Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 5, May -2015

A NOVEL TECHNIQUE TO DETECT CLOUD MOTION USING OPTICAL FLOW ALGORITHM

Dhaval Patel[1], Savitanandan Patidar[2], Prof.S.D.Joshi[3]

[1] *Electronics and Communication Department, VGEC Chandkheda, Ahmedabad, Gujarat,*
[2]*Electronics and Communication Department, VGEC Chandkheda, Ahmedabad, Gujarat,*
[3]*Electronics and Communication Department, VGEC Chandkheda, Ahmedabad, Gujarat,*

Abstract — *Due to global warming and effect of greenhouse gases, there is a certain changes observed in atmosphere, to study those changes we need proper motion detection technique. There are many operational algorithms to detect motion, all the algorithms use some form of pattern matching techniques. The proposed algorithm based on optical flow is fundamentally different from other motion detection algorithms. An optical-flow based technique requires relatively small displacements so this algorithm is best suited for Satellite application. In this paper we have covered the background of optical flow algorithm. Some modification and Pre-processing had applied before implementing the algorithm. For implementation of optical flow algorithm we have used lucas-kanade method. The rest of the paper is devoted to our findings and a brief discussion about future scope of research.*

Keywords- *Optical Flow, Lucas-kanade, Optical Flow Constraint Equation and Aliasing etc...*

I. INTRODUCTION

For pattern matching two most popular techniques are the cross-correlation and least-squares methods, both techniques find the best agreement between first image and second image. In least squared method, first select initial frame then computes the sum of squared differences for all possible scenes in the search array. The scene with the smallest sum is match. This process is repeated backward and then forward in time before a final average vector is computed from the two individual estimates [1].

In this paper, one new alternative approach to correlation-based methods is proposed the computation of the "optical flow" algorithm. Optical flow algorithm works on two assumptions Brightness consistency and relatively small motion. Change in the intensity pattern is assumed to be the result of a simple transformation of the intensity contours. Second assumption Small motion suggests that high temporal resolution images are required to get trustworthy results [1].

In this paper, Optical flow algorithm is used to determine whether or not it may be used to detect cloud motion from satellite imagery. In section II we discuss the hypothesis of optical flow techniques. A particular method which is used for optical flow algorithm (Lucas-Kanade) is described in section III. Section IV consists of Modification to overcome certain errors in Lucas-kanade method and pre-processing before implementing the algorithm. Finally Results of experiment are drawn in Section V and in Section VI we have summarizing the paper and future scope.

II. OPTICAL FLOW

2.1. Overview

Optical Flow is the 2-Dimention motion field, u(x, y) to v(x, y), for each point (x, y) in an image. To estimate optical flow two types of algorithms that can be used. First, give a set of corresponding points (high frequency points) in two images, displacement vector between these points compute by optical flow. Second, compute optical flow for all points [2].

One of the two Assumptions is "Intensity constancy" use to estimate Optical Flow at all points in an image, which states that the intensity remains constant with time, i.e.

$$I(x+\mathbf{dx}, y+\mathbf{dy}, t+\mathbf{dt}) = I(x, y, t) \ldots\ldots\ldots\ldots (1)$$

Second assumption is small motion between consecutive frames (small **dx**, **dy**), using first order Taylor series equation 1 can be expanded to get [2]

$$I_x u + I_y v + I_t = 0$$

Where $I_x = \frac{dI}{dx}$ and so on. Also $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$.

2.2. Optical Flow Constraint Equation

I(x, y, t) is indicating an image, and the motion of an image pixel $m = [x, y]^T$ is

$$v_m = m = [v_x, v_y]^T = \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix}$$

Intensity is remain consistent for m during time dt, i.e.

$$I(x + v_x\,dx, y + v_y\,dy, t + dt) = I(x, y, t) \ldots\ldots\ldots (2)$$

If smoothly and slowly brightness changes with x, y and t, the left-hand-side of Taylor series can be expend by:

$$I(x, y, t) + \frac{dI}{dx}v_x dt + \frac{dI}{dy}v_y dt + \frac{dI}{dt}dt + O(dt^2) = I(x, y, t)$$

So, we have

$$\frac{dI}{dx}v_x + \frac{dI}{dy}v_y + \frac{dI}{dt} = 0$$

i.e.

$$\nabla I.v_m + \frac{dI}{dt} = 0 \ldots\ldots\ldots (3)$$

Where $\nabla I = [\frac{dI}{dx} \quad \frac{dI}{dy}]^T$

For pixel m this is image gradient, which is obtained from images. Also dI/dt can also be obtained from images easily. This equation(3) is known as optical flow constraint equation [2].

Here, for every pixel, there is only one constraint equation, but two unknowns vx and v_y, which means optical flow don't have unique solution from this optical flow constraint equation. For constraint equation Figure 1 gives a geometrical justification [2].

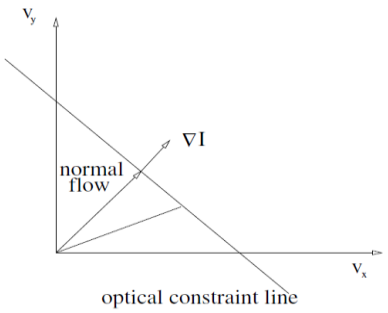


Figure 1. The optical flow constraint equation's Geometrical justification.

From this constrain equation can determine the flow along the direction of image gradient, but cannot determine the direction perpendicular to the image gradient. This phenomenon is known as "aperture problem" [2].

(3) Aperture Problem

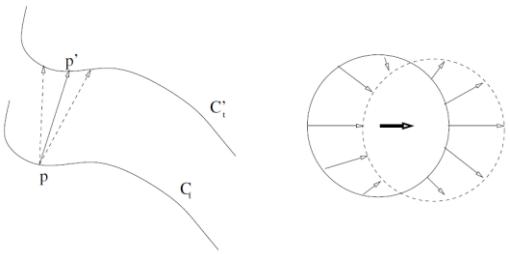


Figure 2. The aperture problem

The aperture problem is stated in above section. Here two examples are given in Figure 2. In first figure many points around p have the same intensity. So, it is impracticable to decide which point p' on Ct' match point p on CT. In second example, only the normal flow determines of ∇I. thus, some other constraints are needed to determine the optical flow uniquely. Below, we describe approaches: employs a local constraint for each pixel. This approach is known as Lucas – Kanade method [2].

III. LUCAS-KANADE METHOD

3.1. Overview

There are various methods available to solve optical flow but in this paper only Lucas – kanade is only mention. It is a local approach. To describe the optical flow constant model of small window Ω is used. Window function can be defined as W (m), m ∈ Ω. In window function weight is more at centre i.e., the window function favours the centre. Thus optical flow at higher weight can be calculated by

$$\min_V E = \sum_{m \in \Omega} W(m) \, (\nabla \mathrm{I}.\mathrm{v} + \frac{dI}{dx})^2$$

Writing out the derivatives, we have:

$$\frac{dE}{dVx} = \sum W^2(m)\left(\frac{dI}{dx}vx + \frac{dI}{dy}vy + \frac{dI}{dt}\right)\frac{dI}{dx} = 0$$

$$\frac{dE}{dVy} = \sum W^2(m)\left(\frac{dI}{dx}vx + \frac{dI}{dy}vy + \frac{dI}{dt}\right)\frac{dI}{dy} = 0$$

To solve it, we let:

$$\mathrm{A} = \begin{bmatrix} \frac{dI\,1}{dx\,1} & \frac{dI\,1}{dy\,1} \\ \vdots & \vdots \\ \frac{dIn}{dxn} & \frac{dIn}{dyn} \end{bmatrix}_{\mathrm{Nx}\,2}$$

$$\mathrm{W} = \mathrm{diag}\,(\mathrm{W}(\mathrm{m}_1), \ldots, \mathrm{W}(\mathrm{m}_N))_{NxN}$$

$$\mathrm{V} = \begin{bmatrix} \frac{dx}{dt} & \frac{dy}{dt} \end{bmatrix}^T = [Vx \quad Vy]^T$$

$$\mathrm{b} = -\begin{bmatrix} \frac{dI\,1}{dt} \\ \vdots \\ \frac{dIN}{dy} \end{bmatrix}_{NX1}$$

From notations,

$$\mathbf{A^T W^2 A v = A^T W^2 b}$$

Therefore, for the image pixel flow can be solved by [6],

$$\mathbf{V = (A^T W^2 A)^{-1} A^T W^2 b}$$

Here, the optical flow exclusively can be resolute for image pixel m if $\mathbf{A^T W^2 A}$ is not singular. The consistency of the evaluation of v is shown by the eigenvalues of $\mathbf{A^T W^2 A}$ (assume $\lambda_1 < \lambda_2$). The flow can be exclusively resolute If both eigenvalues are large; the normal flow can be resolute if λ_1 much larger than λ_2; if $\lambda_2 = 0$, the flow cannot be resolute [6].

Lucas-Kanade method is known as local method because it uses a local window to determine the flow [6].

3.2. Errors and Modifications

Error of large Window Size: If intensity constancy is not satisfied, motion is not small, a point does not move like its neighbours or window size is too large [9].

Modification: To overcome here we have implemented iterative Lukas-kanade algorithm [9].
Iterative LK is explained by following figures 3, 4 and 5.

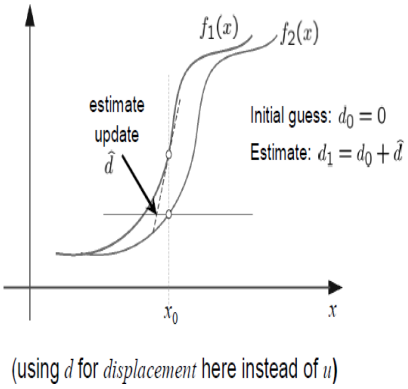


Figure 3.Step One

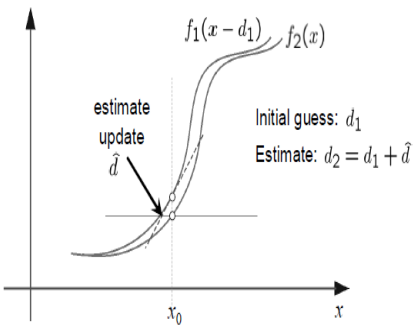
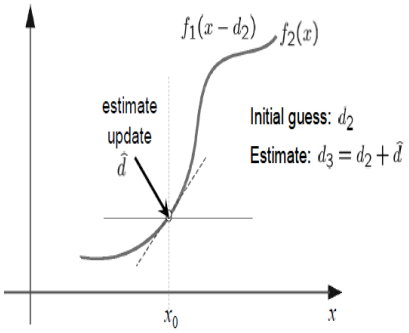


Figure 4. Step two



Figures 5.Step three

Error of Aliasing: In one frame, in given window there is a possibility of many pixels can have same brightness which cause temporal aliasing [9].

Aliasing error is explain by below figures 6 and 7

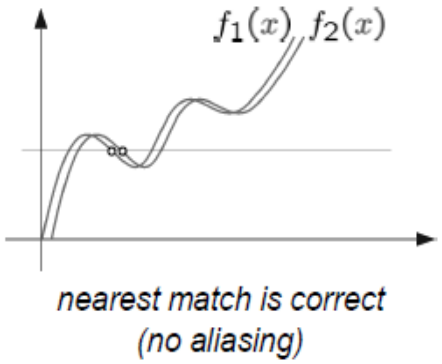


Figure 6. No aliasing

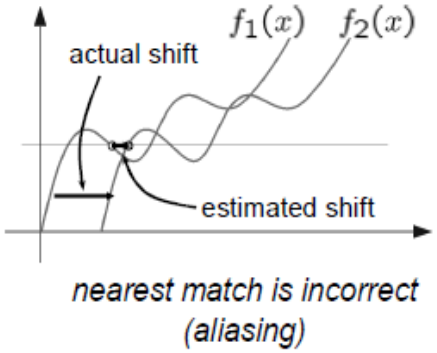


Figure 7. Aliasing effect

Modification: To overcome this error, Apply iterative lk. First apply lk algorithm and then reduce the size in half, repeat this algorithm for several times [9].

Explanation of iterative algorithm is given by figure 8.

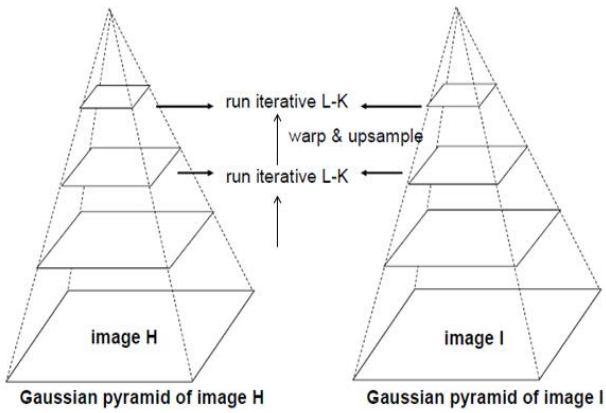


Figure 8. Iterative lk algorithm

Other errors:
If image motion is large (>5 pixels): go for Feature tracking methods, Extract blobs in image and track same blob in other frames.
If image motion is large (<5pixels):
From spatial-temporal image brightness variations using Optical flow-based methods directly determine image motion.
Note: Suitable for video and when image motion is small.

3.3. Pre-Processing

When any sensor develop for imaging and capture the image, but image and object quality are not same, because of system, system itself add some PSF into the resultant image and image get blur. To overcome this issue need to estimate the PSF and need to de-convolute it or de-blur the image in order to get more number of high frequency points.

To De-blur the image four techniques are used most commonly: (1) De-convolution Lucy-Richardson, (2) De-convolution regularized, (3) De-convolution Wiener, (4) De-convolution blind.

From these four techniques which de-convolution technique should be selected is decided by Standard parameter called "PSNR".

As PSNR increase quality of de-blur image improve. To justify the statement we have apply all four de- blur methods and see the improvement of image quilt and PSNR.

Capture Image:

Figure 9.Capture image

As we can see capture image is blurring.
PSNR of capture image is 28.75.

De-blur using Lucy – Richardson method

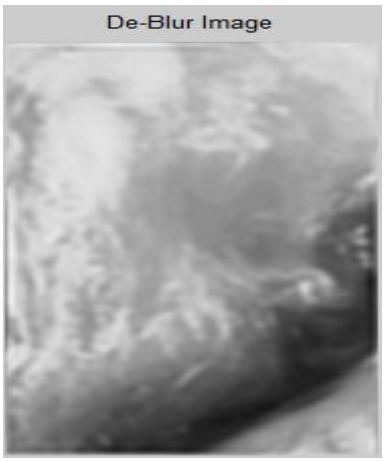


Figure 10. using Lucy- Richardson de-blur image

PSNR of de-blur image is 30.95.
PSNR improvement is 2.20 db.

De-blur using regularized filter method

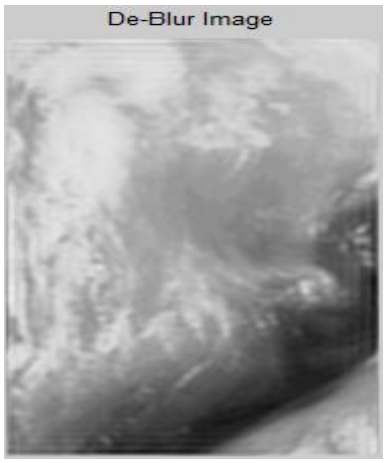


Figure 11. Using Regularized de-blur image

PSNR of de-blur image is 31.12.
PSNR improvement is 2.37 db.

De-blur using wiener filter method

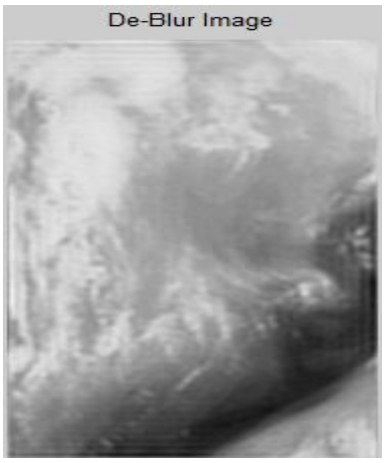


Figure 12. using wiener filter de-blur image

PSNR of de-blur image is 31.13.
PSNR improvement is 2.38 db.

De-blur using blind de-convolution method

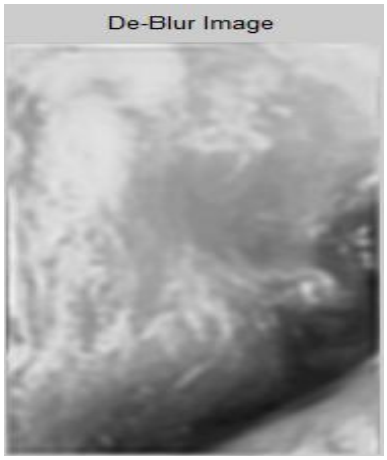


Figure 13. using blind de-convolution de-blur image

PSNR of de-blur image is 30.76.
PSNR improvement is 2.01 db.

Table.1 PSNR for different de-convolution method

	Blur PSNR	De-Blur PSNR	Improvement PSNR
De-convolution Lucy-Richardson	28.75	30.95	2.20
De-convolution regularized	28.75	31.12	2.37
De-convolution Wiener	28.75	31.13	**2.38**
De-convolution blind	28.75	30.76	2.01

For this specific experiment if choose anyone from regularize or wiener filter to de-blur the image will get best results as PSNR improvement is more in these method. Improvement of PSNR depends upon various parameter like Size of image, type of image, number of high frequency point in image. Depending upon these parameters improvement of PSNR is different in different de-blur techniques.

3.4. Target selection

There are two methods two show results:

Motion Field: compute optical flow for all points.
Velocity Vectors: give a set of corresponding points (high frequency points) in two images, displacement vector between these points compute by optical flow.
Here we have used motion field to show results of detected velocity.

IV. RESULTS

Here modified-lk algorithm is applied on various types of images to examine how good it detects the motion in image.

4.1 Experiment 1: To check accuracy by generating synthetic image

We have made two digital images with shift of 1 pixel.
Here detected the motion's direction is true as well as estimated pixel shift also near to original shift.
Estimated shift by algorithm 1.04 (mean)

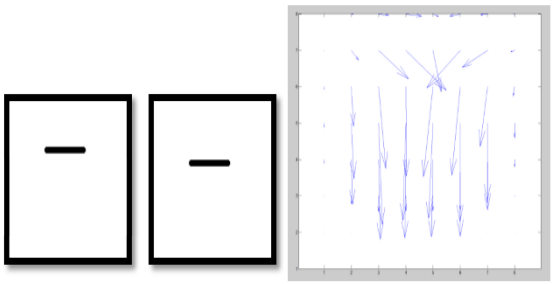


Figure14. Results of first experiment of optical flow

4.2 Experiment 2: Apply 2 pixel shifts and check results

Here we have manually give 1 pixel shift between frame 1 & frame 2 and 2 pixels shift between frame 1 and frame 3.
Estimated motion is true and even Estimated Velocity is nearly equal to original shift.
Shift between fr1 and fr2: 1.1644 (mean)
Shift between fr1 and fr3: 2.3751 (mean)

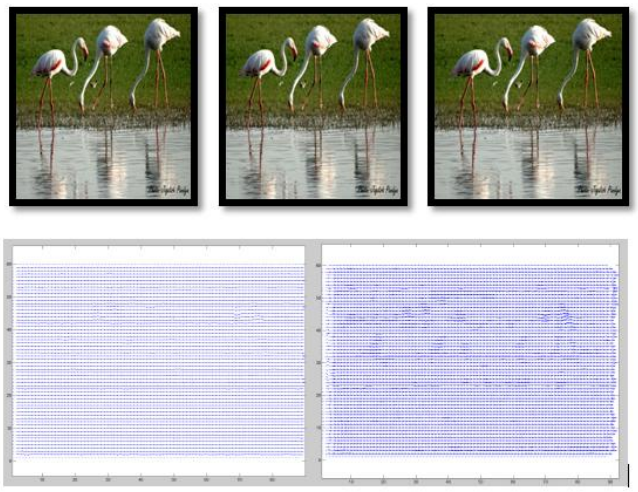


Figure 15. Results of second experiment of optical flow

4.3 Experiment – 3: Detect Cloud Motion

In this experiment algorithm is apply to detect cloud motion which our paper's primary objective.
Here we have take data of 21/4/2015 GOES (time difference between two frames is 30 min)
Here we have tried to archive two things:

1. Direction of clouds

We can see here that movement in this image is not in one direction

2. Velocity vector length

We can see here that movement in this image is not uniform some features are moving faster than other, from results we can see that implemented algorithm is able to detect different and show different velocity.

Velocity between two frames is unknown
Estimated shift (velocity component) =0.7282(mean)

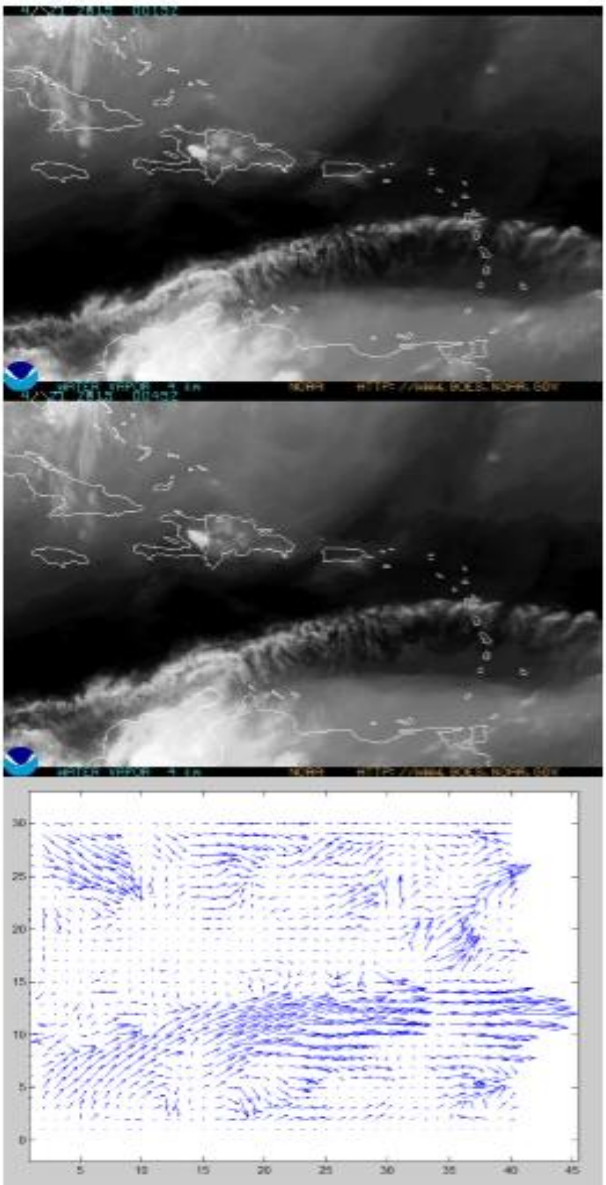


Figure 16. Results of third experiment of optical flow

4.4 Experiment 4 (Video analysis): To check real time reliability

Here algorithm is applied on video clip of cloud movement and get good results of that also. In order to get more high frequency point has done preprocessing also.

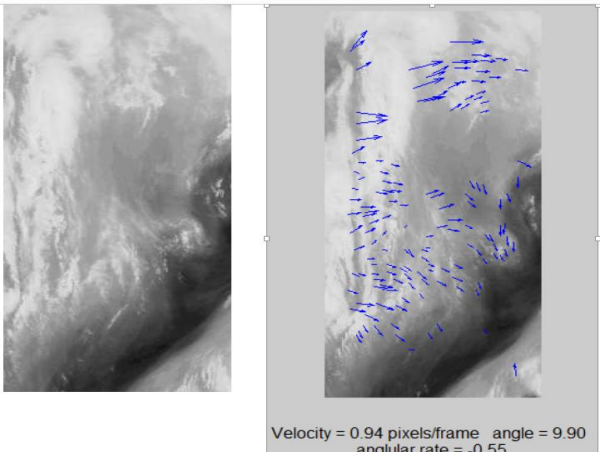


Figure 17. Cloud image and velocity estimation

V. CONCLUSION

Primary objective of this paper is to detect the motion of cloud. With this algorithm, in single frame we are able to detect different feature with different velocity. Further if features are moving in different direction then also algorithm is valid to detect motion in different direction. From the results we can conclude motion detection using this algorithm is having good accuracy and efficiency.

Future work can focus on estimate the velocity of cloud. Here we have detected the motion, compute the velocity in form of pixel/ frame. After several modifications velocity of cloud can be estimated in m/s form. From known velocity we can find the scaling factor and we can use that factor to estimate the velocity. But in order to estimate velocity we need to generate image model for various parameter such as height, angle etc. when we captures cloud image from different height (distance of satellite from earth) than scaling factor will be different. Angle of satellite to capture cloud image also play big part. So to estimate velocity we need to have a model which can compensate the effect of these parameters.

REFERENCES

[1] Wayne Bresky, Jaime Daniels, The feasibility of an optical flow algorithm for estimating atmospheric motion, 2007.
[2] Ying Wu, Optical Flow and Motion Analysis, Advanced Computer Vision Notes Series 6
[3] Barron J.L., Fleet. D.J., Beauchemin S.S., Performance of Optical Flow Techniques. Int. Journal of Computer Vision. 12: 43-77.1994
[4] Fraleigh, J.B., and Beauregard, R.A., Linear Algebra, Addison-Wesley. 1987.
[5] Horn, B.K.P., and Schunck, B.G., Determining Optical Flow, Artificial Intelligence 17: 185-204, 1981.
[6] Lucas, B.D., and Kanade, T., An Iterative Image Registration Technique with an Application to Stereo Vision, Proc 7th. Int. Joint Conf. Artificial Intelligence, 674- 679. 1981
[7] Simoncelli, E.P., Adelson, E.H., and Heeger, D.J., Probability Distributions of Optical Flow Proc. Conf. Comput. Vis. Patt. Recog.,310-315. 1991
[8] Trucco, E., and Verri, A.,. Introductory Techniques for 3-D Computer Vision. Prentice Hall. 1998
[9] 3.cs.huji.ac.il

