Vision Based Velocity Estimation for Soft Landing of Spacecrafts

Vishnu Gupthan V.R, Asha S, Dhanalakshmi P.P

Abstract—**Soft landing means to land a spacecraft on a surface by gradually reducing its velocity to within 0-5m/s. In order to land the spacecraft accurately at the desired landing spot the descent velocity of the spacecraft have to be determined. This paper introduces a vision based velocity estimation of a lander during soft-landing on an alien planet or a satellite. The method estimates the velocity by taking a series of images of the landing surface as the lander descents. Usingtwo consecutive pictures estimate how much vertical and horizontal motion is there which gives an estimate of the lateral velocity.First, the vertical motion is calculated and then it is cancelled in the second image to bring the two images to same scale. Then from the resulting two images the horizontal velocity is estimated.**

Keywords—***velocity estimation, optical flow, lateral velocity, soft landing***

I. Introduction

The future inter-planetary space missions are expected to have more landings on other planets than ever before. As man tries to expand his reach over the universe such missions will be common. For soft-landing on an alien planet or on a satellite accurate navigation systems are required. The velocity, attitude and altitude of the lander has to be determined accurately in order to ensure a safe landing. The conventional system of navigation uses radar altimeters, accelerometer and Inertial Navigation Systems (INS) or Inertial Measurement Units (IMU) to find the velocity, attitude and altitude of the lander. But the error produced by these systems increases as the altitude of the spacecraft decreases. A solution to this problem is to use a vision based velocity measurement system, which can produce less measurement error at the lower altitude of landing. In these type of systems, both the vertical and horizontal velocities are computed based on the images of landing surface captured using an on board camera. As the spacecraft descends for landing, both the vertical and horizontal velocities are simultaneously changing. So the challenge of such vision based systems is to accurately estimate the horizontal velocity from those images, which are at different scale and rotation angle

Vishnu Gupthan V.R:Department of Electronics and Communication SCT College of Engineering,Trivandrum, Kerala (vr.vishnugupthan@gmail.com)

Asha S:Asst. Professor, Department of Electronics and Communication SCT College of Engineering,Trivandrum, Kerala (ashasasi@gmail.com)

Dhanalakshmi P.P:Scientist/Engineer , IISU, ISRO,Trivandrum, Kerala (pp_dhanalakshmi@vssc.gov.in)

Vision based systems generally uses a CCD camera placed on board to take series of images of the landing surface while the lander descents. Then the vertical motion of the spacecraft is initially nullified using two consecutive images by bringing those images in to same scale and rotation angle. From the vertical motion compensated images, horizontal motion vectors are computed. Using horizontal motion vectors, along with the help of camera calibration, accurate horizontal velocity is estimated for safe landing.
This paper is organized as follows: Section 2 details about the existing methods of vision based velocity estimation for soft-landing of spacecraft. Section 3 explains the proposed algorithm in detail and the work done so far. Section 4 and5 discusses the experimental results and the future work respectively. Section 6 is the conclusion.

II. Related Work

The paper by Mourikis et al.used an Extended Kalman Filter estimator to combine data from a Inertial Measurement Unit and from image feature such as Marked Landmarks and Opportunistic Features to calculate the velocity of the lander [1]. The algorithm produced highly accurate results with errors of the order of 0.16m/s for velocity and 6.4m in position.

Another work on this field was from Johnson and Matthies. Their method was based on feature tracking between a pair of images. From the tracked features between the two images an estimate of the scaling and rotational motion are obtained [2]. Then using laser altimetry the translational motion is obtained. The algorithm gives good results on attitude, position and motion covariance between the two pair of images.

The Descent Image Motion Estimation System (DIMES) was the first system that was used to control the descent of a lander [3]. It combines information obtained from camera with altimeter readings and data from inertial measurement unit to accurately predict the descent velocity within an accuracy of 5m/s. It was used in the Spirit and Opportunity Missions to Mars.

Ming Jie, Xianlin Huang, Hang Yin and Hongqian Lu developed an algorithm to obtain the 6DoF from descent images [4]. They used dual-quaternion to find out the absolute 3D coordinates of the features detected in the images. Using which estimate of pose of the lander can be found. An Unscented Kalman Filter was used to estimate the 6DoF motion parameters.

978-1-5090-0774-5/16/$31.00 © 2016 IEEE

Another interesting work was done by Florent Valette, Franck Ruffier, StéphaneViollet, and Tobias Seidl who used neuromorphicprincples to process optical flow obtained [5]. They tested their algorithm on a simulated Lunar environment to determine the velocity of the lander at low gate within an accuracy of 10-5m/s. Data from Inertial Measurement Unit was also used to estimate the velocity of the lander.

All the earlier developed methods for lateral velocity estimation use measurements from other hardware systems to draw an estimate of horizontal velocity. Here, we propose a novel vision aided navigation system which can estimate lateral velocity independently based on the images captured by an on board camera. The advantage of our system is that, it produces very less measurement error at lower altitudes of landing phase compared to the conventional navigation systems and it doesn't depends on the measurements from other hardware systems.

III. ALGORITHM

The algorithm has the following steps:

- Step 1: Take series of images of the landing surface using a CCD camera placed on board at fixed interval of time.
- Step 2 Compute the matching features between two consecutive images usingSpeed-Up Robust Features (SURF) [6] algorithm.
- Step 3: Bring both images into same scale and orientation using Random Sample Consensus (RANSAC) [7] algorithm. Thus the vertical motion of spacecraft is compensated.
- Step 4: Compute the horizontal motion vectors using Lucas-Kanade with Pyramids algorithm.
- Step 5: Estimate the horizontal velocity by means of camera calibration and horizontal motion vectors.

Fig. 1 shows a simplified block diagram of the proposed method. The vertical and the rotational motion components are removed by using SURF and RANSAC algorithms. SURF is used to match the corresponding features in consecutive images. RANSAC algorithm extracts the amount of scaling and rotation using the result of SURF, and this information is used to bring both the images to same scale and orientation. After the vertical and rotational motion component has been removed the remaining, the two remaining images will have only horizontal

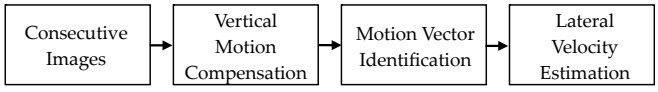


Fig. 1: Simple block diagram of proposed method

motion component. From these two images using the Lucas-Kanadewith Pyramids Optical Flow algorithm, motion vectors are determined. Simple Lucas-Kanade algorithm was also implemented and its performance was compared with the Lucas-Kanade with Pyramids Optical Flow algorithm. Lucas-Kanade with Pyramids is incorporated in our method as this algorithm outperforms the simple Lucas-Kanade in terms of angular error produced. Finally the horizontal velocity is estimated using camera calibration.

A. Speed-Up Robust Features

The SURF algorithm involves detection of features in the two images and then matching the detected features between the two images. Detection and matching of features between two images involves the following steps: First, the Interest Points are found. Second, Descriptors for the interest points are determined. Then the descriptors are matched between two images to find the matching interest points. Interest points maybe blobs, corners or T-junctions. For detecting Interest Points, Hessian Matrix of the image at a point $\mathbf{X} = (\mathbf{x}, \mathbf{y})$ is found first. Hessian Matrix is a square matrix of second order partial derivatives of a scalar valued function. It describes the local curvature of a function of many variables. Let $\mathbf{f}: \mathbf{R^n} \longrightarrow \mathbf{R}$, then the Hessian Matrix of $\boldsymbol{f}$ is

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^{\,2}} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^{\,2}} & & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^{\,2}} \end{bmatrix} \quad (1)$$

The Hessian matrix used in this algorithm is as given below

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (2)$$

where$L_{xx}(X, \sigma)$ is the convolution of the image at point X with second order partial derivative of the Gaussian function with standard deviationX. The suffix xx denotes that the convolution is done with Gaussian function or mask in the x direction. While yy denotes the Gaussian function or mask in the y direction and suffix xy denotes the Gaussian function in the xy direction. The determinant of the Hessian Matrix at every point is found. The maxima of the determinants of Hessian matrices at different locations are estimated using non-maxima suppression method. The maximum values of the Hessian matrix are takenas blob response of the image. Theseblobresponses are stored in a response map. This represents the Interest Points of the image. By varying the value of σthe Hessian Matrix at different scales are also found. Then the maximum values of the Hessian Matrix at different scales are interpolated to form the overall blob response.Once the Interest Points have been detected next the Descriptors for each of the Interest point is found. The Descriptor is basically a vector, which gives the orientation, or direction, of the Interest Point. Descriptors are found using Haar Wavelet response of the image at each point. The Haar Wavelet responses are found in the x and y direction and are stored as vector with x and y components. Then for matching the Descriptors of each Interest Point the orientation of each descriptor is used, because the

orientation will be unique for each Descriptor. To find the orientation describe a circular region around an interest point and then summing the Haar Wavelet responses along a sliding window $\pi/3$ radians wide. The summed values are stored and compared with each other. Then the orientation of the largest of the sum is choosen as the orientation of the Descriptor.

B. RANSAC

RANSAC is a model estimation algorithm. To give an idea about what RANSAC does, consider a plane containing a line that serves as the model. There are points lying randomly, some of which are close to the line, called inliers, and some very far from the line, outliers. The algorithm takes two points at a time and computes the equation of the line going through it. The distance of the points lying on this line from the model line is calculated. This procedure is repeated and the line that has maximum number of points, which give the minimum distance, is selected as the model line.

C. Lucas-Kanade Optical Flow Algorithm

Optical Flow is a measure of how much a pixel has moved from one frame or image to another frame. The amount ofmotion for each pixel is represented by Motion Vectors. The goal of Optical Flow Estimation is to compute an approximation to the motion field from time varying image intensity. Let the intensity of a pixel with coordinates $(\mathbf{x}, \mathbf{y})$ at a particular time$\boldsymbol{t}$be denoted by $\mathbf{f}$ $(\mathbf{x}, \mathbf{y}, \mathbf{t})$. The brightness of the image taken at time intervals $\mathbf{dt}$ is assumed to be a constant, this is called brightness constancy assumption. Therefore it can be written,

$$f(x, y, t) = f(x + dx, y + dy, t + dt) \tag{3}$$

taking Taylor series on both sides, and then dividing each side bydt, we get

$$f_x u + f_y v + f_t = 0 \tag{4}$$

where $u = dx/dt$, $v = dy/dt$, $f_x = \partial f/\partial x$, $f_y = \partial f/\partial y$ and $f_t = \partial f/\partial t$. This equation is called the Optical Flow equation [8]. Here the variables u and v constitutes the motion vector. This equation has two unknowns. So that it can't be solved as such. Lucas-Kanade algorithm assumes that the optical flow in the 8-neighbourhood of the central pixel is same; this is called as smoothness assumption. Which means that every pixel in the 3×3 neighbourhood gives the same equation. Therefore we can write

$$\begin{bmatrix} f_{x1} & f_{y1} \\ f_{x2} & f_{y2} \\ \vdots & \vdots \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ -f_{t2} \\ \vdots \\ -f_{t9} \end{bmatrix} \tag{5}$$

where(f_x, f_y) will be different. Thus there are nine equations and two unknowns, which gives a over constrained system. Thus it's solvable and motion vectors can be computed using equation 6. In short form

$$\bar{u} = (\bar{A}^T \bar{A})^{-1} \bar{A}^T \bar{f}_t \tag{6}$$

where$\bar{A} = \begin{bmatrix} f_{x1} & f_{y1} \\ f_{x2} & f_{y2} \\ \vdots & \vdots \\ f_{x9} & f_{y9} \end{bmatrix}$, $\bar{u} = \begin{bmatrix} u \\ v \end{bmatrix}$and $\bar{f}_t = \begin{bmatrix} -f_{t1} \\ -f_{t2} \\ \vdots \\ -f_{t9} \end{bmatrix}$

The disadvantage of this method is that it fails for areas where there is large motion, because the algorithm is depending on derivatives and derivatives are small 2×2 masks. But if the motion is, say, fifteen pixels wide, then 2×2 masks will not be able to make a distinction between two pixels and fifteen pixels. This is why simple Lucas-Kanade fails in case of image sequences with large motion. To overcome this problem the Lucas-Kanade with Pyramids algorithm is used.

D. Lucas-Kanade with Pyramids

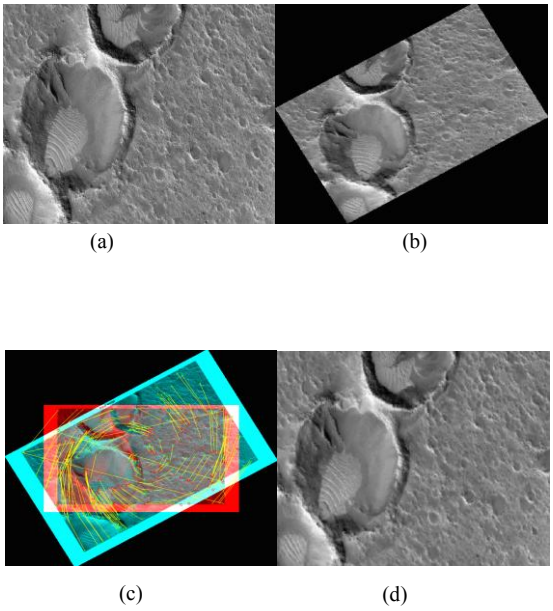
Pyramids can be used to compute large Motion Vectors. In this algorithm Gaussian Pyramids are used. Gaussian Pyramid is a technique used in which a series of images are created which are weighted down using a Gaussian average and scaled down. This creates a stack of successively smaller images of diminishing size and resolution. Each level in the pyramid is $1/4^{\text{th}}$ the size of the previous level. Lowest level in the pyramid has the highest resolution and the highest level has the lowest resolution.

The Lucas-Kanade with Pyramids algorithm has the following steps:

- Step 1: Construct the image pyramid
- Step 2: Compute the "simple" Lucas-KanadeOptial Flow at the highest level. Assuming the motion is small at the highest level in the pyramid.
- Step 3: At level i, take the optical flow from the previous level. Let (u_{i-1}, v_{i-1}) be the optical from the $i-1$
- Step 4: Multiply (u_{i-1}, v_{i-1}) by two. Because in the higher level if the flow is x, then in the lower level it should be $2x$, since the resolution is increased. Since there is twice as many pixels in the lower level, itsnot possible to know the value of all pixels (which does not map directly from the higher level to lower level). For this reason bilinear interpolation is used to create (u_i^*, v_i^*)
- Step 5: Multiply (u_i^*, v_i^*) by two.
- Step 6: Apply the time derivative mask at a position displaced by (u_i^*, v_i^*). Then find the motion vectors (u_i', v_i') using this derivative, which is the correction in the flow.
- Step 7: Add corrections (u_i', v_i') that is

$$u = u_i^* + u_i', v = v_i^* + v_i' \quad (7)$$

Thus the residual flow is added to the double of the one computed at the higher level to get actual optical flow.



(a) (b)

(c) (d)

Fig. 2: (a) Input Image (b)Rotated and Scaled Version of Input Image (c) Interest points matched between an input image and the scaled and rotated version of itself. Red circles shows the interest points in the original image and the yellow cross shows the corresponding matched points in the rotated image and (d) image reconstructed using RANSAC algorithm

TABLE I. ANGULAR ERROR FOR SIMPLE LUCAS-KANADE ANDLUCAS-KANADE WITH PYRAMIDS

Image Sequence	Angular Error (in degrees)	
	Simple Lucas-Kanade	***Lucas-Kanade with Pyramids***
Groove3	66.73	60.28
Urban2	51.89	44.30
Rubberwhale	47.40	42.31
Dimetrodon	52.48	43.50
Hydrangea	73.21	70.03

IV. EXPERIMENTAL RESULTS

The simple Lucas-Kanade, Lucas-Kanade with Pyramids, SURF and RANSAC algorithms were implemented in MATALB. First a sample image was taken and then rotated and scaled, refer Fig. 3. It was then given as input the SURF algorithm. The detected and matched features in the scaled and original image are shown in Fig. 2. Then RANSAC algorithm was applied to these images to recover the scale and angle of rotation. Using this scale and angle the two images are brought to the same scale and orientation. By doing this the vertical motion between a pair of images are cancelled.

The Lucas-Kanade with pyramids algorithm was compared with the simple Lucas-Kanade algorithm.The comparison was done on basis of the angular error (AE) [9] of the two algorithms for a given set of images. The reference images with true optical flows were downloaded from theMiddlebury dataset [10]. Fig. 4 shows the optical flow obtained for some image sequences taken from the dataset. From Table 1 we can see that Lucas-Kanade with pyramids algorithm gives much more accurate optical flow vectors.

V. FUTURE WORK

The future works involve estimation of lateral velocity using the computed motion vectors and camera calibration. Mounting a CCD camera on a robot, which can be moved at desired direction and speed, can be used to do camera calibration. The accuracy of the proposed method can also be verified using this camera.

VI. CONCLUSION

The presented method for estimating the horizontal velocity of a spacecraft during landing phase is very unique because it uses only information from images taken by a camera. It works completely independent of other systems. Thus it can be used to measure or verify the accuracy of other parallel velocity estimation systems. The aim is to measure the velocity with high level of accuracy, which is necessary to perform soft landing. The method is also hardware efficient since it uses only a CCD camera and a processing circuit to obtain the velocity. Ultimately it's intended to reduce the error in velocity developed by the Inertial Navigation System as it approaches the final descent.

ACKNOWLEDGMENT

I would first like to thank my thesis advisor Mrs. Asha S., Assistant Professor, at the SCT College of Engineering, Trivandrum for all her support and guidance through out the this thesis work. I'm also deeply indebted to Ms. Dhanalakshmi P.P., Scientist/Engineer at IISU, ISRO. The door to Ms.Dhanalakshmi'soffice was always open whenever I ran into a trouble spot or had a question about my research or writing.

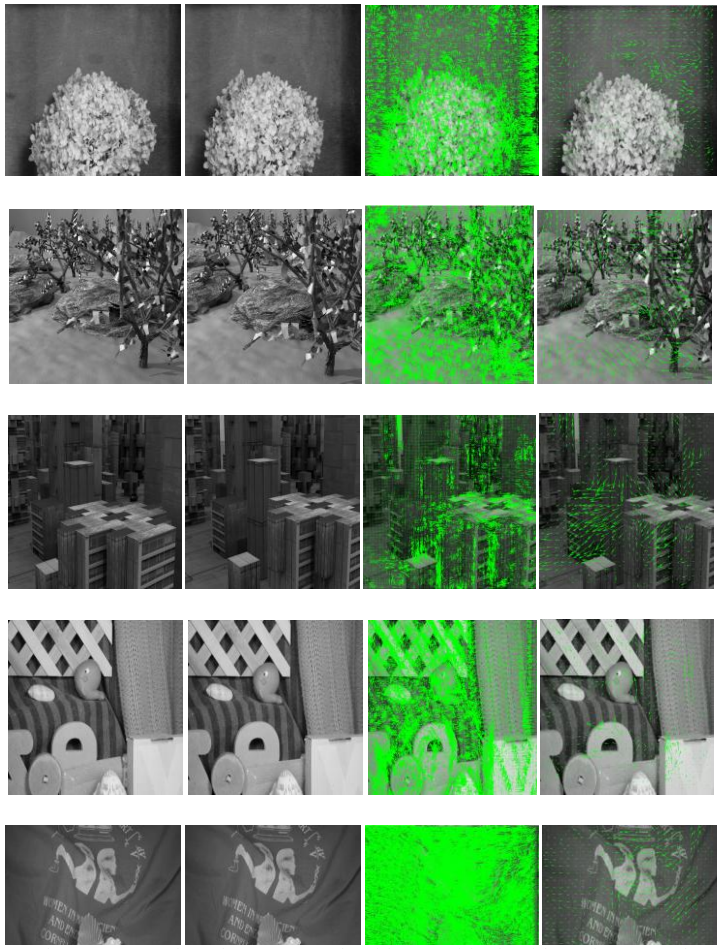


Fig. 3: Results for SImple Lucas-Kanade and Lucas-Kanade with pyramids algorithm. First two column shows the image sequence. The second column is the optical flow obtained using Simple Lucas-Kanade. Third column is the result obtained using Lucas-Kanade with pyramids. (a) Hydrangea sequence, (b) Groove 3 sequence, (c) Urban 2 sequence, (d) Rubberwhale sequence and (e) Dimetrodon sequence

I would like to thank Mr. Vinoj V.S., Scientist/Engineer, of IISU, for his very valuable comments on this thesis work. I would also like to thank Prof. (Dr.) Sheeja M.K., HOD Department of Electronics and Communication, SCT College of Engineering, for her support and guidance.

REFERENCES

[1] Anastasios I. Mourikis, Nikolas Trawny , Stergios I. Roumeliotis, Andrew E. Johnson, Adnan Ansar, and Larry Matthies, "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing", IEEE Transactions on Robotics, vol. 25(2), pp. 264-280, April 2009.

[2] Andrew E. Johnson and Larry H. Matthies, "Precise Image-Based Motion Estimation for Autonomous Small Body Exploration", 5th Int'l Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS'99), pp. 627-634.

[3] Andrew Johnson, Reg Willson, Yang Cheng, Jay Goguen, Chris Leger, Miguel Sanmartin And Larry Matthies, "Design Through Operation ofan Image-Based Velocity Estimation System for Mars Landing", International Journal of Computer Vision, vol. 74(3), pp. 319–341, January 2007.

[4] Ming Jie, Xianlin Huang, Hang Yin and Hongqian Lu, "A Precise Vision-Based Navigation Method for `Autonomous Soft Landing of Lunar Explorer", Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, pp. 1138-1142, December 2007.

[5] Florent Valette, Franck Ruffier, Stéphane Viollet, and Tobias Seidl, "Biomimetic optic flow sensing applied to a lunar landing scenario", 2010 IEEE International Conference on Robotics and Automation, pp. 2253-2260, May 2010.

[6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-Up Robust Features (SURF)", Computer Vision and Image Understanding, vol. 110(3), pp. 346-359, June 2008.

[7] Martin A. Fischler and Robert C. Bolles , "Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography", Communications of ACM, vol. 24(6), June 1981.

[8] B. Horn and B. Schunck, "Determining optical flow". Artificial Intelligence, vol. 17, pp.185–203, 1981.

[9] J. Barron, D. Fleet, and S. Beauchemin. "Performance of optical flow techniques**",** Inter- national Journal of Computer Vision, vol. 12(1), pp. 43–77, 1994.

[10] Simon baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael Black, and Richard Szeliski, "A Database and Evaluation Methodology for Optical flow" International Journal for Computer Vision, vol. 92(1), pp. 1-31, March 2011.