A Powerline-Tuned Camera Trigger For AC Illumination Flickering Reduction

Vito Renò, Roberto Marani, Massimiliano Nitti, Nicola Mosca, Tiziana D'Orazio and Ettore Stella

***Abstract*—Camera triggering represents an essential step for synchronizing artificial vision systems and can affect the quality of acquired images. In fact, a proper trigger signal is mandatory to synchronize in time both stand alone or multiple cameras covering large environments. In addition, indoor environments with artificial light sources can induce flickering noise in captured frames, affecting the performance of the algorithms that are usually executed to interpret a scene or perform various tasks. In this work we describe the design of an embedded system for camera triggering that can be employed to deal with such issues and remove flickering noise while capturing an image with the highest possible dynamic range. Experiments on real data show how the proposed trigger can be effectively employed in artificial vision systems.**

***Index Terms*—Embedded camera trigger, Dynamic range improvement, Flickering reduction**

I. Introduction

ARTIFICIAL Vision Systems (AVSs) usually need the cameras to be properly triggered in order to be highly efficient. The trigger, generally defined as a square wave, is used to define when cameras must capture a frame and how long the shutter has to be opened. Also, the same signal can control multiple sensors to acquire the images exactly at the same time. Synchronized frames are the basis for a large number of applications that need to integrate information coming from two or more cameras, especially high throughput applications like events recognition or scene understanding. Whenever an AVS setup is installed in an indoor environment, artificial lights can introduce problems that must be taken into account in the design stage.

Video signal processing in the general sense of scene analysis and understanding is a real life example that benefits from the application of computer vision techniques such as background subtraction, object recognition or semantic analysis to perform complex tasks [1], [2]. The attention here is focused on the first algorithms that are executed in a typical AVS directly on raw data, namely the low level ones. These procedures are employed to solve early stage problems and can be implemented on embedded systems, but might also produce unacceptable results when the images are affected by flickering noise, thus the necessity of developing also a proper embedded camera trigger generator. An example of flickering is reported in figure 1, where the effects of the noise are immediately observable. For example, in the technological platform described in [3] the first part of the processing stage is demanded to this type of procedures and illumination changes

The authors are with the Institute of Intelligent Systems for Automation of the National Research Council of Italy (CNR ISSIA), via Amendola 122 D/O, 70125 Bari (Italy) e-mail: (reno@ba.issia.cnr.it).

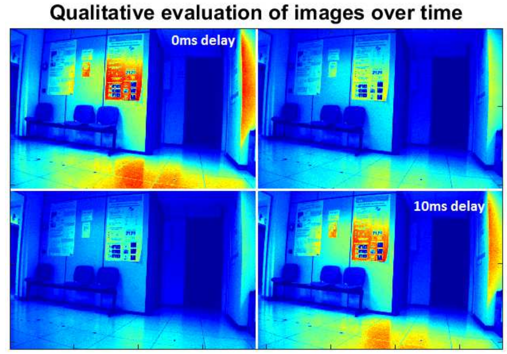

Fig. 1. The picture shows four frames of an indoor environment. Each pixel represents a luminance level normalized in the range $[0, 1]$ and rendered with the aid of a colormap: blue for zeros and red for ones. It is immediate to notice flickering effects in terms of luminance changes over time. The figure shows that brighter images can be acquired if the camera starts exposing its sensor properly.

due to the noise that can result in foreground false positives. Even if a large number of solutions have been proposed to estimate and remove flicker noise in post processing [4], [5], it is of course preferable to directly acquire the best images by properly triggering the cameras in accordance with light wavering. It is known that flickering issues can be avoided by setting the exposure time of the camera to half powerline period. However, the idea behind this work is driven by the need of further limiting the exposure time in order to avoid blur effects when fast moving objects are being captured, since many computer vision applications may require this feature. An example of such phenomenon is given in figure 2, where the same moving object has been captured varying the exposure time and visually showing the tradeoff between blurriness and brightness.

Camera triggering is a task perfectly suited for a dedicated embedded system and can significantly improve the quality of the images acquired by a camera, but when dealing with real life and wide spread solutions it is mandatory to take into account cost-efficiency constraints. For this reason, the scientific community started developing prototypes with the Arduino [6] electronic board: a low cost open-source platform based on easy-to-use hardware and software that has been efficiently employed in many engineering fields in the last years, leading to the implementation of both small applications and large projects. Its general purpose interface as well as the C language support with large availability of libraries make it a suitable solution for developing prototypes that can be easily interfaced with sensor networks, smart cameras or complex architectures [7], [8].

In this paper, a low cost Arduino-based embedded system that can generate a trigger signal tuned with powerline AC frequency is proposed. The purpose of this work is to provide a fast and lightweight solution able to capture the best image when fixed exposure time and sampling rate are set in an artificially lighted environment. Section II briefly describes the basic idea behind this Letter; section III reports its implementation on the Arduino board and discusses on experimental results. Conclusions and future works are finally described in the last section.

II. Methodology

Starting from the empirical evidence introduced with figure 1, it is necessary to define an algorithm able to suitably drive a camera that should be coded on an embedded architecture. The basic idea is to sample the powerline frequency, identify the zero crossing events, that correspond to the starting time of a new period, and generate the trigger signal when a peak is reached. With more details, the best trigger signal for the camera has to be tuned in order to center its exposure around the maximum of the actual light brightness.

With reference to figure 3, let us suppose that a period of the sinusoidal wave, which represents the AC powerline, can be limited by the zero-crossing points z_1 and z_2, denoted by the two red dots in figure 3(a). For the sake of simplicity, the following framework works with an AC powerline having frequency f_{AC} of $50\,Hz$. This signal reaches its maximum at a time $d_p = 5\,ms$ after the point z_1. In an actual implementation of any algorithm for zero-crossing recognition, the point z_1 is estimated with a time delay d_z. At the same time the AC powerline drives accordingly the light brightness, which flickers with a doubled frequency, but experiences a phase shift with respect to the $50\,Hz$ powerline. The two curves in figure 3(b) represent the expected brightness (blue solid curve) and the actual one (red dashed curve), which is delayed of d_l as effect of the lamps response and of the electronics, which are responsible for feeding the lamps.

Knowing the two time delays d_z and d_l, it is possible to set a proper trigger signal to control the camera. In the majority of cases, the trigger signal is a square wave defined accordingly with the TTL or RS-422 standards. In this case, the square wave must have a period equal to $k/(2 \cdot f_{AC})$, $k = 1, 2, \ldots$, to be synchronized with the AC powerline. Here k denotes the number of consecutive zero-crossing points of the AC powerline that are subtended by a period the trigger signal. Equivalently $(k-1)$ is the number of consecutive zero-crossing points that have to be counted after z_1 to close the trigger period and, thus, to enable a new trigger cycle. The properties of the trigger signal are summarized by the parameters in figure 3(c):

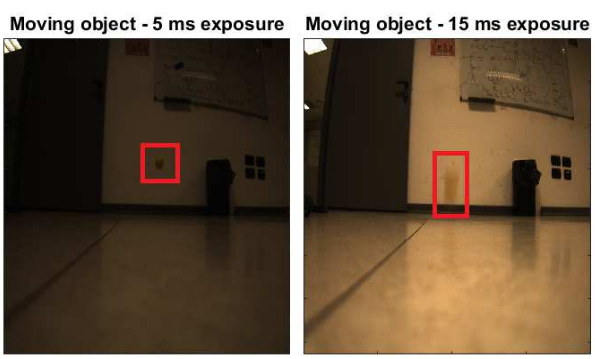

Fig. 2. The picture shows two frames that represent a fast moving object (namely a tennis ball) captured with different exposure time settings: 5ms on the left and 15ms on the right. Higher exposure time produces a brighter image but induces a motion blur effect. Lower exposure time guarantees sharper but darker images, thus the need of properly trigger the camera to maximize the image quality.

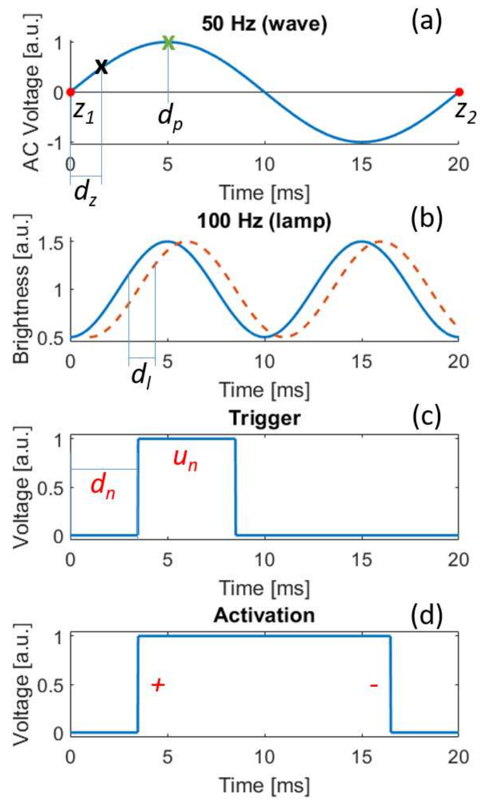

Fig. 3. This figure comprises the parameters involved in this problem. Briefly, in (a), a $50\,Hz$ sinusoidal wave that represents the powerline AC signal is plotted. In (b) the brightness of the lamp is shown, whose shape is similar to the wave plotted in (a), but with a doubled frequency. (c) and (d) represent the signals that can be generated with the algorithm, and that are employed to trigger the camera.

- 'd_n' represents the waiting time after z_1 before the activation of the trigger;
- 'u_n' defines the triggering signal uptime that is strictly related to the exposure time of the camera. Its value is clearly up-limited by the period of the trigger signal, i.e. by the value of the integer parameter k.

Theoretically, the value of d_n should be set equal to $d_p - 0.5u_n$, but in real cases this value needs to be adjusted to $d_n = (d_p - d_z - d_l) - 0.5u_n$, as effect of the presence of the actual delays discussed before.

Finally, figure 3(d) displays the activation signal which is in logical AND with the trigger square wave to enable the camera frame grabbing. In this plot, '+' and '-' set the level of the activation signal, thus starting or stopping the trigger.

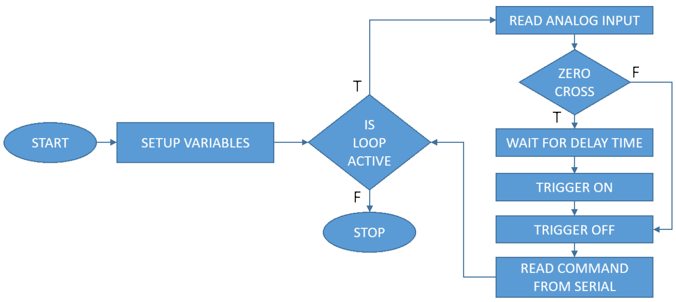

Fig. 4. Flowchart of the proposed approach. The "Setup Variables" block is executed only when the Arduino board is switched on and can be seen as the initialization step. Then, an execution loop (shown on the right) starts the analysis of the AC line to find zero crossings. The user can interact with a serial interface, setting the parameters described in section II.

The choice of the abovementioned parameters will be carried out by tuning the time delays in order to maximize several quality metrics defined on the acquired frames. A deeper insight on this experimental choice will follow in the next section.

III. Experiments and Results

The proposed framework has been implemented on an Arduino electronic board , which is set to behave in accordance with the flowchart in figure 4, enabling the embedded system to be effectively employed in distributed AVSs. As in typical Arduino applications, a setup block of instructions is executed once, before the main loop which is shown on the right of figure 4. At each iteration, the AC line signal is read in order to find the zero-cross time instant, i.e. when the analog input turns from negative to positive voltage values. Whenever a zero cross is found, the system waits for the delay time d_n and activates a trigger signal for a predefined time ($5\,ms$ of exposure in our implementation). The serial interface of the Arduino board is used to adjust the parameters during the execution. The source code that implements this logic can be found at https://github.com/vitoreno/TriggerAC/.

The proposed embedded trigger generator has been employed to trigger an AVT Prosilica GT-1920C camera at a fixed rate of $50\,Hz$, capturing high resolution frames (1920×1024 pixels). The camera is used to monitor an indoor corridor where artificial lights can induce flickering issues if the camera is not properly synchronized, leading to perceivable illumination changes that can affect the performance of video processing algorithms. According to the motivation exposed in the first section, the exposure time T_{exp} of the camera has been reduced to $5\,ms$. In this hypothesis, the global illumination of the scene can significantly vary with respect to the trigger starting time. The key concept of the experiments is to prove that the proposed trigger is able to synchronize a camera with the powerline frequency and capture the best image with respect to the global illumination level. To perform this task, two sets of 21 pictures of the same scene (one indoor and one outdoor) have been taken with a different delay, from $0\,ms$ to $10\,ms$. Each set of of captured images is represented by $I = \{I_{ds} | s = \Delta \cdot i, i = 0, 1, \ldots, 20\}$, where s represents the trigger start delay with respect to

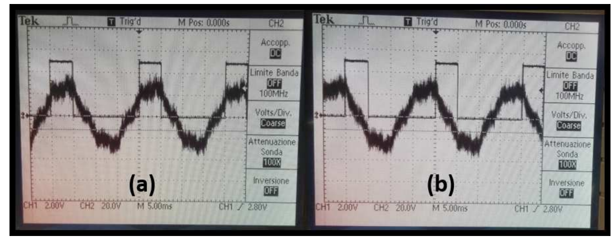

Fig. 5. Evidence of two signals – the sinusoidal AC wave and the triggering square wave – in the best case (a) and the worst one (b). On the left, the camera acquires the frame when the light source is around a local maximum; on the right the dual case, when the sine wave crosses zero.

the peak of the brightness and Δ has been set to $0.5\,ms$. Images with a delay $ds > 10\,ms$ are not taken into account because of the periodicity of the AC signal. Both qualitative and quantitative tests have been conducted considering I_{d0} as the reference frame and evaluating the luminance level as well as image quality metrics Peak Signal to Noise Ratio (PSNR), which is defined as $10\log_{10}(P^2/MSE)$, where P is the peak value of the signal (e.g. 255 for images with 1 byte pixel resolution) and MSE is the mean square error between the reference and test image. It is worth noticing that PSNR estimations suffer from summation errors. For this reason more robust metrics can be used: PSNR-Human Visual System (HVS) [9], its Modified version PSNR-HVS-M [10] and the Universal image Quality Index (UQI) [11]. The motivation behind the choice of these indexes is that they have been designed to overcome the problems introduced by traditional error summation methods like PSNR, whose performance can be ambiguous. HVS variants of the PSNR measurement take into account the perception of image quality from a human point of view and are more efficient than other known metrics. UQI models the quality of the tested image as a combination of three quantities: loss of correlation, luminance distorsion and contrast distorsion. As discussed previously, figure 1 shows qualitative results of the proposed solution: four different luminance images taken with a progressive delay in the time interval comprised between the zero crossing points z_1 and z_2 and rendered with an artificial colormap. In each picture, blue pixels correspond to the lowest luminance value while red pixels are associated to the highest one. It is immediate to notice that the first and the last frames – namely I_{d0} and I_{d10} – show a larger range of values if compared to the two frames in the middle – whose luminance values span from 0 to 0.5 – meaning that the dynamic range of the signal captured from the camera is halved. This phenomenon is due to the starting time in which the shutter has been opened.

Figure 5 gives clear evidence of the presence of a time delay between the zero-crossing point of the AC line signal and the position in time of the rising edge of the trigger which maximizes the dynamic range of the camera. As discussed previously, this delay d_n is the summation of two contributions: first, the algorithm locates a zero after it has been sampled by the Arduino board (d_z) and second, the lights introduce a phase shift in the sine wave when they are switched on (d_l). The experimental evidence can be seen in (a), where it is immediate to notice that the squared wave is not perfectly centered around

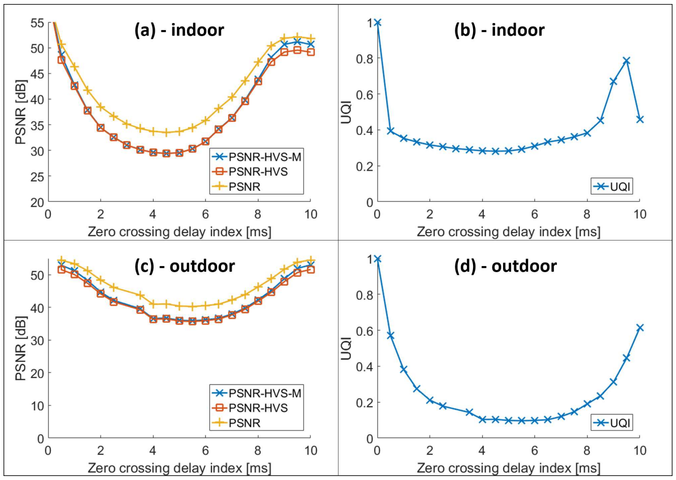

Fig. 6. Evolution of the PSNR- and UQI-based metrics on the chosen datasets.

the peak of the AC component of the powerline.

A quantitative analysis of image quality metrics is finally given in figure 6, where (a) and (b) are referred to the indoor setting, while (c) and (d) to the outdoor one. As stated beforehand, each frame is compared with the reference I_{d0} – namely the best captured one in terms of dynamic range – so that each computed value is always referred to the same image. In all the sub figures, the measured quality (y axis) is plotted against the time delay d_n that has been imposed to the trigger (x axis), confirming that the perceived quality is periodic and resembles the input sinusoidal wave. The alterations with respect to the sinusoidal input profile are mainly ascribable to the whole response of the lamps which is actually not linear and thus leading to the distortion of the AC line signal. Starting from the reference frame, the quality values gradually decrease until a local minimum obtained when the test image is taken with a delay of $4.5\,ms$. Then, the metrics increase again and a new period starts when the delay reaches $10\,ms$. The same behaviour both on indoor and outdoor images confirms the effectiveness of the proposed approach. A comment should be given about the last point of figure 6(b) that shows an important insight about the indoor reference dataset, namely that exists a time instant between $9.5\,ms$ and $10\,ms$ where the UQI should reach its peak value. Since the period of the lamp signal is $10\,ms$, the reference image I_{d0} is not exactly captured at the start of the period, meaning that the period has started slightly before. Finally, figure 7 shows consecutive frames captured when applying the proposed trigger signal. As expected, the flickering noise has been successfully filtered and the luminance values of the pictures are as high as possible in such setup.

IV. Conclusion and Future Works

In this paper, an innovative low cost embedded system for generating a camera trigger signal has been proposed. The prototype has been developed on an Arduino board and can be effectively adopted to capture the best image in indoor environments where artificial lights can induce flickering effects on the acquired videos. Experiments on real data have demonstrated the capability of such solution to acquire brighter images when the exposure time and the frame rate are fixed. Future works will be aimed to the improvement of this system to dynamically adjust the duty cycle of a trigger signal (i.e. control the exposure time) to establish a constant illumination level in higher frame rates setups without any constraints on the choice of the trigger frequency.

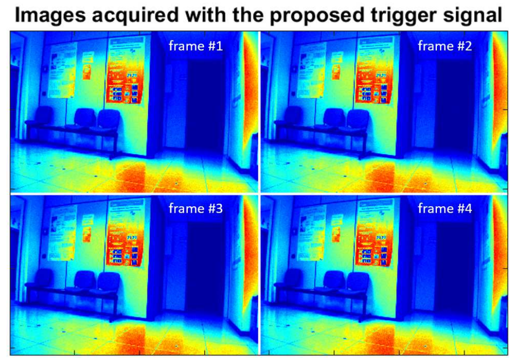

Fig. 7. Effects of the application of the Arduino-based trigger. All captured frames show the same dynamic range, as expected with the proposed implementation.

References

[1] V. Renò, R. Marani, T. D'Orazio, E. Stella, and M. Nitti, "An adaptive parallel background model for high-throughput video applications and smart cameras embedding," in *Proceedings of the International Conference on Distributed Smart Cameras*. ACM, 2014, p. 30. [Online]. Available: http://dl.acm.org/citation.cfm?id=2659059

[2] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple Kernel Learning for Visual Object Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, Jul. 2014.

[3] V. Renò, N. Mosca, M. Nitti, T. D'Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella, "A technology platform for automatic high-level tennis game analysis," *Computer Vision and Image Understanding*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314217300012

[4] Y. Yoo, J. Im, and J. Paik, "Flicker removal for CMOS wide dynamic range imaging based on alternating current component analysis," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 3, pp. 294–301, Aug. 2014.

[5] R. V. Savant and D. Pradhan, "Estimation of noise parameters for captured image," in *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, Mar. 2016, pp. 1029–1033.

[6] "Arduino - Home." [Online]. Available: https://www.arduino.cc/

[7] S. Sbîrnă, P. V. S⊘berg, L. S. Sbîrnă, and M. Coşulschi, "Sensor programming and concept implementation of a temperature monitoring system, using Arduino as prototyping platform," in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2016, pp. 848–853.

[8] S. Jindarat and P. Wuttidittachotti, "Smart farm monitoring using Raspberry Pi and Arduino," in *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, Apr. 2015, pp. 284–288.

[9] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli, "New full-reference quality metrics based on HVS," in *CD-ROM Proceedings of the Second International Workshop on Video Processing and Quality Metrics*, Scottsdale, USA, 2006.

[10] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, and V. Lukin, "On Between-Coefficient Contrast Masking of DCT Basis Functions," in *CD-ROM Proceedings of Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07*, 2007.

[11] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, Mar. 2002.