Vision-based detection of events using line-scan camera

Mirosław Jabłoński, Ryszard Tadeusiewicz
AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics
Computer Science and Biomedical Engineering
30-059 Kraków, Poland

***Abstract*—Vision-based method of event detection for control algorithm of acorn scarification device has been presented in the paper. Instead of frame-based approach, line-scan mode of digital camera has been used. The advantage of this approach is an improved temporal resolution of machine-vision-based event detector and reduced latency as the line-acquisition time is significantly lower than the acquisition time of a frame of 2D image. Segmentation of objects moving in front of the camera is one of the steps of the detection. Usage of two alternative methods of segmentation have been reported in the article: Sum of Absolute Differences and motion detection by means of adaptive Gaussian Mixture Model. Motion signal is then a subject of event detection.**

Keywords*—*image processing, line-scan camera, machine vision, event detection, gaussian mixture model

I. INTRODUCTION

In classical control and signal processing algorithms, periodical sampling of process variables is assumed. Sampling time in systems comprising machine-vision modules is limited by frame rate parameter of the camera system. Another approach, reported in scientific publications, is acquisition and processing of events in non-periodic [1][2][3], continuous time and asynchronous manner which allow for reduction of power consumption and cost [4]. An example of vision sensor that samples data in non uniform frame-free way is Dynamic Vision Sensors (DVS) [5] which latency can be as low as 3μs. Dedicated computing infrastructure is necessary for analysing signals in Adress Event Representation (AER) that is often used in neuromorphic vision systems [6] for simulation of neural processes. In that case both: acquisition and processing mimic biological vision systems [7]. It can be also used for time-critical control tasks at low computing power requirements [8]. Even though acquisition stage is usually asynchronous, subsequent stages of neuromorphic infrastructures comprise synchronous digital logic [9] or custom multiprocessors [10] that handle asynchronous AER events and Clock Domain Crossin (CDC) issues.

In this paper an attempt to improve event detection latency and temporal resolution is presented. The contribution of the work is the application of motion segmentation and event detection to scanned lines. Proposed line-scan camera solution is the trade-off between slow area scan cameras that are compatible with contemporary machine vision interfaces, and low-latency DVS sensors that are not in mass production yet. The paper is organised as follows. In section II an outline scarification process is provided. Chapter III includes details of the set-up used for video data acquisition, segmentation of motion and detection of events. Section IV contains summary and planned future work.

II. THE OUTLINE OF THE SCARIFICATION AUTOMATON

The work described in this paper is aimed at vision-based detection of acorns being conveyed by feeder which is the component of the functional model of the scarification automaton. The functional model will consist of further modules: vision-based detector, rotator, gripper, scarification module, machine vision system for assessment of seed viability and sorter. The sequence of operation can be the following:

- fetch the seed from container and submit to rotator,
- estimate length of the acorn and detect its distant end,
- rotate the seed if necessary and forward to scarifier,
- cut the distal end by scarification device,
- asses the viability of the seed and separate healthy from spoiled ones.

At the initial stage of the project, the model of the feeder was designed so as to acorn falls down into rotator through inclined plane limited by two V-shaped bands with. This causes however that the motion of the acorn is similar to free fall as the friction between the surface of the band is not major. One should note that scarification device should cut the fraction of the acorn at distal end preserve an embryo for sowing. At two stages of the proposed processing scheme, digital video-cameras are used: monochrome camera for detection of orientation of the seed, colour camera for analysis of oak seed's section after cutting its end. Necessity of event detection during operation of the automaton is explained by the need of dosage of subjects for scarification and estimating their lengths by computing delays between events.

III. VIDEO ACQUISITION AND PROCESSING

An industry-standard machine vision acA-2000-165 camera and PC computer has been used. It complies to USB3Vision protocol that enables high level of flexibility in image acquisition mode and transfer rate [11]. Data rate can be adjusted from 165 FPS (Frames Per Second) for full frame image with resolution 2088×1088 pixels up to 7000 Lines Per Second in line-scan mode (2088 pixels in line). In line-scan mode each time the same line of the sensor is acquired whilst in area-scan mode readout is performed for all rows of the camera sensor. Temporal resolution of the video-based detection system in full-frame mode is limited by maximum frame rate and equals about 6.1ms. In actual configuration of the PC, error-free acquisition of the system at full frame readout was 100 FPS whilst in line-scan mode it reached 5000 lines per second. Therefore temporal resolution can reach $200\mu s$, which renders improvement factor 30.

The speed of acquisition and image processing is critical particularly for acorns during its movement across the feeder. Once they are released by the feeder, their motion is out of control until they fall down into the rotator's inlet. Considering geometry of the set-up, it has been estimated that the acorn of maximum length 4cm should appear in field of view of the camera for about 74ms with speed lower than 0.5ms^{-1}. Process of detection of the acorn can be divided into following stages:

- line acquisition,
- motion segmentation,
- event detection.

We have chosen two segmentation methods for evaluation: an easy-to-compute SAD (Sum of Absolute Differences); and computing-power-demanding motion detection based on adaptive GMM (Gaussian Mixture Model). Both methods produce signal that is provided to event detection subsystem.

A. Motion segmentation by means of SAD

SAD algorithm is usually applied to 2D images in order to asses changes in particular areas of a scene [12][13]. Due to low complexity, when applied to line-scan mode, it enables detection of events with very low latency. SAD signal S_i at time i, according to the formula (1) is a sum of absolute differences of pixels $P_i(n)$ within particular line that has N pixels. High values of the signal S_i notify significant change within field of view of the camera.

$$S_i = \sum_{n=0}^{N-1} |P_i(n) - P_{i-1}(n)| \tag{1}$$

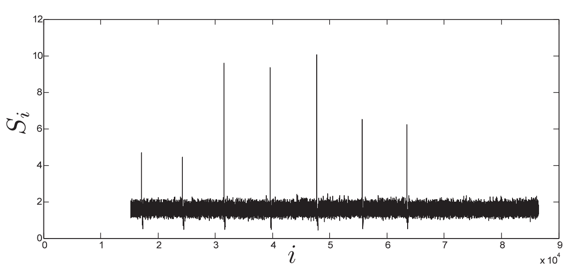

Fig. 1. Sequence of raw SAD signal

An example of signal corresponding to motion several acorns is shown in figure 1. Due to high level of noise existing in S_i signal, it is necessary to perform filtering and produce S_i^* signal which is the subject of further event recognition. Averaging of samples within window of length equal to 5 gave satisfactory result.

B. Motion segmentation by means of GMM estimation

Secondary examined method of motion segmentation was an algorithm based on adaptive estimation of background (GMM), also known as MoG (Mixture of Gaussians) [14][15]. The method is usually applied to frame-based images, but here it was used for scanned lines. An output of the algorithm is a binary image P_i. The image undergoes operations described in previous subsection but in this case, subtraction and filtering are needless. The input signal S_i^* is defined as sum of all N binary pixels $G_i(n)$ in a line which is te result of motion segmentation following estimation of gaussian mixture model.

The first stage of segmentation is adaptive computation of background model according to formulas (2-3). Value of each pixel in a line is treated here as a separate random process that can be described as a series of gaussian mixtures η_k defined by equation (3). Pixels $P_i(n)$, which values do not match existing distributions $(\mu_k, \sigma_k, \omega_k)$ ranked according to ω with constant threshold T, are considered components of foreground $G_i(n) = 1$ i.e. pixels of moving object, while the other are treated as background pixels $G_i(n) = 0$.

$$\rho(P_i) = \sum_{k=1}^{K} \omega_{k,i} \eta \left(P_i, \mu_{k,i}, \sigma_{k,i}\right) \tag{2}$$

$$\eta \left(P_i, \mu_{k,i}, \sigma_{k,i}\right) = \frac{1}{\sigma_{k,i}\sqrt{2\pi}} e^{-\frac{(P_i - \mu_{k,i})^2}{2\sigma_{k,i}}} \tag{3}$$

where: K - number of distributions: set to 3, P_i - pixel value at time i, $\rho(P_i)$ - probability of observing particular pixel value, $\sigma_{i,k}$ - variance of k-th mixture at time i, $\omega_{k,i}$ - weight of k-th mixture determined as $\omega_{k,i} = (1-\alpha)\omega_{k,i} - 1 + \alpha(M_{k,t})$, η - probability density function, $\mu_{k,i}$ - expectation at time i, α - learning rate: initially 1.0 and subsequently compressed down to 0.002 during first 500 lines, $M_{k,i}$ - equals 1 if value of actual pixel matches the model, else 0.

(a) fraction of input sequence (b) segemntation result for $T = 0.7$ (c) segemntation result for $T = 0.9$

Fig. 2. 2D reconstruction of input and segmented images from scanned lines.

GMM-based foreground segmentation does not introduce extensive latency but requires several iterations to initialize mixture of parameters: $(\mu_k, \sigma_k, \omega_k)$. The algorithm is resistant to changes of the surface of the feeder but precise adjustment of parameters and initialization stage is necessary after start-up. If the threshold T is to high, distortion can be observed like in figure 2c), where internal part of the object remains black. Value of T was set experimentally to 0.7. Complexity of the algorithm is significant but various acceleration techniques may be used in order to reach real-time requirements in target application: embedded into FPGA device [16][17] or running on GPU at 200 MPPS (Mega Pixels Per Second) [18].

C. Detection of events

Detector of the events, based on Finite State Machine, accepts S_i^* signal, and can remain in one of three states: ST_IDLE corresponding to neutral state, ST_ENTER signaled when the acorn enters field of view of the camera and ST_LEAVE fired when the seed leaves the area under surveillance. Duration of each state is rendered in figure 3 by the level of lightened solid line: middle - ST_IDLE, high - ST_ENTER, low - ST_LEAVE. Transitions between ST_IDLE and ST_ENTER are used to trigger events that notify entrance of the acorn into the field EV_ENTER. Transition from ST_LEAVE to ST_IDLE signals an exit EV_LEAVE of the seed from the field of view. Transitions are triggered when predefined levels of S_i^* signals are crossed: Th_1, Th_2, Th_3.The algorithm works seamlessly for both types of input signals S_i^*: produced by SAD or GMM-based segmentation. However, thresholds have to be modified in order to match their specific amplitudes. In the latter case, threshold reflect number of pixels that signal motion in the scanned line, whilst in case of SAD-based version they should correspond to averaged changes of brightness.

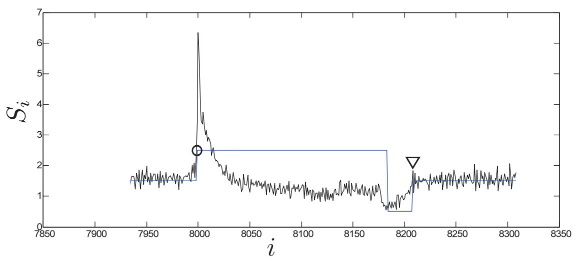

Fig. 3. FSM activity rendered over filtered SAD signal of single event: ○ marks EV_ENTER event, ▽ marks EV_LEAVE event.

IV. Results and Future Work

Accuracy of both methods of motion segmentation in detection of events has been expressed in accuracy of occurrences of EV_ENTER and EV_LEAVE events and intervals between them. The reference data has been annotated manually on the input sequence presented on figure 1. Relative error of interval measurement was 15.2% for SAD-based detection and 13.3% for GMM-based. Errors of EV_ENTER detection are $511\mu s$ and $133\mu s$ subsequently, whilst for EV_LEAVE are the following: $6044\mu s$ and $4978\mu s$. Estimated intervals exceed reference data. This can be explained by presence of shadows moving along with the object. Presented results show that GMM-based detection introduces lower errors and produces better signal for detection. SAD-based detection, is sensitive to changes of the surface of the feeder that are likely to occur during continuous operation.

Event detection, SAD and smoothing filter procedures were implemented in MATLAB. GMM-based segmentation was implemented using OpenCV library (including accleration on GPU). Custom OpenCL kernel was also tested on NVidia GT540M GPU device. Computing time of fastest kernels varied in range $100\mu - 200\mu s$ for line comprising region of 248 pixels. Temporal resolution of event generation in line-scan mode was increased when compared to full frame read-out. This enables faster vision-based feedback for control of the feeder. In future work it is planned to compare performance of detection depending on segmentation method during large-scale test. Characterisation of execution times, transfers and latencies for various lengths of scanned lines in order to find configuration filling requirements of real-time application will be done.

Acknowledgment

The work presented was supported by National Centre of Research and Development - of Republic of Poland (NCBiR) - grant no.: PBS3/A8/134/2015.

References

[1] Y. Tsividis, *Event-driven data acquisition and continuous-time digital signal processing*, Custom Integrated Circuits Conference, 2010 IEEE, San Jose, CA, pp. 1-8, 2010

[2] D. Kościelnik, M. Miśkowicz, *Time-to-digital converters based on event-driven successive charge redistribution: A theoretical approach*, Measurement, vol. 45, issue 10, pp. 2511-2528, 2012

[3] D. Rzepka and M. Miśkowicz, *Fast reconstruction of nonuniformly sampled bandlimited signal using Slepian functions*, 22nd European Signal Processing Conference, Lisbon, pp. 741-745, 2014

[4] M. Velasco, P. Marti and W. Bini, *Optimal-Sampling-inspired Self-Triggered Control*, Int. Conf. on Event-based Control, Communication, and Signal Proc., 2015

[5] T. Serrano-Gotarredona and B. Linares-Barranco, *A 128x128 1.5% Contrast Sensitivity 0.9% FPN 3µs Latency 4mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers*, IEEE Journal of Solid-State Circuits, vol. 48, no. 3, pp. 827-838, 2013

[6] B. Linares-Barranco and T. Serrano-Gotarredona, *Neuromorphic Sensors, Vision*, Encyclopedia of Comp. Neuroscience, Springer, pp 1-5, 2014

[7] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco and T. Delbruck, *Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output*, Proc. of the IEEE conf., vol. 102, no. 10, pp. 1470-1484, 2014

[8] T. Delbruck and M. Lang, *Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor*, Frontiers in neuroscience, vol. 7, article 223, 2013

[9] M. Jabłoński T. Serrano-Gotarredona, and B. Linares-Barranco, *High-Speed Serial Interfaces for Event-Driven Neuromorphic Systems*, Int. Conf. on Event-based Control, Communication, and Signal Proc., 2015

[10] L. A. Plana S.B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, and S. Yang., *A GALS Infrastructure for a Massively Parallel Multiprocessor* IEEE Design & Test of Computers, vol. 24, no. 5, pp 454-463, 2007

[11] Basler ace Users's manual for USB 3.0 cameras, version 04, 2015

[12] M. Gorgoń, P. Pawlik, M. Jablonski and J. Przybylo, *FPGA-based Road Traffic Videodetector*, 10th Euromicro Conf. on Digital System Design Architectures, Methods and Tools, Lubeck, pp. 412-419, 2007

[13] T. Sathish Kumar and S. Pavya, *Segmentation of visual images under complex outdoor conditions*, International Conference on Communications and Signal Processing, Melmaruvathur, pp. 100-104, 2014

[14] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246–252, 1999

[15] T. Bouwmans, F. El Baf, and B. Vachon. *Background modeling using mixture of gaussians for foreground detection - a survey*, Recent Patents on Computer Science, vol. 1, pp. 219–237, 2008

[16] E. Calvo-Gallego, S. Sánchez-Solano and P.B. Jiménez, *Hardware implementation of a background substraction algorithm in FPGA-based platforms*, IEEE Int. Conf. on Industral Technology, Seville, pp. 1688-1693, 2015

[17] T. Kryjak, and M. Gorgoń. *Real-time implementation of moving object detection in video surveillance system using FPGA*. Computer Science, vol. 15, issue 3, pp. 149–163, 2011

[18] M. Jabłoński, J. Przybyło, *Evaluation of MoG video segmentation on GPU-based HPC system*, Seventh ACC Cyfronet AGH users' conference: Zakopane, pp. 23–24, 2014