Asynchronous stroboscopic structured lighting image processing using low-cost cameras

Frank H. Borsato
Federal University of Technology - Paraná
Email: frankhelbert@utfpr.edu.br

Carlos H. Morimoto
University of São Paulo
Email: hitoshi@ime.usp.br

***Abstract*—Structured lighting (SL) image processing relies on the generation of known illumination patterns synchronized with the camera frame rate and is commonly implemented using syncing capable cameras. In general, such cameras employ global shutters, that exposes the whole frame at once. However, most modern digital cameras use rolling shutters, which expose each line at different intervals, impairing most structured lighting applications. In this paper we introduce an asynchronous SL technique that can be used by any rolling shutter digital camera. While the use of stroboscopic illumination partially solves for the line exposure shift, the phase difference between the camera and lighting clocks results in stripe artifacts that move vertically in the video stream. These stripes are detected and tracked using a Kalman filter. Two asynchronous stroboscopic SL methods are proposed. The first method, image differencing, minimizes the stripe artifacts. The second method, image compositing, completely removes the artifacts. We demonstrate the use of the asynchronous differential lighting technique in a pupil detector using a low-cost high-speed camera with no synchronization means, with the lighting running independently at a higher, unknown frequency to the application.**

I. Introduction

Video synchronization is required by a large range of applications, from stereo matching [1] to the study of animal behavior [2]. Synchronization is commonly achieved using specialized hardware [1], or by the introduction of special methods to process image features [3], or both [2], [4].

Typically, the role of synchronization is to allow controlled scene exposure and/or illumination such that the output follows an expected pattern. This is the case for differential lighting (DL), which demands that two consecutive frames be lit by sources at distinct spatial positions [5], [6]. This is also the case for time-multiplexed surface model acquisition systems, which encode pixels by illuminating the scene with different patterns over time [7], [8].

Structured lighting (SL) applications rely on cameras with explicit synchronization input and/or output that are typically more expensive than cameras that do not provide any synchronization means. Figure 1 shows a block diagram of a typical system. A solution presented in [9] synchronizes low-cost cameras and illuminators based on the dynamic estimation of the camera sensor exposure and number of lines. The solution exploits artifacts in the captured images lit by very short pulses of light. However, an important issue of the technique is the requirement of an external electronic circuit to allow the host computer to dynamically control the light triggering.

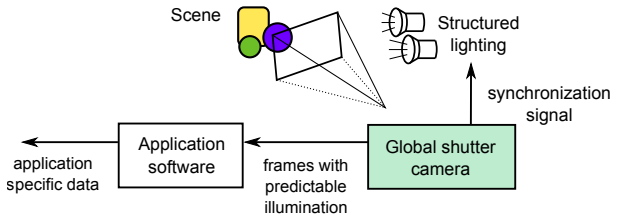

Fig. 1. Traditional setup of a structured lighting system. The frame synchronization data is used to trigger the lighting at the right instant, resulting in frames with predictable illumination patterns.

In this paper we introduce a method that predicts the artifacts produced in a rolling shutter camera lit by stroboscopic SL at frequencies close to the camera but not synchronized. No control over the illumination is assumed, simplifying both the hardware and the software. The illumination is completely decoupled from the computer system. Our software, instead of working to obtain images with a particular illumination, just keeps track of the artifacts produced in the images.

To use asynchronous lighting we developed a novel method to estimate how much the camera frequency diverges from the lighting frequency by accurately tracking image artifacts created by the lighting and rolling shutter lack of synchronization.

We focus on applications that exploit SL and might benefit from using cameras with no synchronization means. The state-of-the-art is based on specialized hardware [1], [10], still photography [11], and more recently, on software synchronization solutions [9]. Our method has great potential to be applied in pervasive and ubiquitous applications as it makes it possible to apply SL as if true synchronization is present. Additionally, by decoupling the illumination from the camera, simpler triggering circuits can be readily made from analog components, such as RC oscillators.

The rest of the paper is organized as follows. The next section describes the camera model associated with a non-synchronized high speed lighting source and its implications to the images produced. We also present two different ways of dealing with dual illuminated frames. Section III describes an implementation of the technique, followed by Section IV, that presents experimental results using a low-cost high-speed camera, including a practical application. Finally, Section V presents a brief discussion of the findings followed by Section VI, where we draw the conclusions.

II. Asynchronous camera-lighting model

Our camera model assumes a rolling shutter that exposes each line slightly shifted in time [9], [12]. Most digital cameras today use rolling shutters [13]. However, when it comes to SL applications, their overlapping behavior and time delay between each row exposure may introduce artifacts, such as partially illuminated frames and multiple light overlap, which occurs when a single frame receives light from different instants in time [14].

Figure 2 shows the rolling shutter capture model using asynchronous stroboscopic lighting. In the model, S represents the total number of sensor scanlines and N is the effective number of transferred (or visible) lines. The difference between S and N defines the invisible scanline range. The frame period is given by Δt, and the exposure length by Δe. The lighting pulse duration is given by Δs and the period by Δ_{clk}. Note that in Figure 2, the frame is the result of two snapshots taken by different stroboscopic light sources (strobes) at different instants, $s^{(j-1)}$ and $s^{(j)}$. The symbols k and k' represent the first and last scanlines lit by the first strobe and, k'' and k''', by the second strobe.

The interval defined by the pair (k, k') and (k'', k''') correspond to a stripe artifact, whose height is denoted by h_s. Scanlines in the stripe might be partially illuminated or receive light from two different strobes, which might impair their use [9], [15]. The stripe position depends on the phase difference between the camera and the lighting clocks [9]. As we assume no synchronization control over the illumination, Δ_{clk} is likely to diverge from Δt, resulting in a relative movement of the stripe when more than one frame is taken into account.

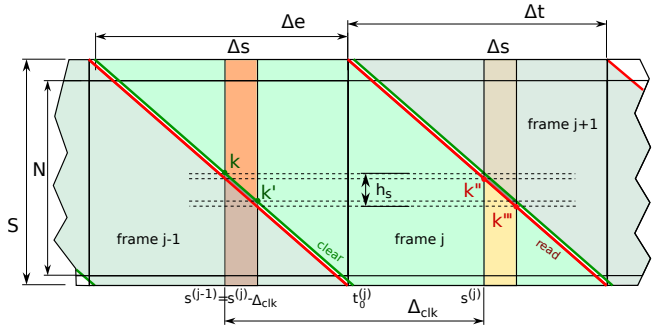

Fig. 2. Rolling shutter capture model using asynchronous stroboscopic lighting (Adapted from [9]).

Consider $t_0^{(j)}$ to be the readout time instant of the topmost scanline in frame j, and let $r_y^{(j)}$ be the readout time of an arbitrary scanline y, that can be computed as [12]

$$r_y^{(j)} = t_0^{(j)} + \frac{y}{S} \cdot \Delta t = t_0^{(0)} + \left(j + \frac{y}{S}\right) \cdot \Delta t, \tag{1}$$

then the time the scanline y in frame j starts to be exposed can be estimated by [12]

$$e_y^{(j)} = r_y^{(j)} - \Delta e. \tag{2}$$

A. Stripe characterization

We assume the nominal frequency of the camera is known (such as 30 or 60 Hz) so the lighting can be activated at approximately the same frequency. We also assume the camera exposure is close to the frame period and a triggered high-speed light source is in use. Under such conditions, the stripes can take three configurations, depending on the difference between the camera exposure Δe and the lighting clock period Δ_{clk}. Lets first assume they are equal, i.e., $\Delta e = \Delta_{clk}$. Let k'' be the first scanline subject to the second light pulse in a given frame j, $r_{k''}^{(j)} = s^{(j)}$. Thus, the last line to receive light from such strobe is given by k''' as follows [9]

$$k''' = k'' + \frac{S \cdot \Delta s}{\Delta t}. \tag{3}$$

Let k be the first scanline subject to the first light pulse in a given frame j, thus $e_k^{(j)} = s^{(j)} - \Delta_{clk}$. Replacing $e_k^{(j)}$ by its definition in (2), $s^{(j)}$ by $r_{k''}^{(j)}$, and Δ_{clk} by Δe, we have

$$\begin{aligned} r_k^{(j)} - \Delta e &= r_{k''}^{(j)} - \Delta e \\ r_k^{(j)} &= r_{k''}^{(j)}. \end{aligned} \tag{4}$$

Thus, $k = k''$ and $k' = k'''$. Therefore, the height of the stripe can be computed from the difference between k''' and k''. Applying (3) we have

$$h_s = k''' - k'' = \frac{S \cdot \Delta s}{\Delta t}. \tag{5}$$

Now, if we assume a lighting period shorter than the exposure, i.e., $\Delta_{clk} = \Delta e - \epsilon$ for $\epsilon > 0$, and keep the assumptions about k, k', k'', and k''' we obtain

$$\begin{aligned} e_k^{(j)} &= s^{(j)} - \Delta_{clk} \\ r_k^{(j)} - \Delta e &= r_{k''}^{(j)} - \Delta e + \epsilon \\ k - k'' &= \frac{\epsilon \cdot S}{\Delta t}. \end{aligned} \tag{6}$$

Therefore, the height of the stripe can be computed by taking the difference between k' and k''. Note that the distance between k' and k is the same from k''' to k'' (and depends on Δs). Thus

$$h_s = k' - k'' = k + \frac{S \cdot \Delta s}{\Delta t} - k''. \tag{7}$$

Replacing $k - k''$ from (6), we obtain

$$h_s = \frac{(\Delta s + \epsilon) \cdot S}{\Delta t}. \tag{8}$$

And lastly, if we assume a lighting period longer than the exposure, i.e., $\Delta_{clk} = \Delta e + \epsilon$ for $\epsilon > 0$, and keep the other assumptions unchanged, (6) will result in the same value with the opposite sign. Thus, the stripe height is given by the difference in the position of k''' with respect to k. Thus

$$h_s = k''' - k = \frac{(\Delta s + \epsilon) \cdot S}{\Delta t}. \tag{9}$$

We can join the three cases together, (5), (8), (9), to obtain the stripe height as follows

$$h_s = S \cdot \frac{\Delta s + |\Delta e - \Delta_{clk}|}{\Delta t}. \tag{10}$$

Note that the smallest stripe is obtained when $\Delta e = \Delta_{clk}$. However, if $\Delta_{clk} > \Delta t$, Δe cannot be increased to keep the stripe at minimum size. In this case, the stripe will be characterized by dark lines.

B. Asynchronous stroboscopic lighting

As disclosed in (10), the stripe height is lower bounded by the lighting period. The difference to the camera exposure only adds to the stripe height. Assuming scanlines at the stripe cannot be processed, the fraction of frames affected by artifacts is given by $p_c = (2 \cdot h_s + N)/S$. Thus, while it is possible to obtain frames without artifacts, in practice, N/S is close to unity as is p_c.

We propose two asynchronous stroboscopic lighting variations, frame differencing and frame compositing. The first allow us to exploit every available frame, despite the scanlines at the stripe. The second allow us to recover artifact-free frames at the lighting frequency.

1) Frame differencing: If a synchronized application requires the difference between consecutive frames, we can provide that difference with minimal losses, as we describe next. The frame differencing in the asynchronous stroboscopic lighting context can be regarded as a special case of the differential lighting (DL), as frames can suffer from multi-light overlap most of the time.

Besides the stripe height, to minimize the affected scanlines in the frame differencing, we also need to minimize the inter-frame stripe movement. To estimate the number of scanlines affected by the stripe and its movement, consider that the stripe position at frame 0 is $D_s(0)$. At the next frame, its position is given by $D_s(1) = D_s(0) + S \cdot (\Delta_{clk} - \Delta t)/\Delta t$. At the next, by $D_s(2) = D_s(0) + 2 \cdot S \cdot (\Delta_{clk} - \Delta t)/\Delta t$. We can calculate the position at any frame as follows

$$D_s(j) = D_s(0) + j \cdot S \cdot \frac{\Delta_{clk} - \Delta t}{\Delta t}, \tag{11}$$

thus, the translation of the stripe between any two adjacent frames is given by

$$D_s(j) - D_s(j-1) = S \cdot \frac{\Delta_{clk} - \Delta t}{\Delta t}. \tag{12}$$

When we compute the difference between adjacent frames, the lines affected by the stripe correspond to the stripe height plus the stripe translation, and accordingly

$$\hat{h}_s = h_s + S \cdot \frac{|\Delta_{clk} - \Delta t|}{\Delta t}, \tag{13}$$

where $\hat{h}_s$ are the lines affected in the difference image. Note that h_s is computed using (10) and already incorporates the effect of different clock frequencies in the stripe height.

The frame differencing can be used to robustly detect pupils by augmenting a DL-based pupil detector with the scanlines affected by stripes in real-time [16]. In Section IV-C, we show practical results of a pupil detector running at frame-rate using non-synchronized SL.

2) Frame compositing: The idea of the frame compositing is to join areas illuminated by the same strobe in successive frames. For example, in Figure 2, a frame J illuminated by the strobe at $s^{(j-1)}$ can be composed by scanlines $[0, k']$ from frame j, and $[k'', N]$ from frame $j-1$. Compositing has the potential to allow other modalities of SL to be used, such as photometric stereo [11]. Despite the lack of synchronization, artifact-free images can be obtained under certain conditions that are explained in what follows.

Let k'' be the first scanline subject to a given strobe in frame j, thus $r_{k''}^{(j)} = s^{(j)}$. In our model, this same strobe begins its exposure on the next frame at scanline k, and thus, $r_{k''}^{(j)} = e_k^{(j+1)}$. From now on, lets rename scanline k in frame $j+1$ to k_+ to avoid confusion. The difference between k_+ and k'' gives how many scanlines will not be able to recover the full illumination at every composed frame when the stripe is on the visible scanline range. According to our model, and applying (2) and (1), we have

$$\begin{aligned} r_{k''}^{(j)} &= r_k^{(j+1)} - \Delta e \\ k'' - k_+ &= S \cdot \left(1 - \frac{\Delta e}{\Delta t}\right). \end{aligned} \tag{14}$$

Thus, if the camera exposure matches the frame period, we can assure that each pulse of light was captured in full. Note however, that despite the lighting period Δ_{clk} was not accounted for in (14), scanlines might get illuminated by two light firings depending on the lighting frequency. This is the case of a multi-light overlap as depicted in Figure 2. To avoid that, we must ensure that $k'' \geqslant k'$. According to our model:

$$\begin{aligned} r_{k''}^{(j)} &= s^{(j)} \quad \text{and} \\ e_{k'}^{(j)} &= s^{(j)} - \Delta_{clk} + \Delta s. \end{aligned} \tag{15}$$

Replacing definitions (2) and (1) in (15), we obtain

$$\begin{aligned} r_{k'}^{(j)} - \Delta e &= r_{k''}^{(j)} - \Delta_{clk} + \Delta s \\ k'' - k' &= S \cdot \left(\frac{\Delta_{clk} - \Delta s - \Delta e}{\Delta t}\right), \end{aligned} \tag{16}$$

and as $k'' - k' \geqslant 0$, it holds that

$$\Delta_{clk} \geqslant \Delta e + \Delta s. \tag{17}$$

Therefore, the smaller the strobe length Δs, the higher the lighting frequency that can be employed.

III. Implementation

For both asynchronous stroboscopic lighting variations, frame differencing and compositing, a key information is the exact position and height of a stripe in the images due to the lighting and rolling shutter interaction. In Section II, we presented closed form equations to relate camera and lighting parameters to stripe unknowns.

In practice however, the camera parameters, such as the number of lines (S) and even the frame period (Δt) and exposure (Δe) might not be available. To overcome this limitation, the work in [9] presented a method which exploits the stripe size as a function of exposure to recover both S

and β. However, the stripe estimation is affected by non-linearities in the acquisition which compromise the non-continuous piecewise segmentation method employed.

In what follows, we present a new estimation method that is not biased by the camera gain settings and is particularly simple to implement, as it is based on linear regressions. Moreover, the residuals fill in the initial error covariance matrix of a filter used to accurately track the stripe over time (see Section III-C).

This section deals with the aforementioned practical implementation issues, as well as the uncertainty about the lighting frequency and how a stripe that is not always detectable can be accurately tracked.

A. Estimating the number of scanlines (S) and the relative lighting frequency

The number of scanlines of the camera (S) is used in the expected stripe size calculation and also to set the correct camera exposure. We propose computing S by estimating the time the stripe is hidden in the invisible scanlines. The idea is to record a long sequence of frames, enough to allow the stripe to pass the visible range several times. Then, estimate the stripe position in each frame, fitting several regressors of the form $\mathbb{B}(t) = a \cdot t + b$, where $\mathbb{B}(t)$ gives the stripe position given time t. The samples for a new linear regressor begin when the position drops while the time increased (or rises, depending on the stripe movement direction). Considering we have k regressors $\mathbb{B}_i(t) = a_i \cdot t + b_i, i = 1..k$, with $i = 1$ being on the earliest samples, we can estimate $\hat{S}$, the number of scanlines weighted by the ratio between the camera frequency and the lighting frequency, as follows

$$\hat{S} = \frac{1}{|\Pi|} \sum_{i \in \Pi} \frac{a_i \cdot b_j - a_j \cdot b_i}{a_j \cdot (i - j)}, \quad j \in \Pi, i > j \tag{18}$$

where $\Pi = \{t \in \mathbb{N} | 1 < t \leq k\}$ and j is the greatest index smaller than i in the set. Outliers, such as diverging angular coefficient regressors, can be filtered by excluding the index from the set Π. Figure 3 depicts the calculation process.

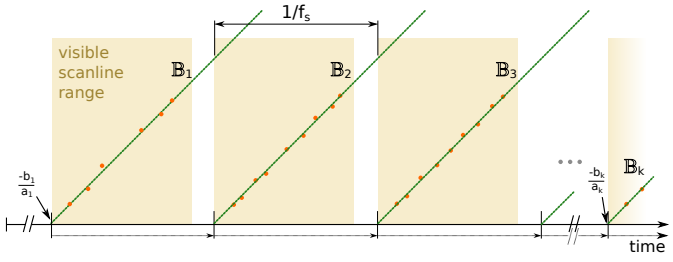

Fig. 3. Estimating S from the stripe position versus time.

The number of scanlines is calculated as follows

$$S = \hat{S} \cdot \hat{f}_r \tag{19}$$

where $\hat{f}_r$ is an estimate of the ratio between the camera frequency and the lighting frequency, and can be computed as follows

$$\hat{f}_r = \left(\frac{1/f_s \cdot 1/\Delta t}{1 + (1/f_s \cdot 1/\Delta t)} \right)^r \tag{20}$$

where r is 1 for positive $\mathbb{B}$ slopes and -1 otherwise, and $1/f_s$ is the period the stripe takes to travel all the scanlines (in seconds). The value of $1/f_s$ can be computed using the regressors by replacing a_j by $a_i \cdot a_j$ in (18) denominator.

B. Detecting the stripe position (D_s)

We assume the exposure is adjusted in order to produce a stripe composed by three distinct regions, a gradient to dark, a dark region which can ideally be one scanline wide, and a gradient from dark, with the darkest scanline belonging to the non-illuminated region. This stripe can be obtained adjusting the exposure as defined in [9]

$$\Delta e = \Delta_{clk} - \Delta s - \frac{1}{S} \cdot \Delta t. \tag{21}$$

The stripe position (D_s) is then approximated by the position of minimum intensity in a mean column image [9].

Adjusting the camera exposure in UVC compliant cameras is straightforward [17]. In such cameras, the exposure value $\mathbb{E}$ is related to the exposure length Δe by a constant $\beta = 0.1 \cdot 10^{-3}$. For the other cameras, the exposure can be in any arbitrary unit, and can also vary with the frame format. In such cases, the user might provide two exposure values that can be obtained by visual inspection. We will call them $\mathbb{E}_{vis}$ and $\mathbb{E}_{min}$, for the exposure values which render a visible stripe (as in (21)) and a minimal one, respectively. To obtain $\mathbb{E}_{vis}$, the user must set $\mathbb{E}$ to maximum and reduce it until a stripe composed by three distinct regions become evident (a gradient to dark, a dark region which can be one scanline wide, and a gradient from dark). The value of $\mathbb{E}_{min}$ depends on the relation between Δt and Δ_{clk}. If $\Delta_{clk} > \Delta t$ then $\mathbb{E}_{min}$ is given by the maximum $\mathbb{E}$. If the relation is inverted, starting from maximum, $\mathbb{E}$ must be reduced until a bright stripe vanishes. Note that in this later case, if β is known, the exposure can be automatically adjusted using Δ_{clk}, which can be estimated using (20) as follows

$$\hat{\Delta}_{clk} = \frac{\Delta t}{\hat{f}_r}. \tag{22}$$

To completely avoid user interaction, the image processing method proposed in [9] can be used to estimate β.

C. Tracking the stripe position with Kalman filtering

The stripe position detection is very simple and might not be available during some frame intervals, such as when the stripe enters the invisible scanline range. We assume the stripe moves at constant velocity, driven by the difference between the camera frame period and the lighting period. If this velocity is known as well as the stripe position in a given point of time, the position on a new frame can be determined ahead. We propose to use a Kalman filter to estimate the stripe position, as the detection can be notably noisy.

The Kalman filter is a linear minimum variance of error filter. It is a set of equations which might be used to estimate the position of a moving target from noisy measurements. It involves two steps, prediction and correction or update. The first step uses previous states to predict the current state.

The second step uses the current measurement to correct the state [18].

The stripe position estimation based on Kalman filtering can be formalized as follows. The state of the stripe at each time instance (frame) can be characterized by its position and velocity in the $\mathbb{R}^1$ domain. We assume constant velocity, as it is directly related to the drift in frequency between the camera and the lighting clocks. Let $D_s(j)$ be the position of the stripe in the frame at discrete time j and $D'_s(j)$ be the velocity. The state vector at time j can be represented as $X_j = [D_s(j)\ D'_s(j)]^T$.

The state vector at the next time frame $(j+1)$ is related to the current state by the system model as follows

$$X_{j+1} = FX_j + w \tag{23}$$

where F is the state transition matrix and w represents the system perturbation, which is normally distributed as $p(w) \sim \mathcal{N}(0, Q)$ where Q represents the process noise covariance matrix.

The stripe can be detected in the frame image when on the visible scanline range. We will call the detected stripe at time j as Z_j. As the detection is subject to noise, the measurement model is given by

$$Z_{j+1} = HX_j + v \tag{24}$$

where vector $H = [1\ 0]$ describes the state to measurement transformation and v represents the measurement uncertainty, which is normally distributed as $p(v) \sim \mathcal{N}(0, R)$ where R represents the measurement noise covariance matrix.

A Kalman filter usually operates in a two-step predict-update manner. The stripe is detected around a region predicted by the projected state (estimated stripe position based on previous Kalman filter state) and its uncertainty at time $j+1$. The detected stripe is combined with the estimation to produce the final stripe position.

Let X^-_{j+1} denote the a priori state estimate at time $j+1$ and X_{j+1} denote the a posteriori state estimate at time $j+1$. The prediction of the state X^-_{j+1} at time $j+1$ is the result of the previous state X_j, and the system model. The prediction is adjusted in the light of measurement data which result in X_{j+1}. The state X_{j+1} relies on both the system model (23) and measurement model (24). Associated to the state estimates are the error covariance matrices P^-_{j+1} and P_{j+1}, which characterize their uncertainties.

The Kalman filter prediction can be summarized as follows [18]:

$$X^-_{j+1} = FX_j \tag{25}$$

$$P^-_{j+1} = FP_jF^T + Q \tag{26}$$

where F is the state transition matrix and is updated every time the prediction is invoked to reflect the time past from the last frame received to the current, as follows

$$F = \begin{bmatrix} 1 & \bar{\Delta}t \\ 0 & 1 \end{bmatrix} \tag{27}$$

where $\bar{\Delta}t$ denotes the time past between the current frame and the previous (between discretized time $j+1$ and j), and might not be equal Δt.

If the stripe position on the projected state is greater than $\hat{S}$ in (25), it is updated to the difference of the two (not shown in equation). Accordingly, it is summed to $\hat{S}$ if smaller than zero. Note that this simple procedure accounts for the cyclic nature of the stripe position. The filter is updated by means of the three following equations [18]:

$$K_{j+1} = \frac{P^-_{j+1}H}{HP^-_{j+1}H^T + R} \tag{28}$$

$$X_{j+1} = X^-_{j+1} + K_{j+1}\left(Z_{j+1} - HX^-_{j+1}\right) \tag{29}$$

$$P_{j+1} = \left(I - K_{j+1}H\right)P^-_{j+1} \tag{30}$$

The initial conditions are brought as byproducts of the estimated camera parameters (Section III-A). The initial state X_0 is composed of the estimated stripe position for the current frame and the velocity derived from a median slope from linear equations $\mathbb{B}_i, i = 1..k$. The initial error covariance matrix is initialized with an estimated variance in the stripe calculation and stripe velocity

$$P_0 = \begin{bmatrix} s^2_{Ds} & s^2_{D's} \cdot \Delta t/2 \\ 0 & s^2_{D's} \end{bmatrix} \tag{31}$$

where s^2_{Ds} is the covariance on the stripe position estimation, considering the detected positions and the associated best fitting line. The velocity variance is given by $s^2_{D's}$, and is calculated as the variance of the first derivatives of $\mathbb{B}$ (Section III-A). We assume $R = s^2_{Ds}$ and the process noise matrix is initialized with zeros, because we assume $q = 0$, despite the effect of the estimated $\hat{S}$ in the prediction.

D. Frame compositing

Assuming that (17) holds, i.e., $\Delta_{clk} = \Delta e + \Delta s + \epsilon$ for some $\epsilon > 0$, we can easily build a composed frame J out of two adjacent frames, $j-1$ and j. Let $I^{(j)}(a, b)$ represent lines in the range $[a, b-1]$ from image j. Let k_- represent k in a previous frame $j-1$. The frame J can be composed as follows:

$$\begin{aligned} I^{(J)}(0, k) &= I^{(j)}(0, k) \\ I^{(J)}(k, k') &= I^{(j)}(k, k') + I^{(j-1)}(k''_-, k'''_-) \\ I^{(J)}(k', N) &= I^{(j-1)}(k'''_-, N). \end{aligned} \tag{32}$$

Applying our lighting period Δ_{clk} to (9), we have a stripe height $h_s = S(2 \cdot \Delta s + \epsilon)/\Delta t$. The stripe position $D_s(j)$ can be detected in frame j using convolution with h_s spatial support [9]. Then, k is computed as $D_s(j) - h_s/2$, $k' = k + S \cdot \Delta s/\Delta t$, $k'''_- = D_s(j-1) + h_s/2$ and, $k''_- = k'''_- - S \cdot \Delta s/\Delta t$. If the exposure is adjusted to maximum, i.e. $\Delta e = \Delta t$, then $k = k''_-$ and $k' = k'''_-$.

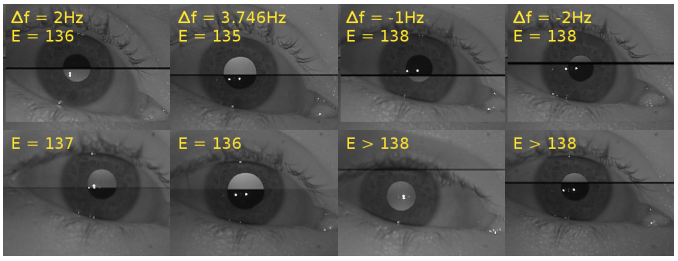

Fig. 4. Sample images captured at different lighting frequencies and corresponding exposures to produce a typical stripe, $\mathbb{E} = \mathbb{E}_{vis}$ (top), and minimal stripe $\mathbb{E} = \mathbb{E}_{min}$ (bottom).

IV. Experimental Design and Results

We performed experiments using a Sony PS3 Eye [19] camera, a low-cost and high-speed camera capable of capturing frames with 240 lines at 187 fps. This frequency was employed to be considered the most challenging. The lighting timing source was implemented using an Arduino board [20], a low-cost micro-controller board which provided an adjustable timer with a resolution of 62.5 ns. The SL employed two groups of infrared LEDs, spatially arranged on the camera axis and off-axis.

The results are organized in three parts. In the first we present our method to estimate the number of lines in the camera, which is of key importance to the stripe height computation and to the stripe tracking. Next we show our results related to the stripe tracking using a Kalman filter. Lastly, we show a practical application of the frame differencing, in which a pupil detector run at frame-rate using the non-synchronized illumination.

Except for the last, the experiments were performed considering four stroboscopic lighting frequencies, namely -1 Hz, -2 Hz, $+2$ Hz and $+2\%$, all with respect to the camera. While larger differences are possible, they are not typical, even for low-cost analog oscillators.

A. Stripe detection and number of scanlines estimation

The estimation process rely on the detection of the stripe position in a given frame. The stripe is detected in the frame column image, a vector containing the average of each image line. Calculating such vector is time consuming, thus we employ a sampled version, in which only 10% of the columns enter the sum. In our simplified version, the stripe is detected as the line with the lowest intensity, as far as it satisfies the condition of being at most λ the median intensity. We determined $\lambda = 1/3$ empirically, which results in some false negative detections, as the stripe is present but not sufficiently dark.

We adjusted the camera exposure to produce the minimal visible stripe for each lighting frequency, using the procedure described in Section III-A. Examples of the images obtained are shown in Figure 4.

Figure 5 shows the calculation for the first three seconds for each strobe period. In blue are the first order curves representing the stripe position at a given time and in red, the estimated S. We also added the actual estimated S, computed after from 26 to 39.5 seconds. Note that the S estimated using

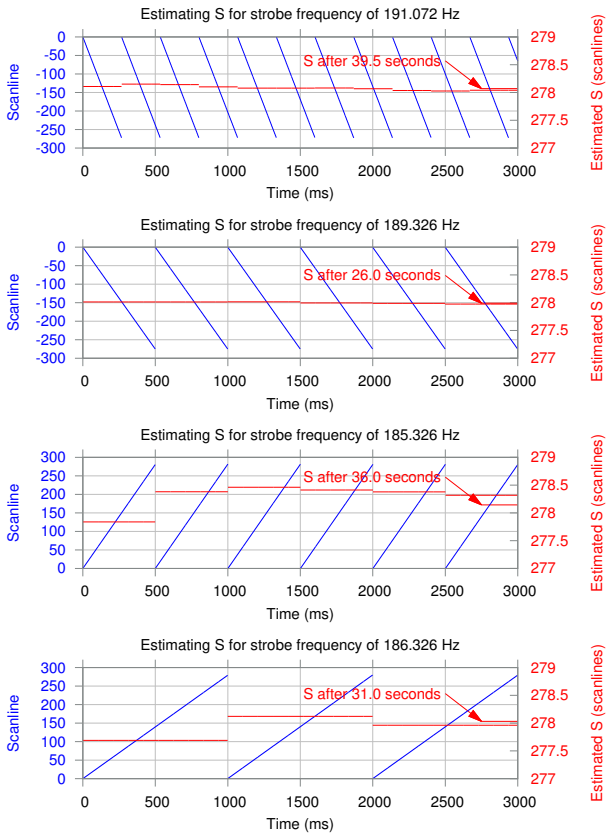

Fig. 5. Number of scanlines (S) estimated using the procedure described in Section III-A. Each graph shows a time excerpt of three seconds at different strobe frequencies with the estimated first order regressors.

only two or three regressions is a reasonable approximation of the value computed after several seconds.

B. Stripe position tracking

The stripe position tracking is an important step for the frame differencing and compositing techniques. The stripe position tracking must be accurate, precise, and lightweight. To meet these requirements, we proposed a Kalman filter. The filter is employed to deal with the noisy stripe detection and to estimate the position when the image-augmented detection is not available (prediction step).

In our particular experiment, the update step of the Kalman filter runs on 34.5% of the total frames captured. The filter prediction is exploited for $3/5$ of the visible scanlines, where the exposure is increased to reduce the stripe height the most, such that their effects are minimized in a frame differencing based asynchronous stroboscopic lighting application, as the one described next (IV-C).

Figure 6 shows the filtering result for the four lighting frequencies. For all cases, the gain drops steadily after some initial fluctuations (red points). The graphs also show the change in the estimated stripe velocity with respect to the calculated when the total scanlines were estimated (violet points). Note that the velocity at the end of more than 50 seconds do not differ more than 0.01% from the initial estimation. The graphs also show the difference between the predicted

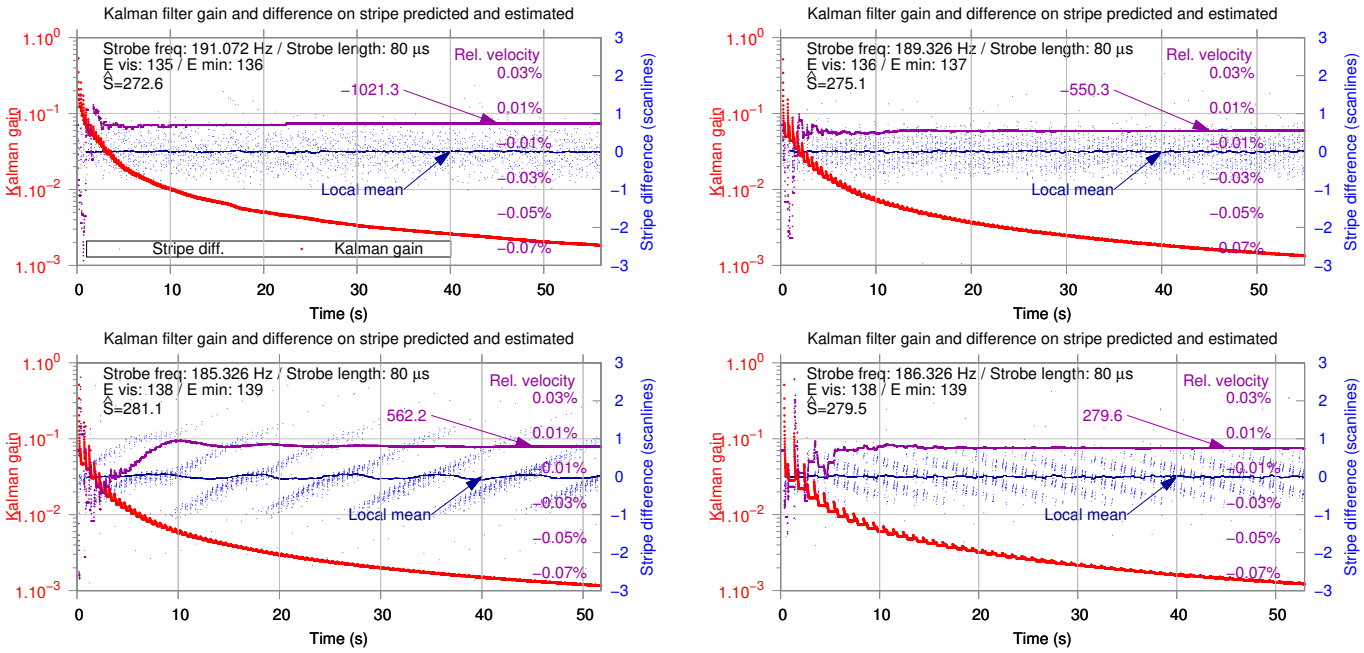

Fig. 6. Kalman filter gain, stripe velocity estimation (internal filter state) and difference between the stripe position as estimated by the filter and the calculated from the image.

stripe position with respect to the image-based estimation (blue points). A simple moving average (SMA) using a time window of one second where computed to highlight deviations from zero (dark blue points). The overall processing time average was 62.67 μs. When considering only the frames with stripe detection, the average rises to 186.56 μs. A density plot of the total processing time is presented in Figure 7. Note the bimodality, result of the prediction and update steps. Note also that more frames count on the prediction step when the stripe velocity is higher, as expected.

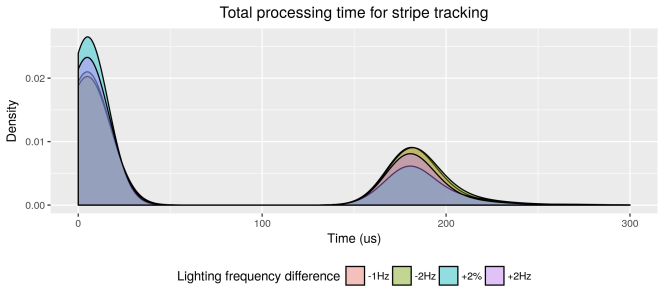

Fig. 7. Consolidated processing time to track the stripe position based on 40403 frames.

C. Sample asynchronous stroboscopic lighting application: Robustly detecting the pupil in eye images using frame differencing

When used to image the eye, our setup produces bright pupil images with the on-axis illumination and dark pupil images with the off-axis. The difference between two consecutive frames is used to detect pupil candidates as high contrasting areas. The best candidate is further refined to give more accurate results [15].

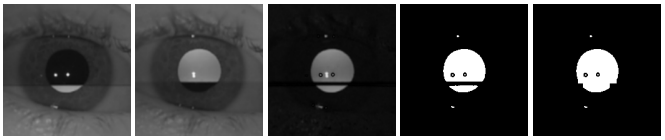

Fig. 8. Pupil candidate detection with frame differencing: region of interest excerpts for dual-illuminated pupils. a) frame j; b) frame $j+1$; c) frame difference; d) result of thresholding; e) corrected pupil candidate.

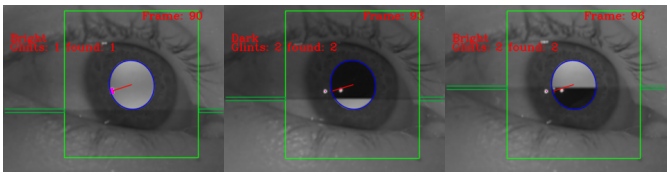

Fig. 9. Frame differencing: region of interest excerpts showing the pupil detection and the tracked stripe position.

When the stripe is over the pupil, it becomes dual-illuminated, as seen in Figure 4. As we devised in (13), the lines affected grow as the lighting and camera frequency diverge. We used a 2% frequency difference as representative of a worst case scenario, as simple analog circuits can stay down 1%. The resulting difference image is affected with a sliced pupil candidate, which is corrected with the calculated stripe height and tracked stripe position in real-time.

Figure 8 shows an example calculation using images taken at the nominal camera frequency of 187 Hz (187.325 Hz) with a 320 × 240 frame size, $\Delta s = 80$ μs and lighting at 191.072 Hz. Note that the lighting frequency is unknown to the application. Figure 9 shows the augmented pupil detection results.

V. Discussion

Our results show that our method accurately estimates the number of scanlines using a PS3 Eye Camera at 187 Hz. The difference from the ground truth was at most 0.052% lines, for a sensor with 278 scanlines. The Kalman filter used to track the stripe position also performed well. The internal gain of the filter (K) decreased to a mean value of $1.36 \cdot 10^{-3}$ ($SD = 2.67{\cdot}10^{-4}$). The internal velocity did not change on the last five seconds for 3 out of the 4 tests. The low internal gain and the stable internal velocity are strong evidences that the filter entered a steady state, despite the long intervals without updates (65.5% of the frames processed).

While we have successfully applied our asynchronous stroboscopic lighting frame differencing method in a SL application and despite the potential of the frame compositing to photometric stereo applications, other SL modalities, such as reconstruction based on temporal coded structured light [21] cannot be directly addressed by our technique, as it requires very short pulses of light that might not be feasible with current video projectors. Additionally, as our technique requires the camera exposure to be close to the frame period, the ambient illumination might pose a constrain to the use of the technique.

VI. Conclusions

In this paper we have shown how to exploit structured lighting (SL) in scenarios where the illumination is independent from the camera. SL typically relies on processing images with a known illumination pattern, and thus, is commonly implemented using syncing capable cameras. Despite recent advances that allowed rolling shutters to be used by software synchronization means, the added hardware needs to communicate with the software to allow synchronization, which poses a constrain to the technique. Our method exploits artifacts due to particular combinations of camera and lighting parameters to allow every single frame to be used. The artifacts are horizontal stripes that can be detected by simple thresholding most of the time. A Kalman filter was introduced to accurately track the stripe position. A constant velocity model was chosen, assuming a very slow drift between the camera and the lighting frequency. As the stripe movement is cyclic, we also presented an accurate method to compute the number of scanlines of the camera.

Two solutions to compute the illumination in a given frame are provided: frame differencing and frame composition. The first augment differential lighting applications with the scanlines affected by artifacts. We show in what conditions the least number of lines are affected by artifacts. In the frame composition, we create a virtual frame by joining portions of two adjacent frames. The particular conditions to allow this method are thoroughly discussed. In the first solution, camera frame-rate can be attained and, in the second, frames are composed at the frequency of the lighting.

We have demonstrated that the frame differencing can be used in a differential lighting pupil detector that typically requires explicit synchronization. The application relies on the active illumination to robustly detect the pupil. It uses a low-cost high-speed camera that reaches 187 Hz and an independent asynchronous stroboscopic lighting.

Acknowledgements

This research was supported by Fundação Araucária (DINTER project UTFPR/IME-USP) and by the São Paulo Research Foundation (FAPESP), grant #2016/10148-3.

References

[1] P. Agrafiotis, A. Georgopoulos, A. Doulamis, and N. Doulamis, "Precise 3D measurements for tracked objects from synchronized stereo-video sequences," *Lecture Notes in Computer Science*, pp. 757–769, 2014.

[2] B. Evans, B. Parslow, D. O'Carroll, and S. Wiederman, "Quantifying asynchrony of multiple cameras using aliased optical devices," in *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*, Nov 2015, pp. 567–572.

[3] A. Whitehead, R. Laganiere, and P. Bose, "Temporal synchronization of video sequences in theory and in practice," in *WACV/MOTIONS'05 Volume 1.*, vol. 2. IEEE, 2005, pp. 132–137.

[4] L. Hou, S. Kagami, and K. Hashimoto, "Frame synchronization of high-speed vision sensors with respect to temporally encoded illumination in highly dynamic environments," *Sensors*, vol. 13, pp. 4102–4121, 2013.

[5] C. H. Morimoto, D. Koons, A. Amir, and M. D. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing*, vol. 18, no. 4, pp. 331–335, 2000.

[6] Y. Ebisawa, "Improved video-based eye-gaze detection method," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 4, pp. 948–955, 8 1998.

[7] H. Gaertner, P. Lehle, and H. J. Tiziani, "New highly efficient binary codes for structured light methods," in *Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology*, vol. 2599. International Society for Optics and Photonics, 1996, pp. 4–14.

[8] E. Horn and N. Kiryati, "Toward optimal structured light patterns," *Image and Vision Computing*, vol. 17, no. 2, pp. 87–97, 1999.

[9] F. H. Borsato and C. H. Morimoto, "Building Structured Lighting Applications Using Low-Cost Cameras," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Oct 2017, pp. 15–22.

[10] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 438–446.

[11] Y. Zhang, G. M. Gibson, R. Hay, R. W. Bowman, M. J. Padgett, and M. P. Edgar, "A fast 3d reconstruction system with a low-cost camera accessory," *Scientific Reports*, vol. 5, 2015.

[12] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich, "Synchronization and rolling shutter compensation for consumer video camera arrays," in *CVPR Workshops 2009.* IEEE, 2009, pp. 1–8.

[13] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *Computational Photography (ICCP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–8.

[14] QImaging, "Rolling Shutter vs. Global Shutter," Tech. Rep., 2014. [Online]. Available: https://www.qimaging.com/ccdorscmos/pdfs/RollingvsGlobalShutter.pdf

[15] F. Borsato, F. Aluani, and C. Morimoto, "A Fast and Accurate Eye Tracker Using Stroboscopic Differential Lighting," in *2015 IEEE ICCVW*, 2015, pp. 110–118.

[16] F. H. Borsato and C. H. Morimoto, "Design of a Low Cost Pupil Detector Using Asynchronous Differential Stroboscopic Lighting," in *WEGA - Workshop of Eye Gaze Applications. IHC 16*. ACM, 2016.

[17] USB Implementers Forum, "Universal Serial Bus Device Class Definition for Video Devices," p. 130, Jun 2005.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[19] C. Sony, "PlayStation Eye Camera," 2014. [Online]. Available: http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html

[20] Arduino.cc, "Arduino leonardo webpage," [Online, 10/jun/2017], 2017. [Online]. Available: http://arduino.cc/en/Main/arduinoBoardLeonardo

[21] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern Recognition*, vol. 43, no. 8, pp. 2666–2680, 2010.