle avoidance. If the camera is facing in the same direction as the direction of motion, then this direction is what is commonly known as Focus of expansion since it is a point from which optical flow diverges. In this paper, the Match Filter (MF) based technique is applied to compute the FOE since MF technique is robust and can give effective results even for translational and rotational movement of the camera.

Consider a camera moving at a constant velocity, $\vec{V} = (V_x, V_y, V_z)^T$, along its optical ray toward a fixed point, $P = (X, Y, Z)^T$. The FOE is the pixel (x_{FOE}, y_{FOE}) in the image corresponding to the projection of *P* onto the image plane. FOE is characterized by a flow vector with a zero magnitude and the OF field is radially divergent from it. The OF vector has a larger magnitude toward the periphery and a smaller magnitude near the FOE**.** It should be noted that the radial divergence property gives enough information to detect the location of the FOE in the image plane, implying that the magnitude of the OF vectors can be ignored [5]. Moreover, there is no need to know the exact OF vector at each image point, and only the distribution of the vectors is sufficient. Based on the properties mentioned above, the FOE can be detected using the OF and the matched filter. Consider a filter size (2w + 1) × (2w + 1), representing a Cartesian grid with its origin in the centre. Each pixel represents the angle between its corresponding grid point and the origin, as shown in the Figure 11, and also the filter attempts to match only directions. Formally it is given by

$$F(m,n) = \arctan\frac{n}{m} \qquad -w \le m \le w \qquad -w \le n \le w \qquad \text{----- (23)}$$

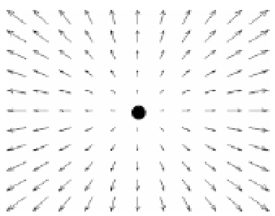


Fig. 11 Radial directions with respect to the central point (marked by a black dot and representing the FOE)[5]

Given two images, I_1 (x, y) and I_2 (x, y), taken $\Delta t \rightarrow 0$ time apart, it is assumed that the OF is computed from them and represented by two images, u(x,y) and v(x,y), corresponding to the flow along the x and y axes, respectively. The FOE is the pixel (x,y) in the optical flow image plane that minimizes the target function defined by [5]

$$(\hat{x}_{FOE}, \hat{y}_{FOE}) = \underset{(x,y)}{\arg\min}\, S(x,y) \qquad \dots (24)$$

Using matched filter F in an ideal manner, S(x, y) should be the sum of squared differences between F and the corresponding directions of optical flow induced by a neighborhood of $(2w + 1)^2$ pixels with (x, y) in the centre. Since the OF vector can be of low quality, there should be an option to select which pixels will be taken into account and what weights be assigned to them. For example, a larger weight value might be given to the pixels in the peripheries since they have a larger magnitude and thus be less sensitive to noise. Another example would be the computation of OF vector using a small number of iterations, resulting in many pixels having approximately zero value of flow magnitude. Using these considerations we define S (x, y) to be [5]

$$S(x,y) = \Psi(u(x,y),v(x,y)) \cdot \sum_{m=-w}^{w}\sum_{n=-w}^{w}\left[F(m,n) - \alpha(u(x+m,y+n),v(x+m,y+n))\right]^2 \cdot \phi(u(x+m,y+n),v(x+m,y+n)) \quad \dots (25)$$

where,

$$\alpha(u(x,y),v(x,y)) = \arctan\frac{v(x,y)}{u(x,y)} \qquad \dots (26)$$

where *α (u(x, y), v(x, y))* is the direction of the OF vector corresponding to the (x, y) pixel and Φ(u(x, y), v(x, y)) is a weight function. In this paper, the following weight function is used [5]

$$\phi(u(x,y),v(x,y)) = \begin{cases} 1 & u(x,y)^2 + v(x,y)^2 \ge a^2 \\ 0 & otherwise \end{cases} \qquad \dots\dots\dots (27)$$

where a is a predefined threshold value. Usually, a is approximately zero, which implies that a pixel having close to zero OF vector value or noise does not contribute to the overall sum. This enables us to use a low-quality estimation of optical flow.

Thus, Ψ (u (x, y), v (x, y)) can be formally stated by [5]

$$\Psi(u(x,y),v(x,y)) = \left(\sum_{m=-w}^{w}\sum_{n=-w}^{w}\phi(u(x+m,y+n),v(x+m,y+n))\right)^{-1} \qquad \dots\dots\dots (28)$$

Ψ (u (x, y) , v (x, y)) is the number of neighbors that actually participated to form the sum of the weighted squared differences for the (x, y) pixel

Method 2: Determination of FOE with clustering method

As will be discussed later, using the conventional MF based technique. It was observed that the FOE point lacked stability when simulated over a number of video frames. Thus to overcome this drawback, a clustering method is proposed in this paper. This process is incorporated before the computation of FOE and after obtaining the u, v matrices.

Algorithm for computation of FOE using clustering method is as explained below:

- Compute the horizontal (u) and vertical (v) components of OF vector applying Horn and Schunck algorithm for two frames.
- Consider the magnitude response of (u,v) vector components: $\sqrt{(u^2+v^2)}$
- Define the least minimum of the elements got from the magnitude response matrix with a suitable amount of offset set to the boundaries. This is because we have experienced approximately zero magnitude vectors along the boundaries of the frame (virtual videos).
- Define three levels of thresholds to obtain cluster of data points
- Three levels of thresholds are:
    - To check if the magnitude response of (u,v) vector components along with the least minimum element lie within 10^{-5}, or 10^{-2}, or 10^{-1} (maximum flexibility) expansion.
- If the magnitude of (u,v) vector satisfies any of above mentioned condition, then the corresponding data points are collected to form a cluster. The cluster method defines that the set of data points whose magnitude response is minimal are the potential data points whose vector response (u,v) can be considered further for FOE computation. The data points with higher vector magnitude response can be neglected.
- Further on, only these set of data points are considered further for the computation of FOE based on the match filtering concept as explained in section IV.
- From the minimum angle information matrix obtained from FOE computation, the least minimum is computed. And this is defined as the substantial point of FOE.
- From the calculated FOE, the range estimation (or TTC) is carried out based on the concept defined by Ted Camus and Didi Sazbon et al [1] on the optical flow images

With various simulation studies applying the clustering method on a set of virtual videos, it has been observed that the FOE points obtained over the continuous run of frames remain stable. Also the FOE point potentially falls in the area of lower magnitude of OF vectors that show pattern of divergence from the point of minimum magnitude. With the inclusion of clustering method the speed of computation for the determination of FOE has drastically improved. The reason being, processing of the entire (u, v) matrices which span the same size of input frame is minimized by considering only the lower magnitude response (u,v) components and excluding the higher magnitude response (u,v) components. This in turn reduces the process time involved in convolving the matched filter window through the entire (u,v) vector matrix. Thus it is concluded that the clustering approach for computation of FOE provides a better and stable FOE points with reduced computations.

Results on Simulation of FOE :

As already mentioned, FOE represents the point in the image plane where the OF vector starts diverging and has zero or minimum magnitude. Figure 12 shows both location of FOE on both the OF and also on the image. Here the FOE is obtained based on the MF concept. A 7x7 matrix is chosen. that slides on the OF vectors and match with a point which has zero or minimal magnitude with its adjacent pixels having increasing magnitude and diverging in all directions

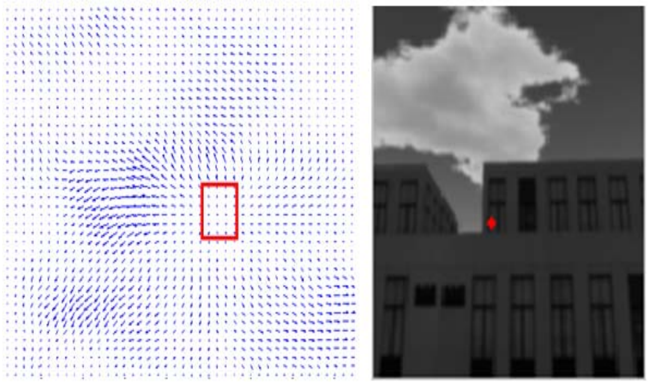


Fig. 12 FOE obtained from Matched Filter Technique

The FOE computed with Matched Filter concept has one drawback that it does not provide stability in the point for consecutive image sequences. Figures 13 to 15 show the variation of x - y co-ordinates of FOE.

Figure 13 illustrates the variation of x-y coordinates of FOE with varying weight/compensation factors (**α**) from 100-400 with fixed iteration value n= 100

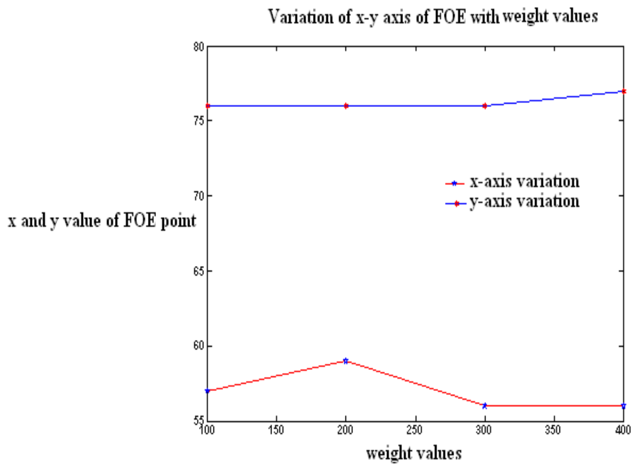


Fig. 13 Variation of x-y co-ordinates with variation in Weight/Compensation values

The influence on variation of x-y coordinates of FOE with varying iteration values from 100-400 with fixed weight value 100 is depicted in Figure 14.

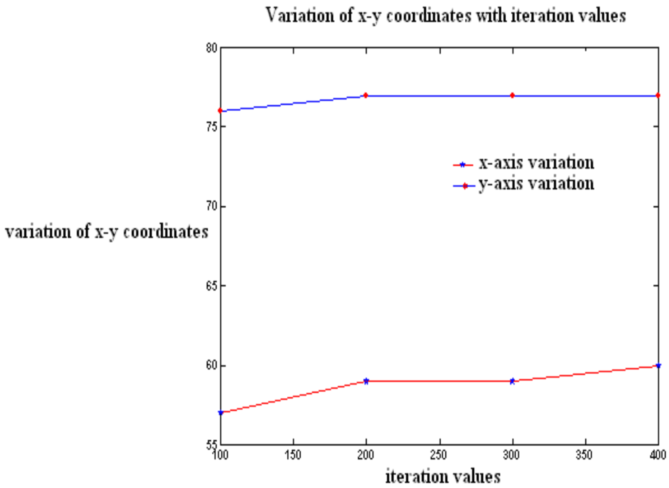


Fig. 14 Variation of x-y co-ordinates with iterations values (n)

The effect of change in image resolution (pyramid levels) on the variation of x-y coordinates of FOE is shown in Figure 15.

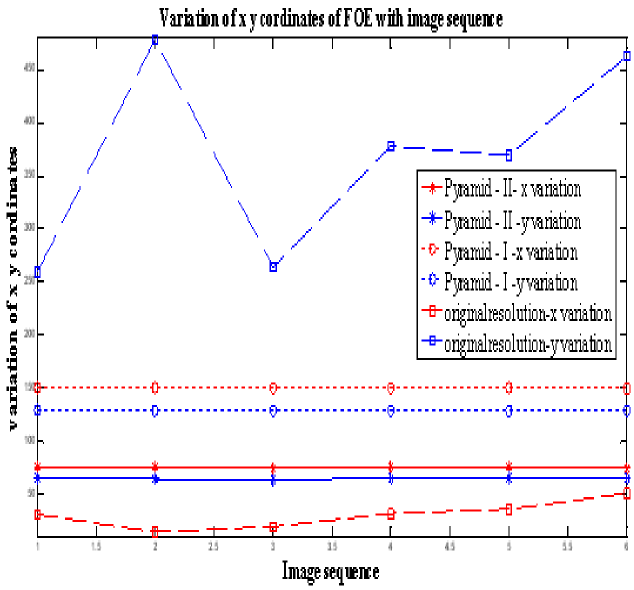
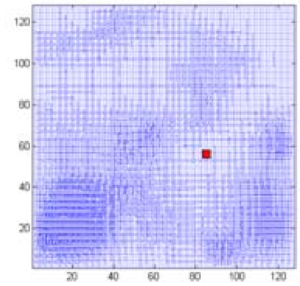
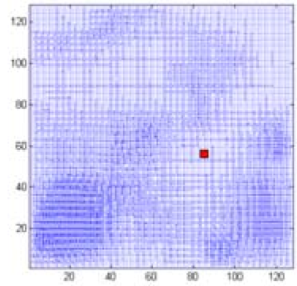
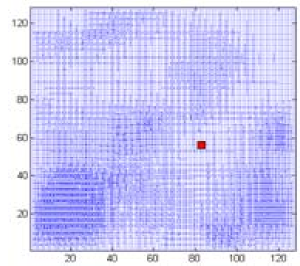
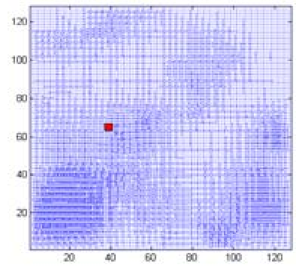
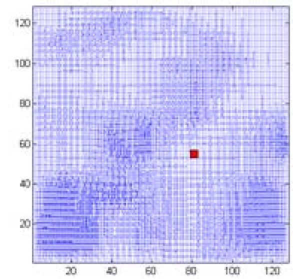
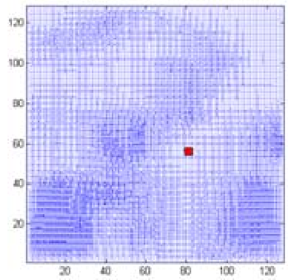
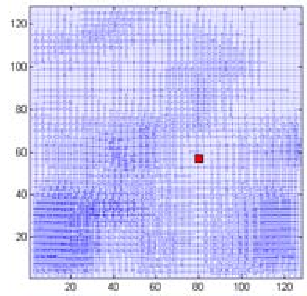
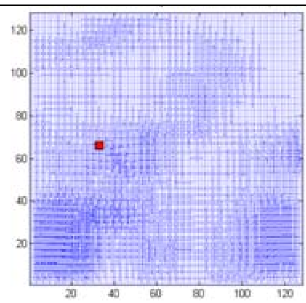


Fig. 15 Variation of x-y co-ordinates with pyramid levels

Table 1 facilitates the comparison of FOE point obtained by applying both non-clustering (MF) method and clustering method for computation of FOE. Table 1 shows the pictorial form of FOE projected over the optical flow vector for a set of consecutive frames.

Table 1. Comparison of Clustering and Non-clustering method for computation of FOE

Clustering Method	Non-clustering Method
 FOE point 56X85	 FOE point 56X85
 FOE point 56X83	 FOE point 65X39
 FOE point 56X81	 FOE point 56X81
 FOE point 57X80	 FOE point 66X33

The results of Table 1 amply substantiates that the change in FOE point for successive frames using proposed clustering method is negligibly small when compared to that predicted by Non Clustering method.

5. Determination of Time To Contact

Ted Camus's Approach

The method proposed by Ted Camus [1] is widely used for determining the TTC. Figure 16 describes the optical geometry (the classic reference for the image- plane-coordinate system). A point of interest P at coordinates (X, Y, Z) is projected through the focus of projection centered at the origin of the coordinate system (0.0.0). P is fixed in a physical space and does not move.

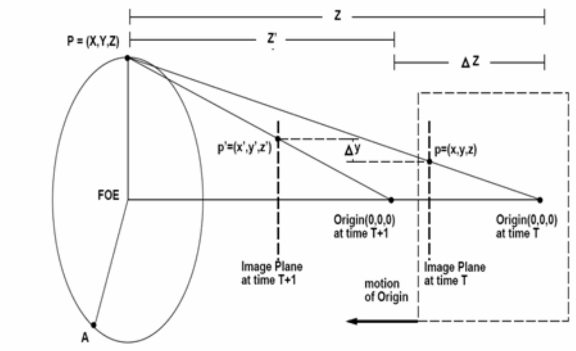


Fig. 16 Optical Geometry [1]

The following are valid in view of the stated assumptions [1].The origin move forward with a velocity dz/dt. The image plane is fixed at a distance Z in front of the origin. The image plane moves along with the origin. P projects onto point p in this plane. As the image plane moves closer to P the position of the image plane changes as well using the similar triangles,

$$\frac{y}{z} = \frac{y}{1} = \frac{Y}{Z} \quad \text{……… (29)}$$

Differentiating with respect to time

$$\dot{y} = \frac{\dot{Y}}{Z} - Y\left(\frac{\dot{Z}}{Z^2}\right) \quad \text{……… (30)}$$

Since p is immobile, set $\dot{Y} = 0$ and substituting (yZ) for Y,

$$\dot{y} = -y\left(\frac{\dot{Z}}{Z}\right)$$

$$\frac{y}{\dot{y}} = -\frac{Z}{\dot{Z}} = \tau \quad \text{……… (31)}$$

Where τ is Time to Contact

Time To Contact based on Match Filter concept:

Time to Contact is the one of the application of OF [5]. Using the OF vectors one can calculate the collision time between the vehicle and the obstacle.

Suppose that the origin of the 3D world is on the centre of the camera and the FOE is the pixel with the (0, 0) coordinates. Also, one can assume that the camera is moving in a known constant velocity V = (V_x, V_y, V_z) and has a known focal length f [5].

$$v = \frac{dy}{dt} \qquad u = \frac{dx}{dt} \quad \text{(32)}$$

$$V = \frac{f}{Z}\left(\dot{Y} - Y\dot{Z}\right) \quad \text{……. (33)}$$

By using the following relation

$$\dot{Y} = -V_y \qquad \dot{Z} = -V_z \quad \text{(34)}$$

$$V = \frac{f}{Z}\left(-V_y - \frac{YV_z}{Z}\right) \quad \text{…… (35)}$$

Multiplying Z to the above equation no 35

$$VZ^2 + fV_yZ + YV_Z = 0 \quad \text{…… (36)}$$

After solving the above equation 36

$$Z = \frac{f}{V}\left[V_y - YV_Z\right] \quad \text{………..(37)}$$

Z is the distance between the camera and the obstacle, and then TTC is given by the following equation

$$TTC = \frac{Z}{V} \quad \text{…….. (38)}$$

A flowchart for TTC computation is shown in Figure 17

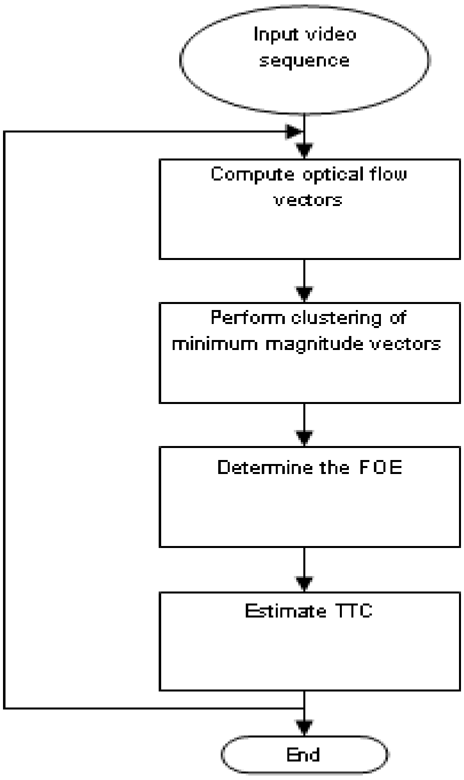


Fig. 17 Flow chart for TTC calculation

6. DECISION LOGIC

Two decision logics are presented in this paper. One based on vector magnitude $\sqrt{(u^2 + v^2)}$ and the other based on an Region Growing (Image Segmentation) principle. The magnitude based decision logic is based on the fact that for a constant background, the OF vectors have no magnitude or direction. The vehicle is made to navigate along the direction of minimal magnitude. This logic gives a three directional movement - left, straight and right. In the region growing based technique, the FOE is used to detect the obstacle. Using the x and y co-ordinates of the FOE point, the neighboring (N-4 and N-8) pixel connectivity is computed. It is known that the obstacles will have a different intensity compared to free space. The region growing method marks the boundary of the obstacle and all the pixels with equal intensity on the obstacle will be connected. By setting a higher threshold, the entire obstacle can be clearly isolated in the image. This logic gives nine directional movements– top, bottom, right, left, top-left, straight, top-right, bottom-left, bottom-right.

Results: Decision Logic

In the magnitude based technique, as explained in section V the UGV just follows the direction with minimal magnitude (Figure 19). Whereas for the MAV navigation based on region growing technique, the image pixels indicating the obstacle will be given a value of 1 (white) and the rest of the image will be given 0 (black). The resultant image is divided into three columns and 3 rows. Whichever segment of the image has highest number of zeros or maximum black area, the MAV will navigate along that direction. This logic gives a 9-direction movement (Figure 20).

Image1 Image2

Fig. 18 Input frames

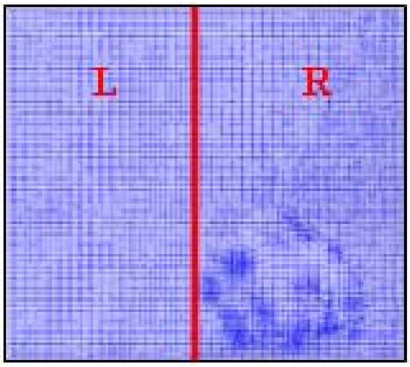


Fig. 19 UGV decision logic

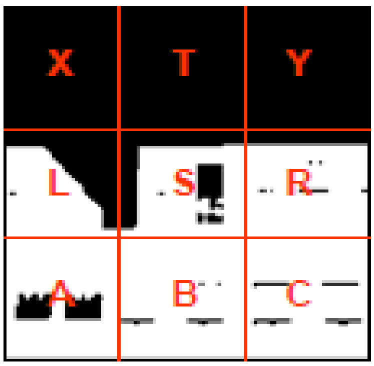


Fig. 20 Decision Logic Based on Region Growing Technique

7. CONCLUSION

Based on the extensive literature survey, analysis and implementation of several OF algorithms, it is demonstrated that Horn and Schunck algorithm is a reasonable and satisfactory compromise for real time applications. The influence of the choice of Weight/Compensation factor (α), the number of iterations (n) and the resolution of Input Image(Pyramid level) on the accuracy and reliability of OF vectors derived through Horn and Schunck algorithm is analyzed through extensive simulation studies. Based on the simulation results of the study undertaken, it is inferred that α of 100 is good for virtual images and 30 for real images with a camera of resolution 288x352. This paper also proposes a new approach called Clustering Method for the determination of FOE overcoming the instability of the earlier (Match Filter) method. The proposed Cluster based method is also found to be computationally efficient. The simulation results of Clustered method shows relatively insignificant variations of FOE position over successive frames thereby exhibiting improved stability. For autonomous navigation of unmanned vehicles, separate decision logics namely vector magnitude scheme for UGV and Region Growing based approach for MAV have been proposed in this paper.

8. REFERENCES

[1] Camus T, *"Real Time Optical Flow Computation",* PhD Thesis, Brown University, 1994

[2] Bachman S.S. and Barron J.L*, "The Computation of Optical Flow*" , ACM *Computing* Surveys , Vol. 27, No. 3, pp. 433-467, 1995

[3] Toby Low and Gordon Wyeth, "*Obstacle Detection using Optical Flow"*, Proceedings of the Australasian Conference on Robotics & Automation, 2005.

[4] Horn B and Schunck B, "*Determining Optical Flow*", Artificial Intelligence, 17:185-203,1981

[5] Sazbon D, Rotstein H, Rivlin E, *"Finding the Focus of Expansion and Estimating Range using Optical Flow Images and a Matched Filter"*, Machine Vision and Applications , Vol. 15, pp. 229–236, 2004