Vehicle Velocity Prediction & Estimation in 2d Video for Night Condition

Satya Kalyan A[1], Divakar T[1], K N Rao[1], Mani Chakravarthi A[2]

[1] *CSE Department, PVP Siddhartha Institute of Technology, Vijayawada, AP, INDIA.*
satyakalyan_a@yahoo.com, thirukkovalluri@gmail.com, drknrao@ ieee.org

[2] *Network Systems, VERIZON Data Services, INDIA*
manichakravarthi.a@gmail.com

Abstract

A 2D-video processing technique for automatic detection of incoming vehicles at night light conditions is a challenging task for any Advanced Driver Assistance Systems (ADAS). *We present a novel image processing technique to be used by ADAS to detect and track incoming vehicle's front headlamp for estimating real time velocity under such conditions. To capture the incoming traffic a standard C-MOS camera is mounted in the front panel of the vehicle and pre-processing filters are applied on each frame for removing impulse noise. Headlamp blobs are segmented from the rectified frame using intensity thresholds. Area of the blob is calculated based on the motion data and is used to predict the real time velocity of the vehicle. Optical flow concept is applied to predict Motion in the headlamps and tracked by kalman filtering. Experiments are conducted to estimate the training set parameters of the distance function. Results that demonstrate system's high velocity prediction rates using the best practices of image processing and optical flow are presented.*

Keywords: *Vehicle velocity, optical flow, image rectification, image processing, computer vision.*

1. Introduction

Studies on Visual performance under dim lighting conditions [1] and over focus of oncoming vehicle light, suggest that in the European Union almost one third of road fatalities occur during night conditions and are an important area of focus for road safety. Demand for advanced driver-assistance systems (ADAS) is expected to assist the driver in controlling the direction and speed of the vehicle knowing the dynamics of the incoming vehicle to eliminate accidents. ADAS can be built using Automatic cruise control implemented with active sensors such as RADARs. Usage of Cameras could be a low cost alternative and complementary technology which minimizes interference problems under dense traffic conditions. Detection of incoming vehicle object from a video sequence is essential in building ADAS navigation system, which provides vehicle velocity information and assist in decision making while moving in traffic and about to cross other objects under night light conditions. The existing techniques are good enough to track and estimate velocity of the vehicle under day conditions [2]. Development of such an intelligent system with vision under night conditions is critical. Identifying the moving objects will be a primary task of such an intelligent systems; existing techniques [3] requires prior information of ground plane which is very difficult task to provide and suffers from cluttering of objects. Other techniques such as GPS systems are very

costliest and time taking process [4]. P.F.Alcantarilla etal [5] has proposed a technique to estimate distance of an incoming vehicle under night light conditions, which requires calibrated camera data; to have dynamic calibration it requires a precise technique. The system fails for different road conditions. In this paper we propose a technique which does not require camera calibration. Our technique captures incoming vehicle's head lamp blob area to estimate vehicles distance and velocity. It is clear from law of optics that the sizes of the object increases as it approach the source which is used to predict the direction of the vehicle. To improve the performance of the system we propose to employ preprocessing noise reduction techniques. Optical flow [6][7] is applied on the segmented frame to classify the static object from the moving object in the captured scene. To increase the efficiency, image rectification [8] is performed on each frame prior to the optical flow. Motion data is used to calculate the area of the head lamp blob. Kalman filtering [11][12] is used in Tracking of headlamps the system is trained with different area values at different distances and for different types.

2. System Overview

This paper implements a velocity measuring technique for ADAS. The block diagram of such a technique is shown in fig (1). The first block is a video capturing unit to record the traffic information along with the static background information such as street lights, hoardings etc. The camera is a C-MOS type, and is mounted in the front panel of the vehicle

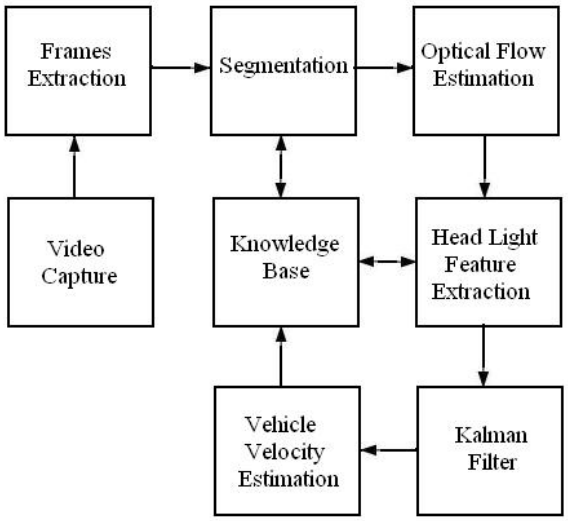


Figure 1: Block Diagram of Velocity Measurement.

The second block is responsible to extract the frames from the video with frame rate that of camera frame rate. Median filtering technique is applied on the frame to minimize the impact of impulse noises generated by other sources of light. The preprocessed frames are sent to the third stage as inputs which perform segmentation to segregate different objects in the frame. Different intensity thresholds are used in segregation and grouping of headlight blobs. This block is also responsible for supplying information to knowledge base. The fourth block of the system calculates optical flow for all the object blobs in the frame to identify motion i.e. All moving vehicles in the input video. This stage takes two rectified images as input and

determines motion of each pixel by comparison. Identified moving object are measured for blob properties like centroid, area, and bounding boxes by the fifth block.

Distance between objects and camera is estimated using a distance function, which is further used to estimate the velocity of incoming vehicle. The paper proposes a method to derive distance function, by low cost and high performance system with less complexity to track and estimate the vehicle velocity at night light condition. In this technique we employed Lucas-Kanade optical flow which is much simpler, effective and a good choice of measuring optical flow than other algorithms like horn and schunck method and a Kalman filter. The method is much accurate as it combines blob properties along with the optical flow techniques in prediction of velocity.

3. Optical Flow

Optical flow technique estimates the motion fields between two image frames. Optical flow is a velocity filed of a given pixel in an image and most commonly optical flow velocity data differ from the actual ground plane motion field data. Motion field data is transformed into optical field represented by the field of vectors that show the displacement of points in the one image frame to another frame. There are numerous ways to compute the optical flow and are classified into gradient-based, energy-based, and correlation-based methods. Lucas–Kanade method used differential method for optical flow estimation and is widely used in computer vision. It assumes that for a given pixel the velocity is constant in a local neighborhood, and solves the basic optical flow equations for all the pixels in that neighborhood, by the least squares criteria. Inherent ambiguity of the optical flow equation is solved by Lucas-Kanade optical flow by combining information from several neighboring pixels.

3.1 Lucas–Kanade Optical Flow Equation

The Lucas-Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p. Namely, the local image flow (velocity) vector (V_x,V_y) must satisfy

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ &\vdots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned}$$

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

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_{21}) & I_y(q_{12}) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \qquad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \qquad \text{and} \qquad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

This system has more equations than unknowns and thus it is usually over-determined. The Lucas-Kanade method obtains a compromise solution with the weighted least squares principle. Namely, it solves the 2×2 system

$A^T A v = A^T b$ or $v = (A^T A)^{-1} A^T b$

Where A^T is the transpose of matrix A. That is, it computes

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

With the sums running from i=1 to n.

The matrix $A^T A$ is often called the structure tensor of the image at the point p. The plain least squares solution above gives the same importance to all n pixels q_i in the window. In practice it is usually better to give more weight to the pixels that are closer to the central pixel p. For that one uses the weighted version of the least squares equation,

$A^T W A v = A^T W b$ or $v = (A^T W A)^{-1} A^T W b$

Where W is an $n \times n$ diagonal matrix containing the weights $W_{ii} = w_i$ to be assigned to the equation of pixel q_i. That is, it computes

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

The weight w_i is usually set to a Gaussian function of the distance between q_i and p.

4. Kalman Filter Tracking

We perform tracking to predict the future position of targets, correct each object's velocity estimation, detect noise and vehicle movement noise, to interpolate position during short periods where detection has failed, and to thus aid temporal association between frames. For tracking, we use the Kalman filter, which is a least-squares estimation of linear movement with an efficient implementation, and it requires the tracking estimate from one previous frame to be stored in memory. Once classified as vehicles, we track targets using the four parameters of a bounding box surrounding the headlamp (x-position, y-position, width, and height); these form a state vector $\hat{x}$. First, predictions of the state vector $\hat{x}^-$ and state error covariance matrix C^- are generated for a target at time k, i.e.

$$\hat{x}_k^- = A\hat{x}_{k-1} \tag{1}$$

$$C_k^- = AC_{k-1}A^T + Q \tag{2}$$

Where, A is the state transition matrix, and Q is the process noise covariance matrix. These predictions are then used to associate detections in the current frame with targets being tracked. This system measurements z is used to correct and update the corresponding trackers. The Kalman gain K is computed by

$$K_k = C_k^- H^T (H C_k^- H^T + M)^{-1} \quad (3)$$

Where, H is the matrix that relates the true state space with the measurement space, and M is the measurement noise covariance matrix. This Kalman gain is then used to correct the previous estimate of state and error covariance, i.e.,

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (4)$$
$$C_k = (I - K_k H) C_k^- \quad (5)$$

5. Object Velocity

To acquire each object's location on the surveillance frames, one must first distinguish these from the background scenario. This process is known as image segmentation and allows the identification of objects on the image plane. To estimate object velocity and correct each object's position a filter based tracking system is employed. This process predicts future positions given a sequence of images, and corrects these estimations with the information provided by blobs extracted from the segmented image and corrects the system's model. In order to do so, this process reiterates four steps for each new frame. The first consists of estimating each object's distance from observer, centroid and area and thus the velocity.

5.1 Distance Function

To estimate the velocity of the system, it requires the distance at which the object is present from the camera position. We propose to use the extracted area of the blob in estimating the distance. The area of the blob increases as it approaches the observer and decreases as it moves away from the observer and hence the direction of the vehicle is known. Figure (2) gives us an idea of how area and distance of the object are related.

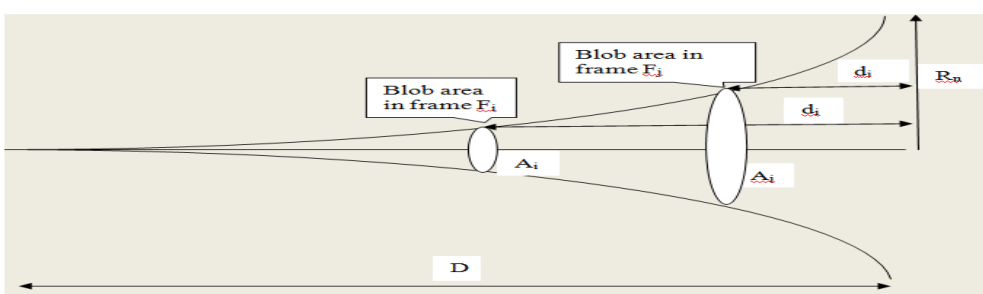


Figure 2: Blob Area and Distance Relation Over Frames

The area of the headlamp blob is initially measured at known displacements to train the system .The prior training data plot between area and distance suggests that they are exponentially related. The exponential coefficients are hence estimated. Equation (7) relates Distance as function of Area.

$$d_i = f(A_i) \quad (6)$$
$$f(A_i) = ae^{bA} + ce^{dA} \quad (7)$$

Where,
a, b, c and d are coefficients and depends on Source of the blob, Camera parameters like focal length, resolution. A is the area of the blob.

5.2 Velocity Estimation

Velocity of the vehicle V is measured as the change in relative distance to difference in time between two frames f_i, f_j as Equation (8).

$$V = S/T,$$

$$V = \frac{di - dj}{T} \tag{8}$$

Time interval T, is calculated using equation (9).

$$T = \frac{1}{C} \times (f_i - f_j) \tag{9}$$

From equations (8) and (9), velocity of the vehicle is calculated as a function of area and frame number. The sign of the function gives us the direction of the vehicle.

$$V = C \times \frac{(f(a_i) - f(a_j))}{f_i - f_j} \tag{10}$$

Where
C - Camera frame rate.
$f(a_i)$, $f(aj)$ - Distance function.
Fi,f_j - Frame numbers.

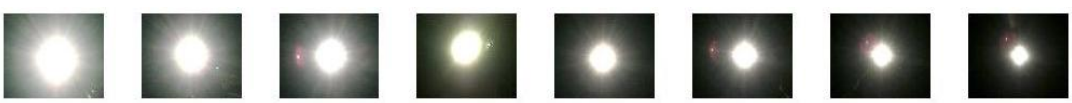


Figure 3: Training Data Image Sequence at Different Distances.

Figure 4: Segmented Training Data

TABLE 1:Prior Distances with Respect to Area of Blob

	Static image Areas Vs distance.								
Distance (mtrs)	4	8	12	16	20	24	28	32	36
Area (pixel^2)	664	414	329	290	204	172	119	75	66

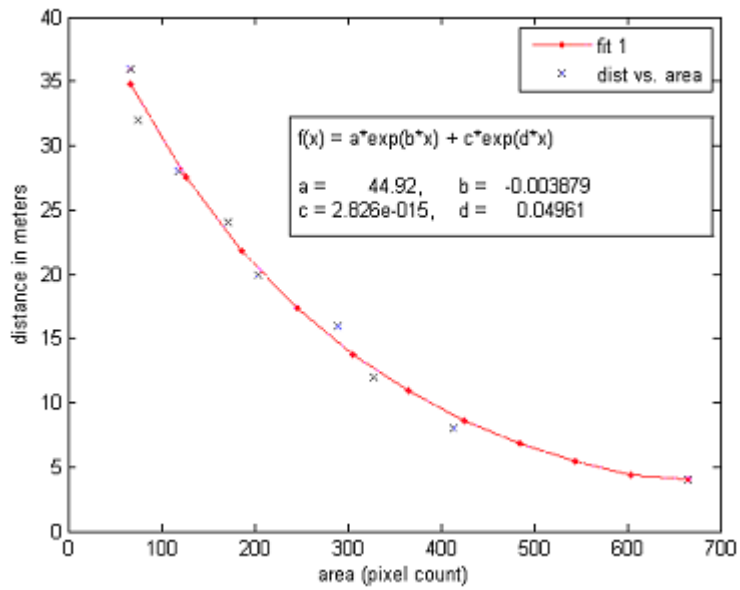


Figure 5: Distance Function From Area Data

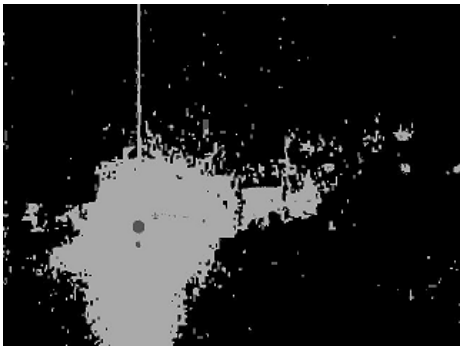
6. Results

A video capture device with frame rate of 25fps and a resolution of 352 X 288 is chosen. A Honda make motorcycle is selected to derive distance function. 20 Headlamp samples are captured at every 1 meter distance starting at 20 meters away from the vehicle. Camera is moved towards the vehicle to keep the vehicle at static position. Having this sample data as shown in TABLE -1 and a plot is drawn to show the relation between area and the distance as in Figure (5). The distance function coefficient values are calculated with 95% confidence bounds as a = 44.92 (40.4, 49.44), b =-0.003879 (-0.004586, -0.003172), c = 2.825e-015 (-1.245e-008, 1.245e-008) and d = 0.04961 (-6633, 6633). More than 10 minutes of on road video were captured, comprising speeds range from 10 to 70 kmph. Frames are extracted with the camera frame rate. The captured color image frames are converted to gray scale and a median filter is applied to eliminate impulse noise. Objects in the captured frame are extracted by applying a simple segmentation with a [200,255] threshold Figure 6(c). Static objects are eliminated by applying a Lucas Kanade optical flow algorithm as shown in Figure 6(d). The area of the object (pixel count) is used to find the distance. Captured video of the Motor vehicle traveling with speed of 20kmph, 30kmph and 40 kmph is chosen to test the distance function. Frames are extracted and the area of the blobs is calculated, derived distance function is used to estimate the velocity. The results of accuracy for different velocities are shown in TABLE (3, 4, 5). The velocity of the vehicle is estimated with >97% of accuracy. Table 6 concludes the summary of experimental results.

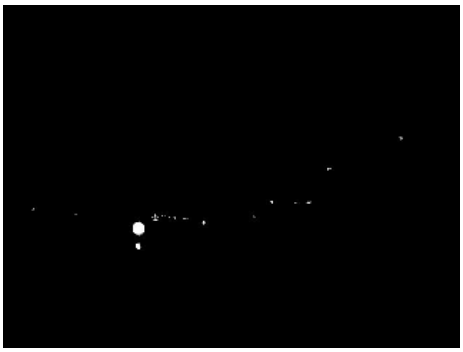
TABLE 2: Areas of Blobs at Different Velocities in Individual Frames

frames	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17	f18	f19	f20	f21
sample vel 1	83	83	82	82	76	76	78	78	81	81	83	83	88	88	87	87	92	92	91	91	93
sample vel 2	84	84	85	85	88	88	88	88	89	89	96	96	100	100	115	115	116	116	121	121	123
sample vel 3	82	82	95	95	107	107	119	119	123	123	128	128	132	132	140	140	157	157	173	173	179

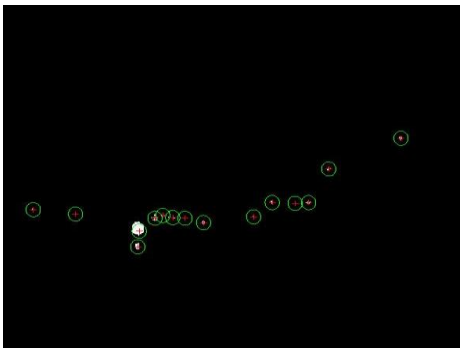
(a) Input frame



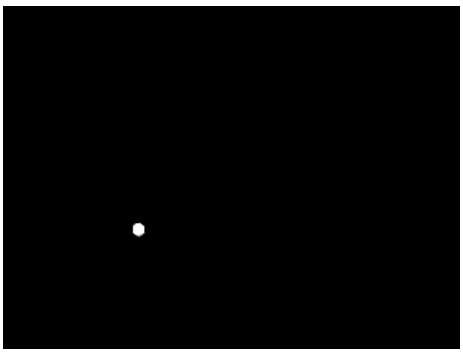
(b) Segmented frame



(c) Region Extracted



(d) Tracking



(e) Extracted Blob

Figure 6: Outputs of Different Processing Stages.

TABLE 3: Results for 20 kmph Data

Results for 20 kmph Data

Frame number (velocity 20 kmph)	Extracted area (pixel^2)	Estimated distance	Actual distance(meters)	Error rate(%)
1	83	32.55484254	32	1.73388293
2	115	28.7546108	28	2.695038554
3	147	25.39799236	24	5.824968158
4	201	20.59820899	20	2.991044959
5	296	14.24917671	16	-10.94264556
6	334	12.29628438	12	2.469036466
7	426	8.605741741	8	7.571771765
9	663	3.975855434	4	-0.60361415
		Average Error Rate (%)		1.304387014

TABLE 4: Results for 30 kmph Data

Results for 30 kmph Data

Frame number (Velocity 30 kmph)	Extracted area (pixel^2)	Estimated distance	Actual distance(meters)	Error rate(%)
1	84	32.42880691	32	1.340021587
2	115	28.7546108	28	2.695038554
3	152	24.91014449	24	3.792268722
4	201	20.59820899	20	2.991044959
5	278	15.27963858	16	-4.5022589
6	340	12.0134053	12	0.111710856
7	428	8.539237181	8	6.740464757
9	678	4.382781462	4	9.56953655
		Average Error Rate (%)		2.526425232

TABLE 5: Results for 40 kmph Data

Results for 40 kmph Data

Frame number (Velocity 40 kmph)	Extracted area (pixel^2)	Estimated distance	Actual distance(meters)	Error rate(%)
1	82	32.68136801	32	2.12927503
2	123	27.87600073	28	-0.442854525
3	173	22.96143639	24	-4.327348391
4	188	21.66355011	20	8.317750557
5	272	15.63942751	16	-2.253578085
6	332	12.39204994	12	3.267082832
7	434	8.342791638	8	4.284895474
9	632	3.987253007	4	-0.318674819
		Average Error Rate (%)		1.184060897

TABLE 6:Summary of Experimental Results

Speed(Kmph)	Distance Range(mts)	Velocity Estimation Rate(%)
20 Kmph	0-20 m	98.696
30 Kmph	0-20 m	97.474
40 Kmph	0-20 m	98.816
Average Estimation Rate		98.328

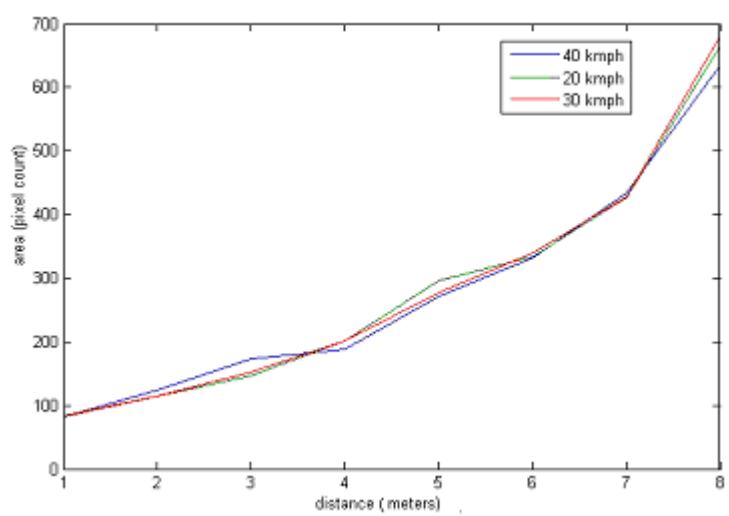


Figure 7: Blob Area Vs Distance from Different Sample Velocities

7. Conclusion

In this paper we have described a technique that detects an incoming vehicle under night light condition, estimate the velocity, the distance to develop an effective ADAS. We discuss the advantages of using C-MOS passive device against the active devices. Lucas–Kanade optical flow algorithm is implemented along with the basic segmentation technique for effective motion detection to improve robustness of the system. Results from on road testing data have been presented to demonstrate high velocity production rates.

Future work may enhance the technique to identify different types of vehicle by strengthening system learning module which needs parameter estimation of distance function for different types of headlamps.

Acknowledgements

We thank Dr.N B Venkateswarlu ,Professor of Computer Science and Engineering and Dr.E V Prasad Professor of Computer Science and Engineering JNTUniversity Kakinada ,India for their constant suggestions in the development of the technique.

References

[1] S Plainis, I J Murray, and I G Pallikaris ,"Road traffic casualties: understanding the night time death toll", Injury Prevention, April 2006, Vol 12 no.2 pp125–128;doi: 10.1136/ip.2005.011056.

[2] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal Mach. Intell.*, ,May 2006, vol. 28, no. 5, pp. 694–711

[3] C. Maduro, K. Batista, P. Peixoto, and J. Batista, *"Estimation of vehicle velocity and traffic intensity using rectified images*". Proceedings of the 15th IEEE international conference on Image processing, 2008

[4] David J. Fleet and Yair Weiss, "Optical Flow Estimation". In Paragios et al- *Handbook of Mathematical Models in Computer Visio*, 2006, Springer. ISBN 0387263713.

[5] D. J. Dailey and F. W. Cathey, "A novel technique to dynamically measure vehicle speed using un calibrated roadway cameras," in *Proceedings IEEE*, 2005.

[6] B. Bose and E. Grimson, "Ground plane rectification by tracking moving objects", in *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[7] Luis Serrano, Donghyun Kim and Richard B. Langley, Kenji Itani and Mami Ueno, "A GPS Velocity Sensor: How Accurate Can It Be? – A First Look", 2004.

[8] T. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," in *IEEE Transactions on Intelligent Transportation Systems*, 2003.

[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings DARPA Image Understanding Workshop*, 1981.

[10] B.K.P. Horn and B.G. Schunck, "Determining optical flow" *Artificial Intelligence*, 1981 volume 17, pp 185-203.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng*, Mar. 1960. vol. 82, pp. 35–45

[12] G. Welch and G. Bishop, "An introduction to the Kalman filters," Dept. Computer. Sci., Univ. North Carolina, Chapel Hill, NC, Tech. Rep. TR 95-041, 2003.

Authors

Satya Kalyan.A

Working with computer science and engineering Department since in PVP Siddhartha Institute of Technology since 2006. He received his B.Tech. Degree in Electronics and Communications engineering with distinction from JNT University, HYDERABAD, INDIA, In 2001, and the M.Tech degree in Computer Science Engineering from JNT University, HYDERABAD, INDIA, in 2004.His area of research is DIGITAL IMAGE PROCESSING.

Divakar T

Born in 1981 and acquired B-tech from Gudlavalleru Engineering College, Gudlavalleru in 2006, in Information Technology. Associated with same college for the period 2006'June to 2008' June, as lecturer. Now doing M tech degree in Computer Science and Engineering from Prasad.V.Potluri Siddhartha Institute of Technology, Vijayawada. His current area of interest is Image Processing

Nageswara Rao K

He was born in 1955. He received the B.E. degree in Electronics and Communication engineering with First Class from Karnatak University, Dharwad, in 1977, and the M.Tech. and Ph.D degree in Computer Science and engineering from Andhra University. After serving Andhra University for nearly 30 years in teaching and research he is currently with PVP Siddhartha Institute of Technology, Vijayawada as Professor and Head of CSE department. His current research interest is in the areas of Robotics, image Processing, Information Security and Refactoring.

Mani Chakravarthi A

He received his MSc degree in Computer science from Nagarjuna University.

He is working as Analyst for the past 2 years at verizon India. He is having around 5 years of experience in software development working on various domains. He is currently perusing his M.Tech degree in computer science and Engineering at JNTUH.

.