A System Model Based on Slantlet Transform to Estimate Optical Flow

Dr. Atheer A. Sabri[1] **Hamsa D. Majeed[1]**

[1]**Electrical Engineering Department, University of Technology, Baghdad**

email:atheeralaa@yahoo.com, hdm_electronic@yahoo.com

Received: 20/9/2012

Accepted: 22/7/2013

Abstract– The estimation of optical flow is the basic step for many engineering applications that exploit the image processing field as a part of their models. In this paper a model called Slantlet Based Optical Flow Estimation (SLT_OFE) is proposed to estimate the optical flow. Slantlet Transform (SLT) used as an effective tool, 2D and 3D- SLT- Level 2 (SLT2) are computed and employed in the proposed model to provide high accuracy estimation of the optical flow. By its definition, optical flow is a velocity field, so the velocities in this paper are computed using the widely used Differential Technique. Two methods from this technique are adopted; Horn-Schunck Method and Lucas-Kanade Method. The optical flow is estimated for two types of image sequences; synthetic sequences and real sequences. Unlike the real sequences, the synthetic sequences have known true velocities which are used for evaluating the proposed model by calculating mean error (Mean Err.), angular error (Ang. Err.) and Standard Deviation (STD). For extreme study of its performance, the proposed SLT_OFE is compared with three other models that are based on level two (Discrete Wavelet Transform (DWT), Discrete Multi Wavelet Transform (DMWT) and Framelet Transform (FT)) which are implemented in this paper and employed in the conventional models, 2D-DWT2_OFE, 2D-DMWT2_OFE and 2D-FT2_OFE. The results show that the proposed model offers minimum values in errors and STD when 2D-SLT2 is used, and these results are improved by using 3D-SLT2. This leads to the fact that the proposed model SLT_OFE through both of 2D and 3D approaches possesses an improvement in the optical flow estimation process with higher accuracy, than the other models produced in the same circumstances. MATLAB Version 7.12 (R2011a) is used to implement the proposed model and the conventional models.

Keywords – Optical flow estimation, Differential technique, Slantlet Transform.

1. Introduction

Motion image analysis is one of important computer vision tasks, and optical flow estimation is the key technology for motion image analysis. The optical flow is a type of motion representation that assigns to every point in the visual field a two-dimensional instantaneous velocity vector. It is considered to be one of the most detailed and rich motion representations of an image from a video signal. The estimation of optical flow is one of the fundamental problems in computer vision, and accurate calculation of optical flow has been the goal of many research efforts in the last three decades. Indeed, the information provided by optical flow is comprehensive enough to be used in many tasks such as object detection and tracking, image dominant plane extraction, movement detection; robot navigation [1,2,3]. Many authors noticed that a good way to enhance the reliability of optical flow estimation is to perform a multi-scale computation. This makes wavelets a well-designed tool for optical flow measurement. The wavelet transform has become a useful computational tool for a variety of signal and image processing applications [4,5,6]. The discrete wavelet transform (which transforms a discrete time signal to a discrete wavelet representation)is usually carried out by filter bank iteration. However, for a fixed number of zero moments, this does not yield a discrete-time basis that is optimal with respect to time-localization. Selesnick [7] proposed a wavelet like filters known as Slantlet filters which can provide better time localization and better signal compression compared to the conventional DCT and classical DWT, it is orthogonal with two zero moments and retains the basic characteristic of the usual filter bank such as octave band characteristic, a scale dilation factor of two and efficient implementation. However, the SLT is based on the principle of designing different filters for different scales [6,7,8].

Optical flow estimation is still an active field. Many techniques and methods have been proposed for the past three decades; others continue to appear. In 1980-1981 Horn and Chunks [9] developed a method based on the observation that the flow velocity has two components and that the basic equation for the rate of change of image brightness provides only one constraint. Global smoothness of the flow was introduced as a second constraint; also Locus and Kanade [10] present a new image registration technique that uses spatial intensity gradient information to direct the search for the position that yields the best match. They implemented a weighted least square of local first order constraints and by taking more information about the images into account. Many techniques and methods proposed over the years were summarized and compared by Barron et .al [11]. They proposed a set of real and synthetic image sequences and reported the results of a number of regularly cited optical flow techniques and the comparison, showed that the performance can differ significantly among the techniques and observed that the differential method is more accurate than other techniques. S. Baker et .al [12] established a new set of benchmarks and evaluation methods for the next generation of optical flow algorithms by updating the set used by Barron et al. using even more challenging sequences. Other researchers uploaded their results which lead to other versions of methods evaluation.

2. Optical Flow Estimation

The optical flow field can be deduced from the differential technique that uses spatial and temporal derivatives of image intensity. This technique is the most widely used for optical flow computation [2].

2.1Constraint Equation

The first rule that governs the evolution of intensity can be expressed as the brightness constancy constraint equation:

$$I_x u + I_y v + I_t = 0 \quad \ldots (1)$$

where (I_x, I_y, I_t) are the intensity of the image $I(x, y, t)$ and (u, v) is the horizontal and vertical velocity components respectively. Because of presence of two unknown components of velocity related in only one linear equation, at least one other constraint is necessary for calculating the flow [11,12] .

2.2 Horn and Schunk`s Global Method

Horn and Schunk [9] used the smoothness of the velocity field with a global smoothness term. The optical flow field is computed by minimizing error that is a function of an image constraint and a smoothness constraint:

$$E = \iint (I_x u + I_y v + I_t)^2 + \lambda(|\nabla u|^2 + |\nabla v|^2)\, dxdy \ldots\ldots (2)$$

where the magnitude of λ reflects the influence of the smoothness term [2,11] .

2.3 Lucas and Kanade`s Local Method

Lucas and Kanade [10] use a weighted least square of local first order constraints error minimization approach to find the minimal constraint error:

$$E^2 = \sum_{(x,y)} G^2(x,y)[V(u_x, v_y).\, \nabla I(x,y,t) + I_t(x,y,t)]^2 \quad \ldots (3)$$

Where E^2 is the squared error and $G^2(x, y)$is the squared weighting function [2,11].

From equation (3), a linear system is defined as [11,13] :

$$A^T G A V = A^T G b \ldots\ldots \quad (4)$$

For a number of points denoted by (p) in the local neighborhood of pixels (g(x, y)), the vectors A, b and matrix G can be defined as below:

$$A = [\nabla I\ (x_1, y_1)\ , \ldots\ , \nabla I\ (x_p, y_p)]^T \quad .. (5)$$

$$G = \begin{bmatrix} g(x_1, y_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(x_p, y_p) \end{bmatrix} \quad .. (6)$$

$$b = \Box\ [I_t(x_1, y_1), \ldots, I_t(x_p, y_p)]^T \quad \ldots (7)$$

Re-arranging equation (4) and solving for V then:

$$V = [A^T G b]\, [A^T G A]^{-1} \quad \ldots\ldots (8)$$

3. Slantlet Transform

The Slantlet Transform (SLT) is a recently developed multiresolution technique especially well-suited for piecewise linear data. SLT has proved that it is a better candidate for signal compression as compared to the conventional DCT and DWT based method and SLT based algorithms retain higher percentage of energy after compression compared to the DWT approach. Moreover, SLT has been used in medical image processing for the classification of magnetic resonance human brain images, also presented as based method for image finger prints

[8,14].

3.1 Slantlet Filterbank

The Slantlet filters are based on the parallel structure of DWT filter bank as shown in Figure (1). Slantlet is employing different filters for each scale whereas DWT is usually implemented in the form of an iterated filterbank, utilizing a tree structure. With this extra degree of freedom obtained by giving up the product form, filters of shorter length are designed satisfying orthogonality and zero moment conditions.

3.2 The Proposed SLT Computation Method

In order to compute SLT, two issues must be considered. Firstly, the length of the input signal should be power of two and equal or greater than the length of analysis filterbank of SLT, that is because all filters' lengths in SLT filterbank are power of two, and the input signal should be compatible with their lengths. Secondly, the transformation matrix must be constructed. The construction of the transformation matrix is accomplished according to the SLT level and the length of input signal. In this work, a transformation matrix (A) shown in equation (9) is suggested and employed for SLT computations.

$$A=\begin{bmatrix}
h_2(0) & h_2(1) & h_2(2) & h_2(3) & h_2(4) & h_2(5) & h_2(6) & h_2(7) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & h_2(0) & h_2(1) & h_2(2) & h_2(3) & h_2(4) & h_2(5) & h_2(6) & h_2(7) & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\
h_2(4) & h_2(5) & h_2(6) & h_2(7) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h_2(0) & h_2(1) & h_2(2) & h_2(3) \\
f_2(0) & f_2(1) & f_2(2) & f_2(3) & f_2(4) & f_2(5) & f_2(6) & f_2(7) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & f_2(0) & f_2(1) & f_2(2) & f_2(3) & f_2(4) & f_2(5) & f_2(6) & f_2(7) & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\
f_2(4) & f_2(5) & f_2(6) & f_2(7) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & f_2(0) & f_2(1) & f_2(2) & f_2(3) \\
g_1(0) & g_1(1) & g_1(2) & g_1(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_1(0) & g_1(1) & g_1(2) & g_1(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & g_1(0) & g_1(1) & g_1(2) & g_1(3) \\
g_{1r}(0) & g_{1r}(1) & g_{1r}(2) & g_{1r}(3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_{1r}(0) & g_{1r}(1) & g_{1r}(2) & g_{1r}(3) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & g_{1r}(0) & g_{1r}(1) & g_{1r}(2) & g_{1r}(3)
\end{bmatrix} \quad \ldots(9)$$

Where h and f represent the lowpass and highpass filters respectively, while g was denoted as support filter and g_r is the reversed-shifted version of g.

The filter coefficients used in the slantlet filter bank (the derived equations can be found in [7]) are:

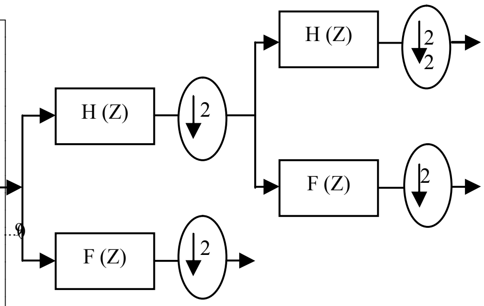
h_2= [(0.2698), (0.3948), (0.5198), (0.6448), (0.2302), (0.1052), (-0.0198), (-0.1448)] ….(10)

f_2= [(-0.0825), (-0.1207), (-0.789), (-0.1971),(0.7533), (0.3443), (-0.0648), (-0.4738)] ……(11)

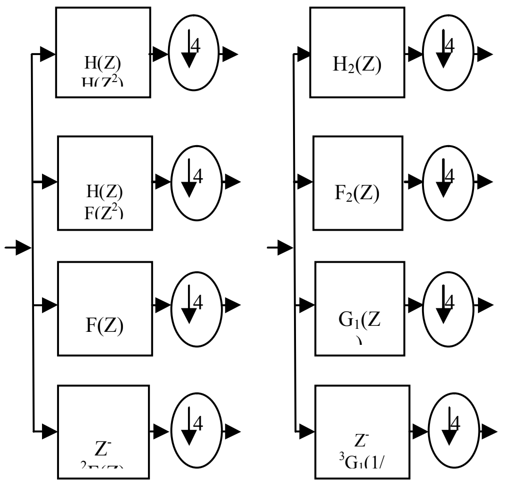
g_1= [(-0.5117),(0.8279),(-0.1208),(-0.1954)] ………………….(12)

g_{1r}= [(-0.1954),(-0.1208), (0.8279), (-0.5117)] …………………(13)

The number of rows for each filter in this matrix depends on the dimensions of the matrix, which in turn depends on the dimensions of the input signal. The blank entries are signifying zeros.



(a) DWT itrated filter level 2



(b) Comparison of level 2 iterated Db4 filterbank (left-hand side) and level 2 Slantlet filterbank

Figure (1) Illustration of the Slantlet Filters

3.2.1 SLT2 for 1D (1D-SLT2) and 2D (2D-SLT2) input signal

To compute SLT2, first, suppose X to be the input signal with length N for 1D (or N×N for 2D).

The computation process is presented in the following procedure:

1) Construct the transformation matrix (A) with dimensions N×N.
2) The transformation for 1D is

$$[X]_{1\times N} \cdot [A]_{N\times N} = [Y]_{1\times N} \qquad \dots (14)$$

while for 2D is

$$[X]_{N\times N} \cdot [A]_{N\times N} \cdot [A^T]_{N\times N} = [Y]_{N\times N} \qquad \dots (15)$$

where Y is the resultant transformed signal

3.2.2 SLT2 of 3D input signal (3D-SLT:

Suppose 3D input signal is X with length N×N×M where N×N is the dimensions of the frames, and M is the number of frames in the input signal X. The 3D –SLT2 computation is combining both 1D and 2D SLT as illustrated in the procedure below:

1) Apply 2D- SLT2 to each N×N frame of the input signal X. the result is a signal with a dimensions of N×N×M.
2) The final 3D-SLT2 transformed signal is obtained by applying 1D- SLT2 to each element resultant from the 2D-SLT2 for all the M matrices in z direction.

4. The Proposed Algorithm for Optical Flow Estimation

To see the benefits of Slantlet transform, it is used to calculate the optical flow estimation as shown in figure (2) and is described in the following steps:

1) input set of images: the number of images read from the specific sequence depends on the way that the partial derivatives are computed and the SLT transform performed (2D or 3D), as below:

a) for 2D and cube differencing 2 images are required.
b) for 2D and central differencing 5 images are required.
c) for 3D and cube differencing 9 images are required.
d) for 3D and central differencing 12 images are required.

2) Slantlet Transform: this transform is considered as a pre-smoothing step in the optical flow estimation process. In the present work, 2D and 3D level 2 Slantlet transform are computed.

3) Derivative Calculations: in this step the partial derivatives (I_x,I_y,I_t) are computed in two different types of differencing :

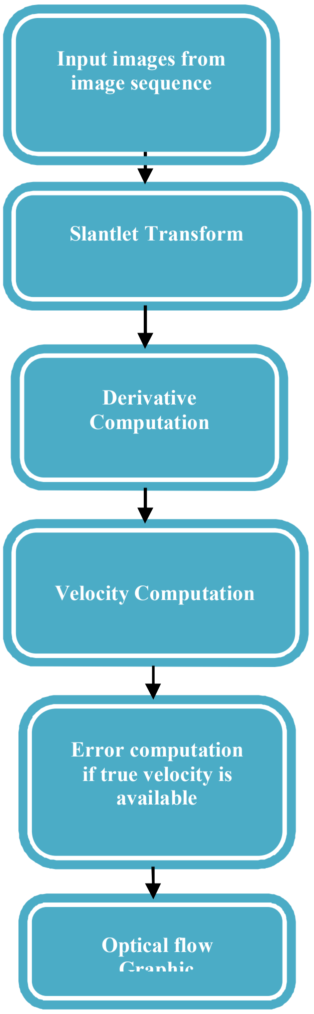
a) Cube differencing: the partial derivatives are obtained by averaging the first four differences taken over a cube with side length of (2) pixels.

b) Central differencing: the partial derivatives are computed by taking differences for two pixels on both sides of the central pixel (with weights of $^2/_3$ and $^1/_{12}$ respectively).

4) Velocity Computation: in this step the optical flow velocity is computed according to:

a) Horn and Schunk`s Global Method.

b) Lucas and Kanade`s Local Method.



Figure(2) Block diagram of the proposed model

5. Experimental Results

Computer programs are performed using MATLAB V. 7.12(R2011a). In this paper two types of image sequence are used as listed below (image sequences are downloaded from *ftp.csd.uwo.ca* in the directory */pup/vision*).

1) Synthetic Image Sequence: These sequences have true 2D motion field. They are clean sequences with true velocity. In the present work, two synthetic sequences are used as listed below and shown in Figures (3) and (4).

a) Horizontal Lines: In this sequence black and white horizontal lines are interferenced by vertical movements with true velocity (1, 1).

b) Cloudy Mountains: In this sequence cloudy sky is moving over few mountains with true velocity (1, 1).

2) Real sequences: These sequences are from real scenes and don`t have true velocities because the 2D motion field is estimated, not real and the sequences not clean (shadowing and transparency). In this work, two real sequences are used as listed below and shown in Figures (5) and (6).

a) Flower Garden: In this sequence, the camera translates parallel to the ground plane, perpendicular to its line of sight and in front of a tree and cluster houses.

b) Crossing Arms Sequence: In this sequence, there was a human crossing both his hands over his head and moving his right arm towards the table next to him.

Three different transformations are implemented and employed in the proposed model for comparison with SLT to estimate optical flow; these are: Discrete Wavelet Transform, Discrete MultiWavelet Transform and Framelet Transform, all 2D-Level 2.

6. Conclusions

In this paper, an SLT based system has been proposed as an effective tool to get more accurate estimation of the optical flow. Moreover, SLT produces for each level L a sample reduction equal to (2L-2) in each filter of SLT filterbank versus the DWT filterbank. This reduction reaches to two thirds (2/3) as L increases. By using SLT-Level 2 in this work, reduction by two samples is achieved. This reduces the computational complexity by reducing the number of multiplications and additions required for the implementation of the transform. Furthermore, SLT compacts the signal energy into a small number of coefficients. SLT compacts the signal energy into a low frequency subband (L_1L_1). This means that only 1/4 of image size is used in the computation of optical flow leading to simplicity in the calculation, reducing the time of program implementation and less memory size is required. From Errors and STD calculations listed in Tables (1) and (2), it is obvious that minimum values of Errors and STD are computed from using SLT in the model. The effect of the differencing used is noticeable; using SLT in the system gives better results using cube differencing for Horn-Schunck Method. On the other hand, Lucas-Kanade method gives better results with central differencing. These results are improved in 3D for both differencing used as shown in Tables (3) and (4).

Table (1) Summary of Horizontal Lines Sequence Results.

Horizontal lines							
Difference Type	**Systems**	**Horn-Schunk Method**			**Lukas-Kanade Method**		
		Mean Err.	**Ang. Err.**	**STD**	**Mean Err.**	**Ang. Err.**	**STD**
cube	2D-SLT2_OFE	0.085	5.26	0.015	0.089	3.047	0.006
	2D-DWT2_OFE	0.094	5.184	1.31	0.124	6.123	2.596
	2D-DMWT2_OFE	0.12	5.273	1.254	0.123	6.092	2.378
	2D-FT2_OFE	0.082	5.259	0.012	0.084	2.82	0.12
central	2D-SLT2_OFE	0.082	4.624	0.086	0.052	4.605	0.004
	2D-DWT2_OFE	0.096	5.817	3.254	0.042	6.275	1.755
	2D-DMWT2_OFE	0.092	6.092	2.374	0.037	6.749	1.717
	2D-FT2_OFE	0.084	4.669	0.072	0.055	4.748	0.012

Table (2) Summary of Cloudy Mountains Sequence Results.

Cloudy mountains							
Difference Type	**Systems**	**Horn-SchunkMethod**			**Lukas-KanadeMethod**		
		Mean Err.	**Ang. Err.**	**STD**	**Mean Err.**	**Ang. Err.**	**STD**
cube	2D-SLT2_OFE	0.075	4.508	3.224	0.102	6.784	3.088
	2D- DWT2_OFE	0.09	10.73	4.876	0.236	12.24	11.24
	2D-DMWT2_OFE	0.083	11.67	4.496	0.227	11.58	8.417
	2D- FT2_OFE	0.054	9.03	1.781	0.187	16.67	2.722
central	2D-SLT2_OFE	0.108	6.57	6.389	0.062	4.695	1.472
	2D- DWT2_OFE	0.209	12.22	3.48	0.234	15.17	9.62
	2D-DMWT2_OFE	0.225	12.77	3.515	0.218	15.26	10.13
	2D- FT2_OFE	0.158	9.614	3.298	0.613	6.371	8.022

Table (3) Horizontal Lines Results for 3D-SLT2_OFE versus 2D-SLT2_OFE Approaches

Horizontal lines							
Difference Type	**System**	**Horn-Schunk Method**			**Lukas-Kanade Method**		
		Mean Err.	**Ang. Err.**	**STD**	**Mean Err.**	**Ang. Err.**	**STD**
cube	3D-SLT2_OFE	0.086	5.006	0.012	0.057	5.028	0.003
	2D-SLT2_OFE	0.085	5.26	0.015	0.089	3.047	0.006
central	3D-SLT2_OFE	0.087	4.12	0.009	0.049	4.604	0.002
	2D-SLT2_OFE	0.082	4.624	0.086	0.052	4.605	0.004

Table (4) Cloudy Mountains Sequence results of for 3D-SLT2_OFE versus 2D-SLT2_OFE approaches

Cloudy mountains							
Difference Type	**System**	**Horn-Schunk Method**			**Lukas-Kanade Method**		
		Mean Err.	**Ang. Err.**	**STD**	**Mean Err.**	**Ang. Err.**	**STD**
cube	3D-SLT2_OFE	0.062	4. 34	3. 189	0. 091	6. 249	2.887
	2D-SLT2_OFE	0.075	4.508	3.224	0.102	6.784	3.088
central	3D-SLT2_OFE	0. 089	6. 178	6. 16	0. 038	4. 301	1.24
	2D-SLT2_OFE	0.108	6.57	6.389	0.062	4.695	1.472

References

[1] L. Bin, D. M. Hui and H. Jinlong, "Color Optical Flow Estimation Based on Gradient Fields with Extended Constraints," International Conference on Networking and Information Technology, IEEE, 2010.

[2] M.Quelin, "Optical flow estimation using insect vision based parallel processing," M.Sc. thesis, University of Wollongong, Australia , 2011.

[3] T. Gkamas and C. Nikou, "Guiding Optical Flow Estimation using Superpixels," 17th International Conference on Digital Signal Processing, Greece: Corfu, July 2011.

[4] H. N. Al-Taai, "Computationally-Efficient Wavelet-Based Algorithms for Optical Flow Estimation,"Ph.D thesis, University of Technology, Iraq, 2005.

[5]Ivan W. Selesnick, "Wavelet Transforms -A Quick Study," Polytechnic University, Brooklyn, NY, 2007.

[6] S. K. Meher, "A Novel Power Quality Event Classification using Slantlet Transform and Fuzzy Logic," International Conference on Power System Technology, India, 2008.

[7] Ivan W. Selesnick, "The Slantlet Transform," IEEE Transactions On Signal Processing, vol. 47, no. 5, May 1999.

[8] S.K.Mutt and S. Kumar, "Secure Image Steganography based on SlantletTransform," International Conference on Methods and Models in Computer Science, 2009.

[9] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence, vol. 17, pp.(185-203), 1981.

[10]B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," proceedings of image workshop, 1981.

[11]J.L. Barron, D. J. Fleet and S.S. Beauchermin, "Performance of Optical Flow Technique," IJCV, vol. 12, no.1, pp.(43-77), 1994.

[12] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M. J. Black and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," International Journal of Computer Vision, vol. 92, Issue 1, March 2011.

[13] C. D. McCarthy, "Performance of Optical Flow Techniques for Mobile Robot Navigation," M.SC Thesis, University of Melbourne, Australia, 2005.

[14] N. B. Patil, V. M. Viswanatha and S. Pande, "Slant Transformation as a Tool for Pre-Processing in Image Processing," International Journal of Scientific & Engineering Research, vol. 2, Issue 4, April 2011.