International Journal of Computer Applications Technology and Research
Volume 2– Issue 1, 32-36, 2013, ISSN: 2319–8656

# Speed Determination of Moving Vehicles using Lucas-Kanade Algorithm

Dolley Shukla
Shri Shankaracharya College of Engg. & Tech.
Junwani, Bhilai-490020
Durg,(CG), India

Ekta Patel
Shri Shankaracharya College of Engg. & Tech.
Junwani, Bhilai- 490020
Durg,(CG), India

**Abstract:** This paper presents a novel velocity estimation method for ground vehicles. The task here is to automatically estimate vehicle speed from video sequences acquired with a fixed mounted camera. The vehicle motion is detected and tracked along the frames using Lucas-Kanade algorithm. The distance traveled by the vehicle is calculated using the movement of the centroid over the frames and the speed of the vehicle is estimated. The average speed of cars is determined from various frames. The application is developed using MATLAB and SIMULINK.

**Keywords**: Tracking, Optical flow, Motion estimation, Lucas-Kanade algorithm,velocity

## 1. INTRODUCTION

### 1.1 Video and Image Sequence:

**Video** is the technology of electronically capturing, recording, processing, storing, transmitting, and reconstructing a sequence of still images representing scenes in motion.[1] An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels.

A video usually consists of scenes, and each scene includes one or more shots. A shot is an uninterrupted segment of video frame sequence with static or continuous camera motion, while a scene is a series of consecutive shots that are coherent from the narrative point of view.

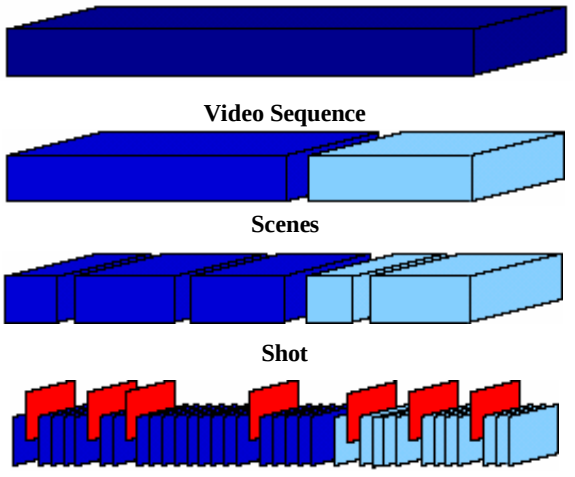


**Fig1: Hierarchical structure of Video Sequence**

### 1.2 Motion estimation:

Motion estimation is an important task of video analysis. [1] It can be used to find the motion fields, to identify moving objects, calculate object trajectory and to find their velocity.

In this paper, we present an efficient method for computing direction of motion of a vehicle and estimate its speed.[1] The proposed method firstly uses optical flow algorithm to calculate changes in the intensity of the pixels of the images. These apparent velocity components are then subjected to image processing techniques to obtain centroid of the vehicle across the frames. The image coordinates of the centroid are mapped to World space. Using this information the velocity of the vehicle is estimated.

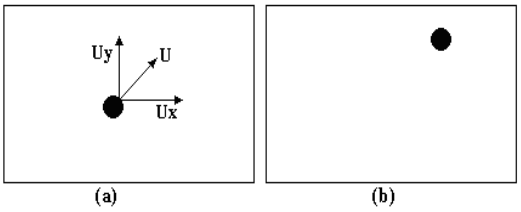


**Fig2: Motion vector definition of current frame a) at time t-1 b) at time t**

### 1.3 Methods of Motion Estimation:

***1.3.1 Feature/Region Matching:*** Motion is estimated by correlating or matching features (e.g. edges) or regional intensities (e.g. blocks of pixels) from one frame to another. [1]

Examples include:

- Block-matching algorithm
- Phase correlation and frequency domain methods

***1.3.2 Gradient based methods*:** Gradient-based methods use spatial and temporal partial derivatives to estimate image flow at every position in the image.

• Optical flow
• Pixel recursive algorithms

## 2. LITERATURE REVIEW

Mehrubeoglu and McLauchlan[2] detect and count vehicles during day and night scenarios and different environmental conditions in 2006. This paper is an extension of the authors' work from detecting vehicles in still images to tracking vehicles in video.

In their paper, An *et al.* report a motion tracking algorithm that tracks different objects in motion in video.[3] The authors achieve motion tracking by segmenting key frames, and then clustering objects whose motion is close to the previously detected objects. Classification is achieved with a distance measure based on epicenter geometry.

Yu *et al.* describe an algorithm that estimates traffic density and average speed from Skycam MPEG compressed images.[4] The authors compute DCT coefficients and analyze motion vector projections across frames. Direction, magnitude and texture filters are used to eliminate redundant motion vectors. After mapping the image plane to world coordinates, the average vehicle speed is estimated over a video clip of 10 seconds, with a frame rate of 10 fps.

Anagnostopoulos *et al.* surveyed license plate recognition methods.[5] The authors broke license plate recognition into three parts, 1) license plate location, 2) license plate segmentation, and 3) character recognition. In addition, the authors have devised a database of license plate images and videos under varying lighting and environmental conditions that researchers may utilize as a common test set to enable comparisons of various algorithms.

Garibotto *et al.* utilize computer vision techniques to estimate a vehicle's speed from two images by tracking license plates and determining the distance traveled.[6] The speed is calculated by using this traveled distance and the time difference between the two images. The researchers describe both monocular as well as binocular vision systems. In our method, license plate information is not needed.

Pelegrí *et al.* developed and tested GMR magnetic sensors to determine car speeds.[7] The vehicle causes changes in the magnetic field of the sensor when it travels over the sensor. Their tracking technique did not use cameras, but is important to show the diversity of technology research for tracking and vehicle speed information.

Li *et al.* determined vehicle speeds by utilizing a CCD camera and looking at the vehicle positions in video frames.[8] The speed is determined geometrically by the two vehicle positions and their spatial relationship to the known fixed CCD camera position. In our work, the spatial relationship of a tracked vehicle across frames is determined through transformation of pixel locations from image to world coordinate system. Transformation to world coordinates and pixel calibration are achieved by using standard lane markings whose length and gap distance are standard and known.

He *et al.* developed an embedded system to take traffic measurements [9]. The authors used background subtraction to aid in vehicle detection. The researchers then used parallelograms for the regions of interest (ROI) due to the image distortion resulting from the camera position and to reduce the computational load.

## 3. METHODOLOGY

### 3.1 Implementation in Simulink:

The Simulink model for this project mainly consists of three parts, which are "Velocity Estimation", "Velocity Threshold Calculation" and "Object Boundary Box Determination".

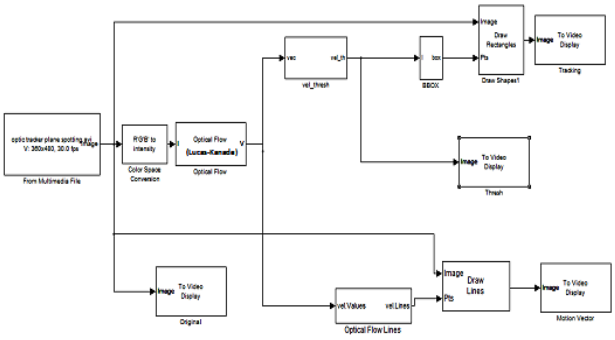

**Fig3: Simulink Block Diagram for Tracking Moving Objects Using Lucas-Kanade Algorithm**

### 3.2 Lucas-Kanade Algorithm:

The Lucas–Kanade method is a two-frame differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade.

It introduces an additional term to the optical flow by assuming the flow to be constant in a local neighbourhood around the central pixel under consideration at any given time.[10]

The additional constraint needed for the estimation of the flow field is introduced in this method by assuming that the flow ($V x$, $V y$) is constant in a small window of size m X m with m > 1, which is centered at Pixel x, y and numbering the pixels within as 1...n, n = m2, a set of equations can be found:

$$I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1)$$

$$I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2)$$

.

.

$$I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n)$$

where $q_1, q_2, \ldots, q_n$ are the pixels inside the window, and $I_x(q_i), I_y(q_i), I_t(q_i)$ are the partial derivatives of the image $I$ with respect to position *x*, *y* and time *t*, evaluated at the point $q_i$ and at the current time.

These equations can be written in matrix form $Av = b$, where

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

The Lucas-Kanade method obtains a compromise solution by the least squares principle. Namely, it solves the 2×2 system

$A^T Av = A^T b$ or

$\mathrm{v} = (A^T A)^{-1} A^T b$ where $A^T$ is the transpose of matrix $A$. That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_x(q_i) I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(q_i) \end{bmatrix}$$

with the sums running from *i*=1 to *n*.

## 3.3 Steps Followed:

1. Input Video stream is captured.
2. Convert AVI file from RGB to Intensity.
3. The Intensity is then sent to Optical Flow block and velocity vectors in the form of matrix are obtained.
4. The matrix is then sent to 'Threshold and Region Filtering Block'.
5. Inside the block there is a Velocity Threshold block which calculates the mean threshold velocity and gives a binary threshold image.
6. This Threshold image is then divided into 2 halves using Submatrix block and processed individually. The video sequence is sent to Blob Analysis block. This block calculates statistics for labeled regions in a binary image. These labeled regions are known as blob.
7. The Blob Analysis block returns region of motion and the coordinates of centroid of the moving objects in the video sequence.
8. The Threshold Image (in Binary form), coordinates of Bounding box and centroid are sent to 'Display Results' block.
9. Centroid is superimposed on the original video using Draw Markers block.
10. The video obtained in step (9) is subtracted from original (divided) video to obtain image with centroid only.
11. The individual halves are stored in different AVI files.
12. The halves are concatenated to obtain the video sequence with superimposed centroid.
13. The video is then read and converted to frames.
14. Each image is converted from gray level to binary.
15. Reading each frame, the 2D coordinates of centroid for car are extracted and stored in a matrix with corresponding frame number.
16. The structure of the matrix is such that:

| Frame Number | x - coordinate | y - coordinate |
|---|---|---|
| | | |

17. The calibration parameters for the fixed camera (calculated earlier) are used to convert 2D coordinates to 3D coordinates. That is, 2D Image coordinates are converted to 3D World coordinates using calibration parameters.[5]
    A new matrix is obtained:

| Frame Number | X -coordinate | Y- coordinate | Z-coordinate |
|---|---|---|---|
| | | | |

The Euclidean distance between each successive matrix element is calculated. Distance between 2 centroid P (xi,,yi, zi) and Q(xj,,yj, zj) in world space is calculated as

**distij = √ (xi-xj)2 +(yi-yj)2+(zi-zj)2**

Where n= total number of frames captured, i= 1 to n-1 and j=i+1 to n

18. The total distance traveled (in millimeters) by the object across the images is calculated. Total number of world coordinates traveled (in mm) is given by:

    **n-1 , n**
    **D = Σ distij**
    **i=1, j=i+1**

19. This distance is converted to actual distance traveled (in centimeters) using pixel to distance ratio (ctod) which is calculated earlier (from preprocessing). The total distance traveled by the vehicle under consideration is calculated as:

    **dtraveled = D * ctod**

    Where dtraveled (in centimeters) is the total distance traveled by the vehicle.

20. The time of the motion (in seconds) is calculated using the following relation:

    **ttraveled = n / fps**

    Where n = number of frames for which motion of car was studied,

    fps = frame rate of the AVI (number of frames per second). This is obtained from the AVI information.

21. Estimated velocity of the vehicle in centimeters per second is calculated as:

    **Vel (cm/s) = dtraveled (cm) / ttraveled (s)** [11]

**Fig 4.1: Original Video**

**Fig 4.2: Motion Vector**

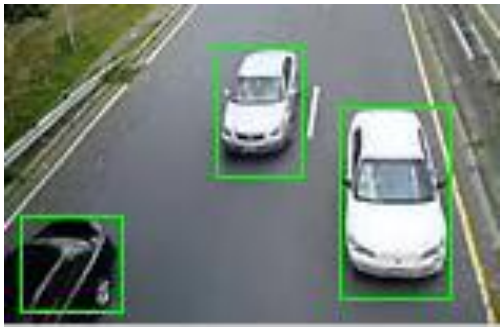

**Fig 4.3: Tracking the vehicle**

**Fig 4.4: Thresholded Video**

## 4. RESULTS

All the coding and matrix representations have been implemented in MATLAB. The proposed method was subjected to various experiments in order to check its accuracy and feasibility.

As a process of initialization, camera is calibrated and coordinate to distance ratio is calculated. Further, the video sequences were captured using a digital camera with 15 fps sample rate and resizing the images to 120 X160 pixel resolution.

The proposed method was first implemented for one vehicle.[12] A camera was mounted on a height and the video was captured. The calibration parameters were calculated using Calibration toolbox. The pixel to distance ratio was calculated and was stored for further analysis.

The experiment was repeated for multiple videos and 8 cars and the result is tabulated. Below are the output results in the form of Speeds and motion vectors of cars.[9]

**Table 4.1: Average Speed of Cars**

| cars | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Speed(km/h) | 1.22 | 1.25 | 1.28 | 1.31 |

| cars | 5 | 6 | 7 | 8 |
|---|---|---|---|---|
| Speed(km/h) | 1.27 | 1.30 | 1.35 | 1.32 |

**Table 4.2 : Motion Vectors of Cars**

| cars | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Motion vector | 0.3506 | 0.3548 | 0.3537 | 0.3586 |

| cars | 5 | 6 | 7 | 8 |
|---|---|---|---|---|
| Motion vector | 0.3671 | 0.3672 | 0.3753 | 0.3798 |

## 5. CONCLUSIONS & FUTURE WORK

The objective has been to detect moving objects and thereafter, calculate the speed of moving Vehicles and motion vector.[12] While earlier we worked with object-intrinsic properties such as the centroid of a moving object in order to make a probable prediction of its immediate future motion, methods to detect a rectangular boundary for the object, then used background subtraction Simulink models and but didn't get fair output for multiple vehicles. Further we made an attempt using the Lucas-Kanade method. [13] Although that it does not yield a very high density of flow vectors, Lucas-Kanade Algorithm is robust in presence of Noise. Vehicle trajectory is shown in fig.5. It is window based local method. Average angular error is less in Lucas-Kanade algorithm. There exist fast and accurate optical flow algorithms which can be applied in future. Hence, in future local and global methods can be combined for requirement of dense flow estimate, preserve discontinuities and to make it robust to noise. .[14]

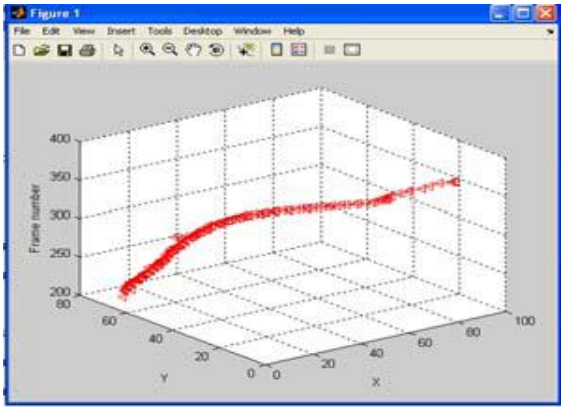


**Fig 5: Vehicle Trajectory using Lucas-Kanade algorithm**

## 6 REFERENCES

1. Y. Wang, J.Ostermann, Y.O.Zhang, E.F. (1995), "Video Processing and Communications", Prentice Hall.

2. Mehrubeoglu, M. and McLauchlan, L., E.F. ( 2006 ), "Determination of traffic intensity from camera images using image processing and pattern recognition techniques", Proc. SPIE-IS&T Electronic Imaging, SPIE Vol. 6063, 60630Q-1 -60630Q-12.

3. An, X., Qin, X. and Bao, H., E.F.( 2006 ), "Automatic and robust classification of independent motions in video sequences", Proc. SPIE-IS&T Electronic Imaging, Vision Geometry XIV, SPIE 6066, 60660B-1 – 60660B-8.

4. Yu, X.-D., Duan, L.-Y., and Tian, Q., E.F.( 2002 ), "Highway traffic information extraction from skycam MPEG Video", Proc.IEEE 5th International Conf. on Intelligent Transportation Sys., pp.37-42.

5. Anagnostopoulos, C.-N. E., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V., and Kayafas, E., E.F.( 2008 ), "License Plate Recognition From Still Images and Video Sequences: A Survey", IEEE Trans. Intelligent Transportation Systems,9(3), pp.377-391

6. Garibotto, G., Castello, P., Del Ninno, E., Pedrazzi, P, and Zan, G., E.F.( 2001 ), "Speed-vision: speed measurement by license plate reading and tracking", Proc. IEEE Intelligent Transportation Sys. Conf. , pp.585-590.

7. Pelegrí, J., Alberola, J., and Llario, V., E.F.( 2002 ), "Vehicle detection and car speed monitoring systems using GMR magnetic sensors", Proc. IEEE 2002 28th Annual Conf. Industrial Electronics Society, pp.1693-1695.

8. Li, Y., Yin, L., Jia, Y., and Wang, M., E.F.( 2008 ), "Vehicle speed measurement based on video images", Proc. 3rd International Conf. Innovative Computing Information and Control, pp.439-442.

9. He, Z., Liu, Y., Yu, H., and Ye, X., E.F. ( 2008 ), "Optimized algorithms for traffic information collecting in an embedded system", Proc. Congress on Image and Signal Processing, Vol. 4, pp. 220-223.

10. S. Baker, I. Matthews, E.F.( March 2004 ), "Lucas-Kanade 20 Years On: A Unifying Framework", IJCV, Vol.56, No. 3, pp. 221-255.

11. Lazaros Grammatikopoulos, George Karras, Elli Petsa, E.F.(November 2005), "Automatic Estimation of Vehicle Speed from Uncalibrated Video Sequences", International Symposium on Modern Technologies, Education and Professional Practice in Geodesy and related fields, Sofia,pp.03 – 04.

12. Savan Chhaniyara, Pished Bunnun, Lakmal D. Seneviratne and Kaspar Althoefer, E.F.( MARCH 2008), "Optical Flow Algorithm for Velocity Estimation of Ground Vehicles: A Feasibility Study", International Journal on smart sensing and intelligent systems, VOL. 1, PP. 1.

13. J.L. Barron, D.J. Fleet, S.S.Beauchemin, T.A. Burkitt, E.F. (1992), "Performance of Optical Flow Techniques", Computer Society Conference on Computer Vision and Pattern Recognition, pp. 236-242.

14. A. M Tekalp, E.F(1995), "Digital Video Processing Englewood Cliffs", NJ: Prentice-Hall.