Object Velocity Estimation Based on Asynchronous Data from a Dual-Line Sensor System

A.N. Belbachir, Member, *IEEE,* M. Hofstätter, K. Reisinger, N. Donath and P. Schön

Abstract— **This paper presents a 2×256 dual-line vision system for object velocity estimation based on the asynchronous output data. For real-time velocity estimation, a processing method has been developed and implemented on the digital signal processor. It exploits the sparse asynchronous data representation from the dual-line sensor with high temporal resolution (better than 100 μs) and low latency. The processing concept includes the object contours extraction, velocity estimation and scaling. Three approaches are used and evaluated for the velocity estimation. The first and second approaches use the mean and median object detection time, respectively for the object velocity calculation. In the third approach, the statistical linear fit RANSAC is used to extract the object time. This processing concept has been evaluated on object velocities that range from 1 to 22 m/s, for the three velocity estimation approaches and a comparative study has been provided. The experimental results included show a velocity estimation error < 1% using RANSAC.**

I. Introduction

NOWADAYS several systems exist for object velocity estimation that include optical sensors like LIDAR (***L**aser **I**maging **D**etection **A**nd **R**anging*) [6][12], non-optical sensors, like microwave RADAR (***RA**dio **D**etection **A**nd **R**anging*) [7], audio systems [10] or inductive loops [1] and using video-based systems [2][3].

Several processing tools [2][5] have been developed as well to improve the velocity estimation accuracy by pre-processing (e.g. data filtering and normalization) and interpreting the sensors' data. Most of the tools have been developed for video-based systems as they offer a wide range of applications, which are driven by the implemented algorithm.

Two parameters are required for the velocity estimation of a moving object: the object trajectory and the time spent by the object to pass over a certain distance along the trajectory. Therefore, several acquisitions of the object have to be taken at different time intervals to extract the trajectory and the time information. Classical video-based systems equipped with array pixels or multiple-line pixels produce constant data volume at a fixed rate ranging from 1 to 1000 images/s. Considering a moving object at a velocity of 10 m/s on a field of view of 10 cm, then the object takes 10 ms to cover the field of view. Therefore, a minimum rate of 100 images /s has to be used to calculate the average object speed. For faster objects motion, the acquisition rate has to be further increased to allow the object velocity estimation. The major problem with the classical video-based systems is the type of the data acquisition that is synchronous, such that a constant data volume is constantly delivered by the sensor for the estimation of the high-speed object velocity. Thus, the readout and processing of the largely redundant data ultimately face limitations due to computational effort and power consumption.

An asynchronous transient vision sensor has been developed [8] containing an array of 128×128 autonomous, self-signaling pixels, which individually respond in real-time to relative light intensity changes by generating events coding the pixel address upon activity on the scene. Pixels that are not stimulated by a change in illumination are not triggered; hence static scenes produce no output. This system has been demonstrated for vehicle speed estimation and results have been published in [9]. For this system, the time information "timestamp" is attached to the address events at the processing unit level with 1ms resolution.

The vision sensor architecture has been modified and a dual-line sensor chip has been developed [11] that includes a clock to provide more accurate time information at the readout electronic level with up to 10 MHz frequency resolution (≥ 100ns time interval). This paper presents the concept and results for the object velocity estimation using this dual-line sensor with three estimation approaches: the Mean, the Median and a linear fit using RANSAC (**RAN**dom **SA**mple **C**onsensus) [4].

The paper is structured as follows. In section II, the dual-line sensor system is presented including the data format. Section III describes three approaches for the velocity estimation. The experimental results after the application of the presented algorithm on the real data are discussed in section IV including a comparative study. A brief summary for conclusions is given in section V.

Manuscript received January 15, 2007. This work was supported by the Austrian Council for Research and Technology Development.

All authors are with the Austrian Research Centers GmbH - ARC, Smart Systems Division , Donau-City Str. 1, A-1220 Vienna, Austria

Ahmed Nabil Belbachir's phone: +43-50550-4215; fax: +43-50550-4125; e-mail: ahmed.belbachir@arcs.ac.at.

II. The Dual-Line Sensor System Description

The dual-line sensor system contains a sensor chip and a processing unit for the analysis and the interpretation of the sensor data. The dual-line sensor system and the data format are briefly described in the following subsections.

A. Dual-line Sensor system

The layout of the dual-line sensor system is depicted in Fig.1. It comprises the following modules:

1. A "silicon retina" sensor chip [11] that contains a dual-line arrangement with 2×256 autonomous pixels, which are sensitive to local temporal contrast. The distance between the two lines is 250μm (on-chip). This distance and the event generation time are important parameters for the velocity estimation of objects crossing both lines.
2. Auxiliary electronics for on-the-fly configuration of the sensor chip. This allows adapting the sensor to different scene conditions like varying illumination, object reflectance and speed.
3. A FIFO to handle peak data rates.
4. A data processing unit that is responsible for treating and interpreting the data. This module contains the implemented algorithms for the real-time object velocity estimation that are described in section III.

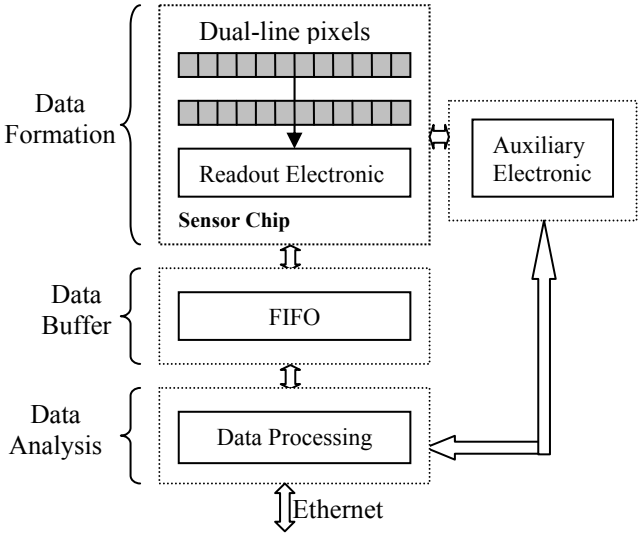

Fig. 1. A diagram of the dual-line sensor system

B. Data Format

The dual-line sensor chip encodes the information in the form of TAE (***T**imed **A**ddress **E**vent*) representation. The TAE stream consists of the event generation time (timestamp) concatenated to the AE (*Address **E**vent*), which encodes the coordinates of the pixel in the 2×256 matrix. The sensor chip can generate a timestamp at a maximum frequency of 10 MHz (time interval ≥ 100 ns) that is far too high for any existing vision sensor chip.

The AE are classified into two types; ON-events that represent a fractional increase in intensity and OFF-events that reflect a fractional decrease. The TAE data word is 16 bit wide where the most significant bit (bit 15) is used to distinguish between the timestamp and the AE. Examples of the TAE data streams are depicted in Fig. 2. One timestamp can be assigned to one or several AEs, which occur in the next timestamp interval (see the upper part of Fig. 2). A timestamp can also be generated without following AEs if the Timestamp counter wraps around (see the lower part of Fig. 2).

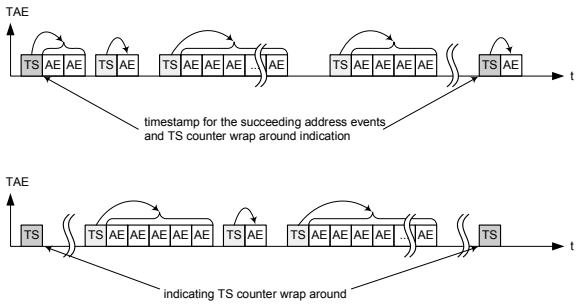

Fig. 2. Two examples of TAE data stream

III. Velocity Estimation

The object velocity estimation concept (Fig. 3) consists of four steps: object detection, contours extraction, velocity estimation and scaling. Those steps are detailed in the next subsections.

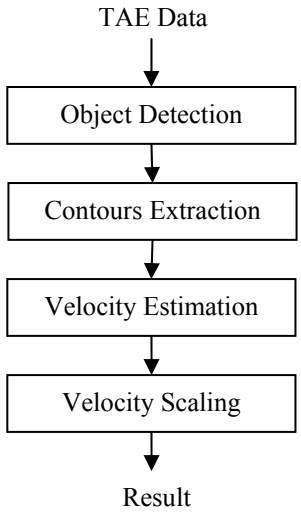

Fig. 3. Overview of the velocity estimation concept

The 2-D object representation with the dual-line sensor consists of the pixel index (y-axis) versus the time (x-axis). The x-axis represents the event generation time in units of the timestamp period, the y-axis is the pixel address (0-255). Fig. 5 shows data examples from one pixel line for a ball and a cube passing across the sensor's field-of-view

with a velocity of 0.8 m/s at a distance of about 15 cm. The filled dots represent the OFF events while the circles show the ON events. The timestamp interval was configured to 10µs. The original object shapes (ball and cube) are depicted in Fig. 4. It can be noticed that the time (x-axis of Fig. 5) can be scaled to retrieve the original shape dimension. The scale factor is the object velocity v as the distance x is calculated as follows:

$$x = v \cdot t \quad (1)$$

where t is the event generation time. Therefore, an accurate velocity estimation is also useful for the exact shape representation.

Fig. 4. Original shapes: ball (right) and cube (left)

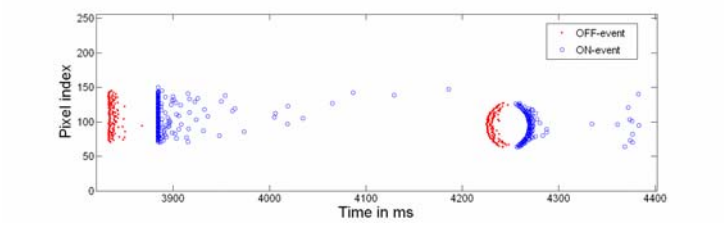

Fig. 5. Representation of data from the ball (right) and the cube (left) with the dual-line sensor chip

A. *Object Detection*

The first processing step consists of detecting an object out of the TAE data stream. The first intuitive idea is to monitor the TAE data rate and to flag an object when the event rate exceeds zero to a maximum rate. Fig. 6 presents the rate (event/ms) of the OFF-events (Fig. 6.(a)) and the ON-events (Fig. 6.(b)) for both objects shown in Fig. 4. It can be noticed two clusters of events and many isolated peaks, mainly due to the ON-events outliers.

To improve the object detection, an average filter on the event rate has been performed for smoothing the event rate distribution to remove the outliers and better localize the object. Fig. 7 shows the average filter results on the events rate. It can be clearly noticed the smoothness of the events rate distribution over the time that facilitate the object detection.

B. *Contours Extraction*

The contours extraction algorithm removes the events, which are considered not being part of the shape, from the data stream. In case of smooth contrasted object edges, every pixel may generate several consecutive events due to the long lasting contrast amplitude differences. Therefore, it is assumed that the object edge can be reconstructed by the first event per pixel. The algorithm distinguishes between ON and OFF events during the events removal. It retains the first ON and OFF event per pixel occurring within a predefined time interval. The time interval is calculated starting from the first event occurrence. The interval strongly depends on the object velocity and size and has to be adapted to the application at hand.

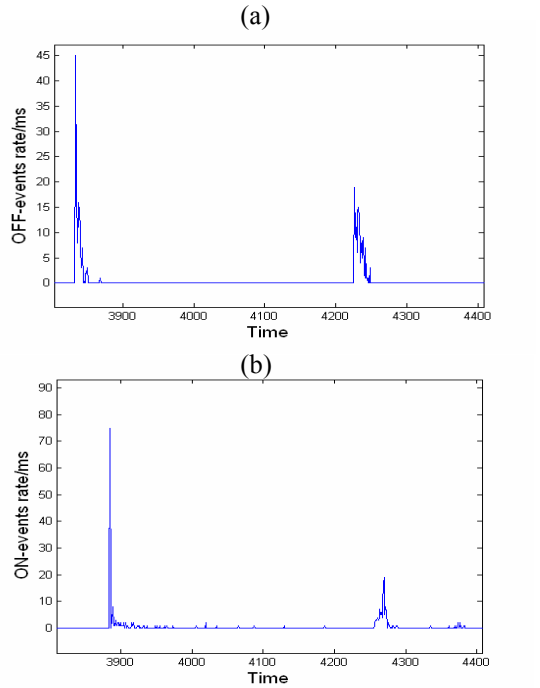

Fig. 6. Monitoring of OFF-events (a) and ON-events (b) rates

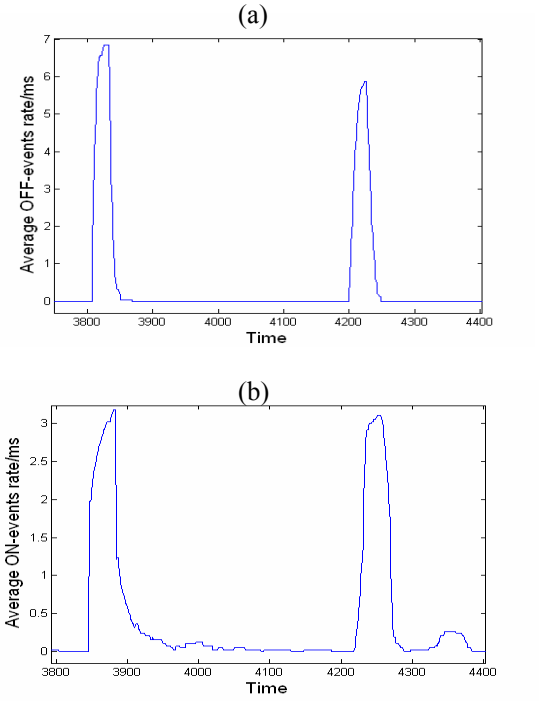

Fig. 7. Monitoring of average OFF (a) and average ON (b) events rates

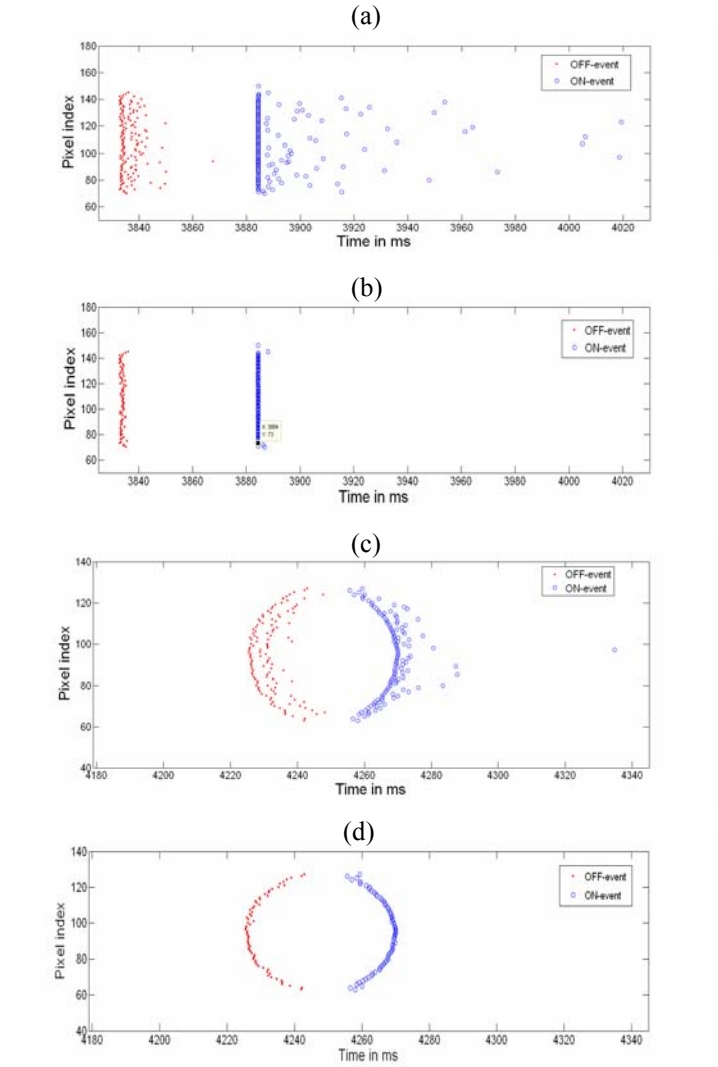

Fig. 8. Original data from a cube (a) and a ball (c) and their resulted data after the contours extraction showed in respectively (b) and (d).

Fig. 8 shows the contours extraction results using data from the cube and the ball. It can be seen that the object shapes have been correctly extracted while removing the isolated events (outliers). All the results given so far were performed on data from one line of pixels. For the velocity estimation, the processing is performed on data from the two lines of pixels.

C. Velocity Estimation

For the velocity estimation either ON or OFF events from both lines of pixels can be used. In our example (Fig. 8), the OFF-events are used for the estimation as they present worse distribution than the ON-events in terms of uniformity of the distribution and the pixel jitter, for the worst case scenario. Three approaches are used for the generation time estimation of the OFF-events: the Mean, the Median and the RANSAC fit [4]. All three approaches aim to extract the time difference of the OFF-events occurrence between both lines of pixels in order to estimate the object velocity v_1 using this equation:

$$v_1 = \Delta x / \Delta t \tag{2}$$

where Δx is the distance between both lines of pixels i.e. 250µm and Δt is

$$\Delta t = t_{OFF1} - t_{OFF2} \tag{3}$$

where t_{OFF1} is the OFF-events occurrence time from one object on the first line and the t_{OFF2} is the OFF-events occurrence time from the same object on the second line. Therefore, the velocity estimation accuracy depends on the t_{OFF1} and t_{OFF2} accuracy that is evaluated using the three different approaches.

The **Mean approach** extracts the average time of the OFF-events. This approach has the advantage to be fast and to provide a good measure of central tendency for roughly symmetric distribution of the events. However, it can be misleading in case of skewed distributions as in the presence of outliers.

The **Median approach** extracts the median time of the OFF-events after sorting the occurrence time in an ascending form. This approach has the advantage to be useable for highly skewed distributions as it is robust for transient phenomenon and against outliers.

The **RANSAC approach** is an analytic procedure, which fit the data to a straight line. It performs the following:

1. Take randomly two samples and calculate the line which passes exactly through these samples.
2. All samples that are within a pre-specified distance θ to the line are put into the support set.
3. Repeat this process many times.
4. Select the line with the largest support set (if there is more than one take the one with the smallest residual error).

The complexity of RANSAC exponentially increases with the number of measurements to fit. However, a subset of points can be taken to speed up the processing. It has been shown that RANSAC obtains the theoretically optimal breakdown point of 50%, i.e. it still can fit a line if not more than 50% of the measurements are outliers.

After extracting the slope and the offset value using the RANSAC approach, a time is calculated for the central (median) pixel for each line that is used for calculating the time difference Δt between both lines.

Fig. 9 presents the time calculation on both line using the mean (:), the median (-.) and the RANSAC (- -) approaches. The calculated Δt is 9.4 ms using the mean, 10.2 ms using the median and 10.6 ms using the RANSAC approach. This shows the accuracy differences between the three approaches. Indeed, the case of the ball is worse than

with the one with cube as its edge distribution is more compact with respect to the occurrence time.

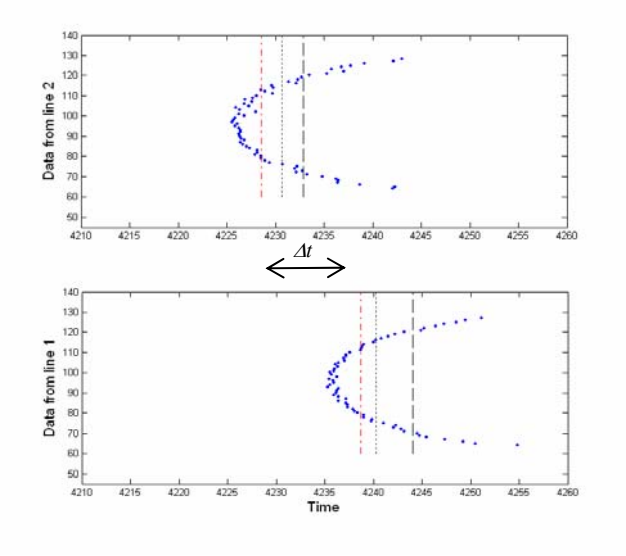

Fig. 9. OFF-events occurrence time estimation on pixels line 1 and line 2 using the Mean (:), Median (-.) and RANSAC (- -) approaches.

D. *Velocity Scaling*

In the previous subsection, the velocity has been estimated on the chip dimension. The aim of this step is to convert this velocity estimate to the scene coordinates. This is a critical issue as it depends on an external factor that is the crop factor of the optics. This problem can be solved if this magnification factor is known for a specific lens and for a dedicated distance between the sensor and the object. Otherwise, a calibration process is needed.

In our case, the calibration is performed for an object with known dimension such that we map the real object dimension to the object in pixel size. The crop factor A is calculated as follows:

$$A = D / N \cdot P \tag{4}$$

where D is the original object dimension, N is the number of pixels, which map the object and P is the pixel length. After estimating the magnification factor, the object velocity v can be calculated using

$$v = A \cdot v_I \tag{5}$$

IV. Experimental Results

The dual-line sensor system has been evaluated using the above-listed algorithms for objects passing the system at different velocities ranging from 1 – 22 m/s. The four processing steps have been performed and the results have been collected. Fig. 10 and Fig. 11 illustrate the average absolute estimation error and the average relative estimation error, respectively, in function of the velocity for 25 objects. Both figures present the experimental results using the three approaches: the Mean (circles), the Median (diamonds) and RANSAC (triangles). The Mean algorithm provides a relative error ranging between 1 – 6%, the Median has an estimation error of 0.5 – 1.3% while RANSAC has an error ranging between 0.4 – 1%. Thus, RANSAC provides better estimation of the event occurrence time than the Mean and the Median estimation as it is robust to transient phenomenon such that it selects the support set (line) with the largest events number.

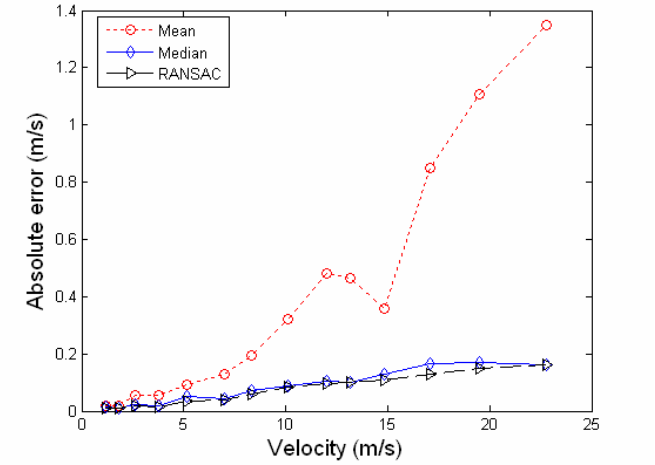

Fig. 10. Illustration of the absolute error (m/s) on the velocity estimation w.r.t. the velocity amplitude using the Mean, Median and RANSAC approaches

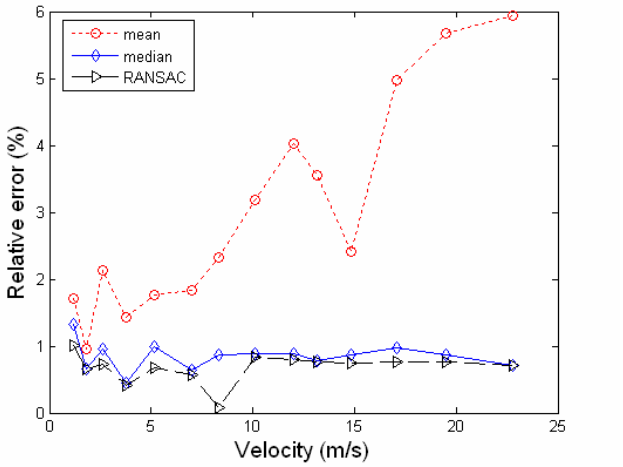

Fig. 11. Illustration of the relative error on the velocity estimation w.r.t. the velocity amplitude using the Mean, Median and RANSAC approaches

There are other sources of error like the optic distortion, the crop factor accuracy and the accuracy of the ground truth. A better optic quality has to be chosen to cope with the optic distortion and an improved calibration method has to be used to have an exact crop factor. Moreover, the ground truth velocity error was estimated to 0.1% in this test. By considering these sources of errors, we can assume that the depicted results are promising and the velocity estimation error can be further decreased.

V. CONCLUSION

In this paper, an object velocity estimation concept has been presented for the asynchronous data from the dual-line sensor system. This system provides sparse events representation with a temporal resolution better than 100 µs on a data stream delivered upon activity on the scene. The processing includes the object detection, contours extraction, velocity estimation and scaling steps. Three approaches have been evaluated for the velocity estimation, the Mean algorithm, the Median and a linear fit using RANSAC. The best estimation performance has been provided by the RANSAC fit with a relative error < 1%. However, it presents high computational complexity as it is 50 times slower that the Mean approach by processing data from up to 100 object /s.

As a perspective, the processing concept will be further extended to improve the velocity estimation accuracy by means of accurately calibrating of the optics and to decrease the computational complexity of RANSAC by means of selecting a subset of points for the linear fit.

REFERENCES

[1] T.Cherrett, H. Bell and M. McDonald, "Estimating Vehicle Speed Using Single Inductive Loop Detectors," *Proceedings of the Institution of Civil Engineers Transport*, vol. 147, nº1, pp. 23-32, UK, 2001

[2] D.J. Dailey, F.W. Cathey and S. Pumrin, "An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras ," *IEEE Trans. On Intelligent Transportation Systems*, Vol.1, No2, pp. 98-107, 2000.

[3] D. Douxchamps, B. Macq, K. Chihara, "High accuracy traffic monitoring using road-side line-scan cameras," *IEEE ITSC 2006*, Toronto, Canada, September, 2006

[4] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications ACM 24*, pp.381–395, June 1981.

[5] L. Grammatikopoulos, G. Karras and E. Petsa, "Automatic Estimation of Vehicle Speed from Uncalibrated Video Sequences," *Proc. Of the International Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*, pp. 332-338, Bulgaria, November 2005.

[6] A. Kirchner, U. Lages and K. Timm, "Speed Estimation with a Laser Rangefinder," *Proceedings of ICARCV'96*, pp. 1875–1879, Singapore, 1996.

[7] J. Langheim, J.F. Henrio and B. Liabeuf, "ACC Radar System – Autocruise – with 77 GHz MMIC Radar," *International Conference of ATA Florence*, 1999

[8] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128×128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change," in *IEEE International Solid-State Circuits Conference (ISSCC2006)*, San Francisco, USA, February 2006.

[9] M. Litzenberger, A.N. Belbachir, N. Donath, G. Gritsch, H. Garn, B. Kohn, C. Posch and S. Schraml, "Estimation of Vehicle Speed Based on Asynchronous Data from a Silicon Retina Optical Sensor," *IEEE Conference on Intelligent Transportation Systems*, Canada, 2006.

[10] R. López-Valcarce, C. Mosquera and F. Pérez-González," Estimation of Road Vehicle Speed Using Two Omnidirectional Microphones: a Maximum Likelihood Approach," *EURASIP Journal of Applied Signal Processing*, vol. 8 pp.1059-1077, 2004.

[11] C. Posch, M. Hofstätter, D. Matolin, G. Vanstraelen, P. Schoen, N. Donath and M. Litzenberger, "A Dual-Line Optical Transient Sensor with On-chip Precision Timestamp Generation," *IEEE International Conference on Solid-State Circuits,* USA, February 2007.

[12] A. Rakusz, T. Lovas and A. Barsi, "LIDAR-Based Vehicle Segmentation," *in International Conference on Photgrammetry and Remote Sensing,* Istanbul, Turkey, 2004.

Ahmed Nabil Belbachir received the Electronic Engineering degree in 1996 and the Master degree in Signal Processing, 2000, from the University of Science and Technology of Oran, (USTO) Algeria. In March 2005, he was awarded the Ph.D. degree in computer science from the Vienna University of Technology. Currently, he is research fellow at the Austrian Research Centers GmbH – ARC with research focus on real-time processing for the embedded systems with bio-inspired sensors. Previously, he was involved in the ESA-Herschel project, where he was responsible for the Data Reduction and Image Compression for PACS instrument. He has developed the on-board reduction/compression software for the IR photo-detector camera PACS. His research interests include real-time systems, data compression and signal/image processing and he has published more than 30 scientific publications. Ahmed Nabil is an IEEE member, a member of the IAPR TC13 (Technical Committee for Pattern Recognition in Astronomy and Astrophysics), AAPR (Austrian Association for Pattern Recognition), OeGAA (Austrian Association for Astronomy and Astrophysics), and EURASIP.

Michael Hofstätter received M.Sc. degree in electronics engineering from Vienna University of Technology. He collected wide experience in the ASIC design methodology, verification methodology, and the digital circuit development during working as digital design engineer for Application Specific Integrated Circuits (ASICs). Afterwards he was concept engineer for UMTS baseband processors with focus on the definition of the hardware architecture on system level (SoC) and peripheral level, including the definition of the hardware-software interface. In 2004 Michael joined the Smart Systems division at Austrian Research Centers GmbH – ARC, where he is developing novel CMOS image sensors and high integrated vision systems for surveillance and industrial applications. Michael is author and of some publications and inventor and of several patents.

Karl M. Reisinger received his M.Sc. in electrical engineering from Vienna University of Technology; currently working on his Ph.D degree. Since 2005 research fellow at Smart Systems division at Austrian Research Centers GmbH – ARC.

Nikolaus Donath received a MSc in theoretical physics from Vienna University of Technology. Since 2003 he is developing and implementing signal processing algorithms at the Smart Systems division at Austrian Research Centers GmbH – ARC.

Peter Schön took his M.Sc. degree in electrical engineering and his Ph.D. degree in technical physics at the University of Technology in Vienna. He worked as Hardware-Designer for measurement systems for nuclear research, medical devices and industrial production; afterward, he was responsible for assuring the quality of the design and production of industrial data collection systems according to the ISO9001 standard. As a specialist for digital signal processing he contributed to the design of SHDSL and ADSL transceivers in hard and software designs. Dr. Schön is an author and co-author of some scientific publications.