# Fast normalized cross-correlation for measuring distance to objects using optic flow, applied for helicopter obstacle detection

Christophe Viel, Stéphane Viollet

# Fast normalized cross-correlation for measuring distance to objects using optic flow, applied for helicopter obstacle detection

Christophe Viel*, Stéphane Viollet*
Aix-Marseille Université, CNRS, ISM, Marseille cedex 09, France
* Email: christophe.viel,stephane.violet@univ-amu.fr

**Abstract**

The fast normalized cross-correlation (NCC) calculation method presented here features low computational requirements, which makes it suitable for being implemented in real time onboard micro-controllers with very few computational resources. This method was adapted for making distance measurements, using a high speed optic flow sensor operating at 20m/s. An application of this study is to develop a proof of concept of an innovative optic flow sensor fixed at the tip of an helicopter's blade (from Airbus Helicopter) to measure the distance to various obstacles (wall, cliff...) in the azimuthal plane (rotor plane) during hovering flight. Due to its small size, this sensor developed in this paper can also be also adapted for Micro aerial vehicles (MAVs). This method of calculation requires less memory than the reference method, at the expense of some extra arithmetical operations, but it is still significantly lighter than the classical NCC method. An algorithm for implementing this method in real-time robotic applications is presented. Experimental results confirm the efficiency of this highly time-saving method.

*Keywords:* Optic flow sensor, normalized cross-correlation, time delay, distance estimation, real-time calculation

## 1. Introduction

Obstacle detection for aircraft, and in particular for rotary wing aircraft moving at low altitude, is a subject which has not yet found a commercially viable technical-operational response for the greatest number of operators. This is an ongoing concern because collision with obstacles, particularly for main rotor and tail rotor helicopters, is still a significant cause of accidents. Thus, this study proposes the implementation of optic flow sensors fixed at the tip of the blades, to measure the optical flow in order to estimate the distance to a frontal, lateral or even rear obstacle (wall, cliff, post, tree...). However, due to the extremely fast dynamic of an helicopter's blade, this sensor must be the lightest and the smallest as possible, requiring greatly miniaturized system integration processes, power supplies, and sensors.

This problems of miniaturization are similar to constraint of Micro aerial vehicles (MAVs). Micro aerial vehicles are insect-sized robots which can be used for remote observation purposes in hazardous or small-scale environments. Thanks to their size, they provide useful means of performing tasks such as inspecting inaccessible areas, search-and-rescue operations, and greenhouse monitoring. Contrary to larger unmanned aerial vehicles (UAVs), this miniaturization often means replacing traditional sensors such as radars, laser rangers, motion sensors, infrared rangers, sonar and GPS systems as means of self-localization, and turning to biological sources of inspiration. MAV drones' sensors therefore often consist of RGB cameras, which provide highly energy efficient and versatile sensing performances. The many variables which can be monitored based on the optic flow include the presence of objects and obstacles and the occurrence of motion in the environment. The stereo vision implemented in previous studies using two cameras gives reliable distance [6], velocity estimates [10, 15, 16] or obstacle avoidance [18]. However, due to the strict limitations imposed on the energy, sensing, and processing resources of MAV drones' microprocesors, even the most efficient stereo vision methods are too computationally demanding to be implemented on-board on their microprocessors. More efficient stereo vision algorithms and sensors based on the optic flow (OF) have therefore been developed, which require much fewer/lighter resources. Bio-inspired OF sensors provide low-cost, lightweight sensors which are cheaper and lighter than traditional sensors, making it very interesting for aircraft and mini-UAVs. The OF can be used to measure several parameters such as a robot's velocity [13, 15], its altitude [5, 17] and to detect the distance to obstacles [16, 19, 7]. Contrary to methods based on stereo vision using two cameras, OF methods require only a few pixels: it often suffice to measure just two signals in order to obtain the information required for navigating safely. OF measurements can be challenging when it comes to measuring the time elapse between two fast signals at speeds greater than 10m/s. One of the most widely used methods here is the classical normalized cross-correlation (NCC) method [9, 2, 8, 1, 11]. A reference window is defined on the first signal and compared with windows in the second signal delayed by a fixed time lag: a cross-correlation coefficient is then determined between the two windows. After calculating the cross-correlation coefficient for several different time delay, the maximum correlation obtained between the two signals is defined as the current delay. [4] showed that the normalized cross-correlation method gives bet-

ter time delay estimation than phase-shift [3] or pulse counting methods [13, 21, 16]. However, the cross-correlation method is rather time-consuming and requires many computations to obtain the maximum correlation, which is a significant drawback in the case of real-time applications in which it has to be run on-board a MAV microprocessor, for example. Because of this disadvantage, the pulse counting method is still mainly used in small embedded optic flow sensors although it gives less accurate results. Some authors such as [1, 11, 24, 12] have presented faster methods of calculation for finding maximum correlations. The method proposed by the latter authors [1, 12] consists of interpolating the cross-correlation curve with a parabolic interpolation or a SINC function. A maximum correlation can be obtained by calculating only three correlations and extrapolating the maximum correlation from them. However, the results obtained in this way are highly sensitive to noisy signals, which affect the interpolation. In [24] a fast algorithm involving no multiplications was presented for calculating normalized cross-correlations (NCCs). Since multiplications require more computational time than additions, they were replaced in the method presented by [24] by logical operations giving just a slight increase in the false positive rate. Some redundant calculations are still performed, however. Thus, [11] developed a fast NCC method based on the use of sum tables in view of the fact that many calculations tend to be redundant because of the exhaustive search conducted on the windows compared with the reference windows. The correlations calculated in the overlapping region between two closely neighboring time delay values therefore becomes redundant. The sum table method was therefore developed in order to calculate cross-correlations faster. The main disadvantage of the latter method is that the number of sum tables to be memorized is too large for embedded systems to cope with, since they have only a small memory.

Nowadays, it does not exist commercial system in the industry that can inform the pilot about the position and distance of surrounding obstacles. This paper deals with an innovative optical system based on the use of a custom-made sensor fixed at the tip of a rotating helicopter's blade to measure the distance to an obstacle during hovering flight. The method developed here is inspired from OF sensor developed for Micro-Air vehicles to obtain a cheap, compact and light sensor to be fixed on any helicopter's blade. Pure translational OF allows to measure optically distance to objects by making some assumption about the translational speed (see [20]). Our OF-based principle benefits from the high tangent speed of an optical sensor placed off-centered from the axis of the helicopter's main rotor. To obtain the required performance, the computational time of the NCC must be reduced and be implemented in real time to obtain more accurate results than using pulse counting method. For this purpose, a moving cross-correlation coefficient has been developed to reduce the time required to calculate usual NCCs. The idea is similar here to that proposed in [11] for reducing redundant calculations, but fewer sum tables are required, which keeps the memory requirements small, thus meeting the demands of embedded OF sensors. The method of calculation presented here involves some extra basic arithmetical operations in comparison with the reference method, but not uses root square operation making it faster to compute, and the method is still significantly lighter than the classical method of calculating the NCC. The OF sensor developed here can be used to assess either the distance to an obstacle, or the vehicle's velocity if the distance to the obstacle is known.

This paper is organized as follow. In Section 2, the problem of using optic flow to determine the distance based on time lag data is presented. The Section 3 exposes the parameters required to calculate the distance to obstacle and the size of the smallest obstacle which can be detected. The theory of normalized cross-correlations used here to assess the time delay is described in Section 4.1. A method to reduce the number of NCC to be calculated is presented in Section 4.2. The Section 5 describes the new method developed for calculating the NCC faster. The implementation of the algorithm on which this method is based and comparisons with the reference method [11] is described in Section 5.3. Experiments in which the method was tested and the results obtained are presented and discussed in Section 6. Section 7 concludes the paper.

## 2. Optic flow

Optic flow sensors are usually composed of several (at least two) photosensors measuring motion. The visual motion sensor designed and developed in this study for measuring the distance to contrasting objects is a small, lightweight 2-pixel motion sensor. A defocused lens placed in front of the two photosensors determines the inter-receptor angle $\Delta\phi$, defined as the angle between the two optical axes of two adjacent photosensors. Defocusing the lens makes the angular sensitivity of each photosensor follow a Gaussian curve. The inter-receptor angle therefore depends on the photosensor's pitch and the lens defocusing ratio. A contrasting object is detected by measuring the delay between two photosensor signals $S_1$ and $S_2$, as

$$\omega = \frac{\Delta\phi}{\Delta t} \tag{1}$$

where $\omega$ is the angular speed (or the Optic Flow (OF)) and $\Delta t$ is the time delay between $S_1$and $S_2$.

The optic flow $\omega$ can be divided into two components:

$$\omega = \omega_{trans} + \omega_{rot} \tag{2}$$

where $\omega_{trans}$ is the translational optic flow and $\omega_{rot}$ is the rotational optic flow. By definition, $\omega_{rot} = \Omega_r$ where $\Omega_r$ is the relative angular velocity between the sensor and oncoming object.

Since $\omega_{trans}$ is by definition an angular speed, a contrasting object can be detected only if there exists a relative velocity $V_t$ between the sensor and the object. The OF $\omega_{trans}$ of a moving object such as a wall or a bar subjected to a purely translational velocity $V_t$ can be defined as follows [19]

$$\omega_{trans} = \frac{V_t}{D}\sin^2(\alpha) \tag{3}$$

where $D$ is the distance to any surface, and $\alpha$ is the orientation of the sensor with respect to the surface. Since the sensor is

placed in front of the surface, or if we are dealing only with a small object placed in front of the sensor, $\alpha$ can be taken to be equal to $\alpha = \frac{\pi}{2}$. Thus, the relation between the velocity $V_t$ and the optic flow $\omega_{trans}$ becomes

$$\omega_{trans} = \frac{V_t}{D} \tag{4}$$

Then, combining (1) and (4) gives

$$D = \frac{V_t}{(\omega - \omega_{rot})}, \tag{5}$$

$$= \frac{V_t}{\left(\frac{\Delta\phi}{\Delta t} - \Omega_r\right)}. \tag{6}$$

If the velocities $V_t$ and $\Omega_r$ are assumed to be known by the sensor (and/or $\Omega_r$ is assumed to be equal to zero), the distance $D$ can be therefore determined by measuring $\Delta t$. Distance to object estimation is therefore based here on the time delay measured between two photosensors' output signals. It has been previously established [4] that the normalized cross-correlation method gives more accurate time delay estimates, than phase-shift or pulse counting methods. The cross-correlation method is described in Section (4).

**Comment:** In (6), the distance $D$ is calculated using the known velocity $V_t$ and the value of $\Omega_r$. However, the velocity $V_t$ can also be estimated if the distance $D$ and the value of $\Omega_r$ are known. This method can be applied to a case in which it is required to determine a ground vehicle velocity, where the distance $D$ between the sensor and the ground is known and the value of $\Omega_r$ is assumed to be null.

## 3. Parameters of distance estimation

### *3.1. Definition of distances calculated*

The optic flow sensor is fixed at the blade tip of the helicopter to measure the optical flow generated by the tangential speed of the sensor (see Figure 1). The proposed methods in this paper is developed for the hovering flight condition.

As explained in Section 2, the distance $D$ between the sensor and a potential obstacle can be evaluated by measuring the time delay between the two visual signals provided by the photodetectors of the optic flow sensor. The estimation of this delay requires a minimum number of samples $M$ (using for example cross-correlation method described in next sections), corresponding to a recording time $T_r$. However, during the measurement time $T_r$, the rotor continues to rotate, involving the distance $D$ and so the measured delay change within the $M$ measurement (illustrated in Figure 1): the estimated delay is therefore an average value of the delays.

Due to the number of measurement $M$ required to detect the time delay between the two signals measured by the sensor, all distances inside a defined cone are averaged in only one measurement $D$. This cone can be theoretically expressed as follows:

$$\alpha = \frac{2\pi}{N_t} \tag{7}$$

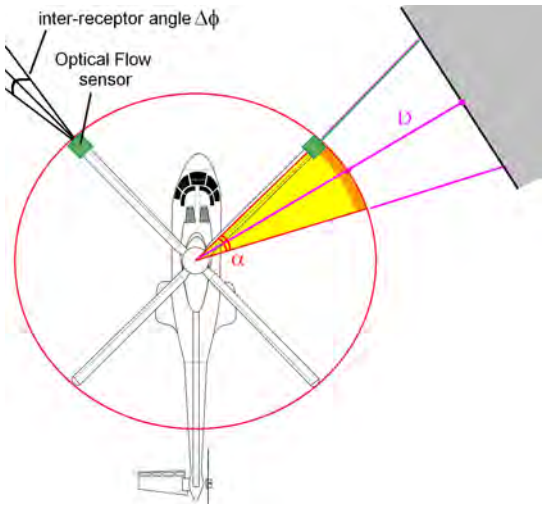


Figure 1: An optic flow sensor (in green) is placed at the tip of an helicopter blade. The yellow area is used to evaluate a distance using measurement made inside a cone of chosen size $\alpha$. Measured distances between the sensor and the surface inside the yellow area are averaged into a single value $D$.

where $N_t$ is the chosen number of sector around the helicopter such that all the measurement made in this sector are used to evaluate an unique average distance $D$ (yellow area in Figure 1). $M$ can so be calculated as follows:

$$M = \text{floor}\left(\frac{F_a}{N_t * \Omega_{Hz}}\right) \tag{8}$$

where $\Omega_{Hz}$ is the rotation velocity at hovering flight in Hz, $F_a$ is the acquisition frequency and the $floor()$ function gives the nearest integer up.

The value of $N_t$ must be chosen by taking into account the constraints described in Section 3.2.

### *3.2. The smallest obstacle detected*

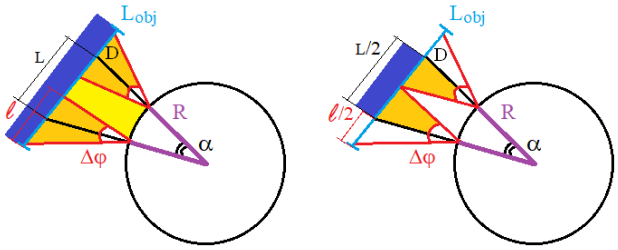


Figure 2: Parameters to evaluate the size of the smallest obstacle which can be detected with $N_t$. Left: largest obstacle which can be seen by the sensor considering the cone of vision $\Delta\phi$. Right: smallest obstacle which can be seen by the sensor considering the cone of vision $\Delta\phi$.

Let define $R$ the distance between the sensor and the center of rotation. Let define also an obstacle at a distance $D$ from the sensor and thus a distance $D+R$ from the center of rotation (see Figure 2). In a first step, we can define the length $L$ seen by the sensor when the blade performed the angular displacement $\alpha$ as follows:

$$L = 2\tan\left(\frac{\alpha}{2}\right)(D + R)\,. \tag{9}$$

In a second step, we can consider also the length $\ell$ seen at a distance $D$ by the cone of vision of the sensor $\Delta\phi$ at each instant defined by:

$$\ell = 2\tan\left(\frac{\Delta\phi}{2}\right)D. \tag{10}$$

If we make the assumption that one photo-sensor sees only one object at a given time, an obstacle can be detected if it occupies the overall viewing angle $\Delta\phi$. Moreover, since all measured distances inside $L$ are averaged, the object can be detected if all the distances occupy at least one half of the viewing angle $\alpha$ (*i.e*, $\frac{1}{2}L$). Finally, an object can be perfectly detected at a distance $D$ if it features a size equal or larger than $\frac{1}{2}L + \ell$.

Thus, the minimum size $L_{\min}$ can be expressed as follows: or

$$L_{\min} = \frac{1}{2}L + \ell = L_{\min} = \tan\left(\frac{\alpha}{2}\right)(D+R) + 2\tan\left(\frac{\Delta\phi}{2}\right)D. \tag{11}$$

The minimum length $\frac{1}{2}L$ seen by the cone $\alpha$ must be larger than the length $\ell$, *i.e*. $\frac{1}{2}L \geq \ell$. Thus, by considering $\frac{1}{2}L \geq \ell$, we have:

$$\alpha \geq 2\text{atan}\left(2\tan\left(\frac{\Delta\phi}{2}\right)\frac{D}{D+R}\right). \tag{12}$$

Since $\Delta\phi$ is small and $\frac{D}{D+R} < 1$, a general condition can be defined such that $\alpha \geq 2\Delta\phi$. From (7), one can define a condition on $N_t$ such that:

$$N_t \leq \text{floor}\left(\frac{\pi}{\Delta\phi}\right). \tag{13}$$

where the $floor()$ function gives the nearest integer up. From (13), $N_t = \text{floor}\left(\frac{\pi}{\Delta\phi}\right)$ allows to obtain the maximum accuracy and to detect the smallest obstacle possible. A smaller value of $N_t$ leads to a larger number of measurement $M$, which is interesting when the acquisition frequency is low (see (8)). However, a too small $N_t$ involves a reduction of the smallest obstacles which can be detected due to the average of the distances estimated, that's why we recommend to take $N_t \geq 12$.

If an obstacle is smaller than $L_{\min}$, there is no guarantee that this obstacle could be detected reliably. However, this obstacle can disturb the detection and the estimation of an other obstacle placed behind it. The minimum size $L_{\min}$ could be adjusted because the further away the object is, the more difficult it will be to detect.

## 4. The cross-correlation method

### *4.1. The discrete normalized temporal cross-correlation method*

Let us take two signals $S_1(t)$ and $S_2(t)$ constituting the outputs from two adjacent photosensors and $s_1(k)$ and $s_2(k)$ are taken to denote their discrete version where $t = k * dt$, $dt$ being the sample time. $T$ is the discrete version of $\tau$, so that $\tau = Tdt$.

The discrete normalized temporal cross-correlation (DNTCC) can be expressed as follows:

$$c(T,k) = \frac{\sum_{m=k-M}^{k} s_1(m)\, s_2(m+T)}{\sqrt{\sum_{m=k-M}^{k}\left(s_1^2(m)\right)\sum_{m=k-M}^{k}\left(s_2^2(m+T)\right)}} \tag{14}$$

where $M$ corresponds to the number of elements of $s_1$ and $s_2$ used to evaluate the cross-correlation. Without any loss of generality, (14) can be transform such for $\forall k > M + T$

$$c(T,k) = \frac{\sum_{m=k-M}^{k} s_1(m-T)\, s_2(m)}{\sqrt{\sum_{m=k-M}^{k}\left(s_1^2(m-T)\right)\sum_{m=k-M}^{k}\left(s_2^2(m)\right)}} \tag{15}$$

$$= \frac{\Sigma_{12}(T,k)}{\sqrt{\Sigma_1(T,k+1)\,\Sigma_2(k+1)}} \tag{16}$$

where $s_1(k)$ is the latest measurement of $S_1(t)$, $\Sigma_1(T,k) = \sum_{m=k-M}^{k} s_1^2(m-T)$, $\Sigma_2(k) = \sum_{m=k-M}^{k} s_2^2(m)$ and $\Sigma_{12}(T,k) = \sum_{m=k-M}^{k} s_1(m-T)\, s_2(m)$.

To find the maximum correlation, $c(\tau,t)$ must be evaluated using different values of $\tau$. To define these $\tau$ and in order to reduce the number of calculations and to avoid overloading our sensor's calculator, DNTCC is calculated taking only a limited number of delays $\tau$ corresponding to previously chosen distance range, as described in the following subsection.

### *4.2. List of delay $\tau$*

Finding the exact distance between the sensor and the obstacle with a temporal cross-correlation involves testing a large number of delay $\tau$, which makes for long computational times. Our method does not involve attempting to exactly determine the distance $D$: first because the optic flow does not give accurate measurements in all cases, and secondly, because it is often not necessary to know $D$ beyond a specific level of accuracy, especially when the obstacle is far away. Therefore, instead of searching for the real distance, the method will determine which distance $\tilde{D}$, out of a whole list of possible predefined distances $Ld_{test} = \begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_{n_0} \end{bmatrix}$ is nearest to $D$. It will therefore determine which $\tilde{\tau}$ out of a whole list of predefined delay $L\tau_{test} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \dots & \tau_{n_0} \end{bmatrix}$ is nearest to the real delay $\tau$. Only time delay featuring on the list $L\tau_{test}$ are therefore used to calculate the cross-correlation $c(T,k)$.

From a predetermined list of possible distances $Ld_{test}$ where $i \in [1 \dots n_0]$ and (5) and (1), the list of associated delay $L\tau_{test}$ can be calculated as follows:

$$L\tau_{test}(i) = \frac{\Delta\phi}{\frac{V_e}{Ld_{test}(i)} + \Omega_e}. \tag{17}$$

This method considerably reduces the number of operations required to find the maximum correlation between $s_1(t)$ and $s_2(t)$.

## 5. Moving Cross-correlation

### *5.1. New calculating NCC coefficient*

Since a distance must be estimated at each angle, no minimum value of the NCC coefficient is imposed to considered the correlation valid, unlike many methods using NCC: the distance with the highest correlation coefficient is considered to be the correct value. Thus, as described in equation (18), the NCC coefficient calculus can be simplified by removing the sum $\Sigma_2$ .

Moreover, the cross-correlation function indicates the point in time where the signals are best aligned, but actually the sign of this correlation is here not important: only values around zeros are an indication that there is no correlation. Thus, since the square function is an operation that requires computational resource, the following NCC coefficient was proposed to reduce computation time

$$c\left(T,k\right)=\frac{\left(\Sigma_{12}\left(T,k\right)\right)^{2}}{\Sigma_{1}\left(T,k+1\right)} \tag{18}$$

where the calculation of $\Sigma_1\left(T,k\right)$ and $\Sigma_{12}\left(T,k\right)$ is described in the following Section 5.2.

### *5.2. The new method of calculating NCC*

Calculating the cross-correlation coefficient (18) can be extremely time consuming because the sums $\Sigma_1\left(T,k\right)$, $\Sigma_2\left(k\right)$ and $\Sigma_{12}\left(T,k\right)$ have to be recalculated at each instant $k$. Closely in line with [11], an iterative process is proposed here to reduce the number of operations required and thus to reduce the computational time.

Equation (18) can also be written as follows:

$$\begin{aligned}\Sigma_{12}\left(T,k+1\right)&=\sum_{m=k+1-M}^{k+1}s_1\left(m-T\right)s_2\left(m\right)\\&=\sum_{m=k-M}^{k}s_1\left(m-T\right)s_2\left(m\right)+s_1\left(k+1-T\right)s_2\left(k+1\right)\\&-s_1\left(k-M-T\right)s_2\left(k-M\right)\\&=\Sigma_{12}\left(T,k\right)+s_1\left(k+1-T\right)s_2\left(k+1\right)\\&-s_1\left(k-M-T\right)s_2\left(k-M\right)\end{aligned} \tag{19}$$

Likewise, one can write:

$$\begin{aligned}\Sigma_1\left(T,k+1\right)&=\Sigma_1\left(T,k\right)+s_1^2\left(k+1-T\right)\\&-s_1^2\left(k-M-T\right)\end{aligned} \tag{20}$$

where $s_1\left(k+1\right)$ and $s_2\left(k+1\right)$ are the newly measured signals at instant $k$ and $s_1\left(k-M-K\right)$ and $s_2\left(k-M-K\right)$ are former measurements stored in memory.

Then, $\Sigma_1\left(T,k+1\right)$and $\Sigma_{12}\left(T,k+1\right)$ can be calculated faster using previous sums, and the two new measurements and storing in memory the latest $N=\frac{1}{dt}\max_{i=1:n_0}\left(L\tau_{test}\left(i\right)\right)$ values of each signal. The correlation $c\left(T,k+1\right)$ is obtained in this way using (18) with updated sums instead of re-calculating further sums at each iteration. This method significantly reduces the computational burden without any loss of accuracy and without increasing the size of the memory required to compute the time delay.

Note that this method induces us to keep the same time interval $L\tau_{test}\left(i\right)$ between each iteration of the sums $\Sigma_1\left(T,k\right)$ and $\Sigma_{12}\left(T,k\right)$: due to the constraints defined in Section 4.2, the relative velocity $V_e$ and the rotation $\Omega_e$ have to be kept constant.

### *5.3. Algorithm*

The Algorithms 1 and 2 describe the method used to calculate the NCC and the distance to be estimated in real time. To implement them, the following parameters are defined:

- $L\tau_{test}\left(i\right)\forall i\in\left[1\ldots n_0\right]$ is the list of time delays to be tested,
- $LT_{test}\left(i\right)=\frac{L\tau_{test}(i)}{dt}\ \forall i\in\left[1\ldots n_0\right]$ is the list of discrete delays to be tested,
- $n_2=M$ is the number of elements of $s_2$ used to calculate the cross-correlation,
- $n_1=1+n_2+\max_{i=1\ldots n_0}\left(LT_{test}\left(i\right)\right)$ is the number of elements of $s_1$ used to calculate the cross-correlation,
- $L_{S1}=0_{n_1\times1}$ is the list of measurement of $S_1$ stored,
- $L_{S2}=0_{n_2\times1}$ is the list of measurement of $S_2$ stored,
- $L_{sum1}=0_{n_0\times1}$and $L_{sum12}=0_{n_0\times1}$ are the lists of sums $\Sigma_1$ and $\Sigma_{12}$.
- $Start\_Save=0$ Boolean variable used to end the initialization procedure.

---

**Require:** $\forall i\in\left[1...n_0\right]$ choose $L\tau_{test}\left(i\right)$ and so $LT_{test}\left(i\right)$ % *Choose delays to test.*
Do Algorithm 2 **Calculation of** $D$. % *At each new measurement, calculation of the estimated distance D*
**if** $\left(k>\max\left(LT_{test}\right)\right)$, **then**
$Start\_Save=1$ % *End of the initialization. While* $k<\max\left(LT_{test}\right)$*, lists* $L_{S1}$ *and* $L_{S2}$ *are not full: evaluation of distance D can not be correct.*
$j=1$
**end if**
**if** $Start\_Save==1$ **then**
$L_D\left(j\right)=D$ % *stock the estimate distance D.*
$j=j+1$
**end if**

---

**Algorithm 1:** Main algorithm

These algorithms can be used to determine the distance $D$ at each new data acquisition by moving the $\frac{2\pi}{N_t}$ area of one measurement at each new data.

### *5.4. Selection and average of the $N_{tt}$ distances to display*

Let consider $N_{\text{data}}$ equal to the total number of data measured during one turn of the rotor. Using previous algorithm 1, $N_{\text{data}}$ distances are estimated during one turn. Since $N_{\text{data}}$ can be very large, it is recommended to define a number $N_{tt}$ of distances $D_2$ to display, where the $N_{tt}$ distances $D_2$ estimated are the average of $\frac{N_{\text{data}}}{N_{tt}}$ previous distances estimate, *i.e* for $i\in\{1...N_{tt}\}$

$$D_2(i)=\sum_{k=(i-1)*N_2+1}^{i*N_2}\frac{L_D(k)}{N_2} \tag{21}$$

where $N_2=\frac{N_{\text{data}}}{N_{tt}}$. Every sample of the distance array $D_2$ is spaced by an angle $\frac{2\pi}{N_{tt}}$ around the rotor circle.

$S_{2,old} = L_{S2}(1)$
$L_{S1} \leftarrow [L_{S1}(2:n_1), s_1(k)]$
$L_{S2} \leftarrow [L_{S2}(2:n_2), s_2(k)]$
$S_{2,new} = S_2(k)$
$c_{\max} = -1$
**for** $i = 1 : L_0$ **do**
  $ii = LT_{test}(i)$
  $S_{1,new} = L_{S1}(n_1 - ii)$
  $iii = ii + n_2$
  $\bar{S}_{1,old} = S_1(n_1 - iii)$.
  $L_{sum1}(i) = L_{sum1}(i) + (S_{1,new})^2 - (S_{1,old})^2$
  $L_{sum12}(i) = L_{sum12}(i) + S_{1,new}S_{2,new} - S_{1,old}S_{2,old}$
  **if** $(L_{sum12}(i))^2 > c_{\max}L_{sum1}(i)$, **then**
    $c_{\max} = \frac{(L_{sum12}(i))^2}{L_{sum1}(i)}$
    $\tau_{est} = L\tau_{test}(i)$ *% stock the estimate delay corresponding to the maximum correlation.*
    $D_{est} = Ld_{test}(i)$ *% if $V_e$ constant, the estimate distance can directly be stocked.*
  **end if**
**end for**

**Algorithm 2:** Calculation of $D$

### 5.5. Comparison between the present method and the Sum-Table Method

Tables 1 and 2 compare the number of arithmetical operations and the memory requirements between the classical method of NCC calculation, the Sum-Table Method developed in [11] and the present method, after all these methods have been adapted for real-time implementation.

The Sums Table method is described in greater detail in [11]. It is worth noting that the Sum-Table Method and our proposed method require much less operations than the classic NCC. Note also the Sum-Table Method uses less basic operations (sum, subtraction, multiplication and division) than our method, but requires several root square operators which required iteration algorithm (see for example [14]): since a single root operation require several iterations (depending of the accuracy asked by the programmer) each using several sums and multiplications. If the Sum-Table Method used only $5n_0 + 5$ basic operations compare to the $10n_0$ basic operations used by our method, the additional $n_0$ root square operators used by the Sum-Table Method involves many faster operation than the $5n_0$ additional basic operations used by our proposed method, which so requires less operation and computational time. Finally, our method uses less data to store in memory than the Sum-Table Method (see Table 2).

## 6. Experimental results

### 6.1. Matlab results

In this section, the results obtained with the algorithm presented in Section 5.3 are presented. The visual motion sensor designed and developed in this study was based on an off-the-shelf photosensor array (iC-LSC from the company iCHaus, [1] ) consisting of 2 rows of 4 photodiodes 300 m×800 $\mu m$ (see Figure 3). A fixed-gain current amplifier is integrated into each photodiode. To be able to sense a large number of contrasting objects, the photosensors in each column were paired to increase the sensitive surface two-fold to 300 $\mu m \times 1600\,\mu m$. Only the two central pixels were used here to measure the optic flow. A combination of two lenses of focal length 25 mm and 3.6 mm forming the optic, were mounted on the one-dimensional 4-pixel array (see Figure 5). The calibration process, as described in [19], gives an average $\Delta\phi = 0.0122$. Signals are filtered so as to keep the frequency in the $[F_b, F_h] = [1, 800]$ Hz range. This range was determined experimentally so as to obtain the most satisfactory velocity measurements.

To test our sensor, a bench (Figure 4) rotating at an angular velocity of $\Omega_r = [907, 1260]^\circ/s]$ ([2.5, 3.5] Hz) was built. The sensor is fixed on an arm $R = 1$ m in length, so a tangential velocity $V_t = 15.8$ m/s. When the bench rotates, the moving sensor assesses the distance to the surrounding objects placed in the experimental room (see figure 4). Signals $s_1$ and $s_2$ are acquired for 1 s by means of a NI data acquisition board (NI USB 6229) at an acquisition frequency of 80 kHz. Each angular position of the rotating arm was measured 50 times: the mean value and variance are presented in the results. The distances were estimated a posteriori using Matlab after the acquisition of 1 s of signal.

We choose $Ld_{test}(i) = \left[\, 0,\ 0.25,\ \ldots,\ 3,\ 3.5,\ 4\ \ldots, 7,\ 8,\ 9 \,\right]$m so $n_0 = 21$, $dt = 1.25e-05$ and $N_t = 36$ or $N_t = 60$. We choose to display $N_{tt} = 60$ estimated distances.

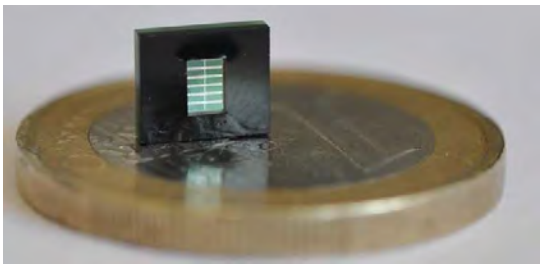

Figure 3: optic flow sensor.

Figures 6 and 7 show experimental results obtained using the proposed method for $N_t = 36$ (green) and $N_t = 60$ (blue) in Figure 6 and Figure 7), corresponding to $M = 888$ and $M = 533$. One observe that, overall, the room's distances are estimated with a good accuracy, except the corners and long distances. A computing time of 0.378 s was required to process the 1 s of measurement, thus making it possible to implement this method in real time. Our method makes it possible to obtain a large number of distance measurements, faster than a classical method cross-correlation method (1.125 s to measure 24 distances around the circle). This new approach also makes it

[1] http://www.ichaus.de

| | classic NCC | Sum-Table Method | Proposed method |
|---|---|---|---|
| **Initialization** | | | |
| Construction of $\sigma_{s_1^2}(k,T)$ | 0 | $M$ multiplication, $M$ addition | 0 |
| Construction of $\sigma_{s_2^2}(k)$ | 0 | $M$ multiplications, $M$ additions | 0 |
| Construction of $\sigma_{s_1 s_2}(k,T)$ | 0 | $n_0 M$ multiplications, $n_0 M$ additions | 0 |
| Calculation of $\Sigma_1$ | 0 | 0 | $n_0 M$ multiplication, $n_0 M$ addition |
| Calculation of $\Sigma_2$ | 0 | 0 | 0 |
| Calculation of $\Sigma_{12}$ | 0 | 0 | $n_0 M$ multiplication, $n_0 M$ addition |
| Total operations: | 0 | $(2n_0+4)M$ | $4n_0 M$ |
| | | | |
| **Main program** | | | |
| Construction of $\sigma_{s_1^2}(k,T)$ | 0 | 1 multiplication, 1 addition | 0 |
| Construction of $\sigma_{s_2^2}(k)$ | 0 | 1 multiplication, 1 addition | 0 |
| Construction of $\sigma_{s_1 s_2}(k,T)$ | 0 | $n_0$ multiplications, $n_0$ additions | 0 |
| Calculation of $\Sigma_1$ | $M$ multiplications, $M$ additions | 1 subtraction | $2n_0$ multiplication, $n_0$ addition, $n_0$ subtraction |
| Calculation of $\Sigma_2$ | $n_0 M$ multiplications, $n_0 M$ additions | $n_0$ subtractions | 0 |
| Calculation of $\Sigma_{12}$ | $n_0 M$ multiplications, $n_0 M$ additions | $n_0$ subtractions | $2n_0$ multiplication, $n_0$ addition, $n_0$ subtraction |
| Normalization | $n_0$ root square operators, $n_0$ divisions, $n_0$ multiplications | $n_0$ root square operators, $n_0$ divisions, $n_0$ multiplications | $n_0$ multiplication, $n_0$ divisions |
| Total operations: | $(4n_0+2)M + 2n_0$ $+n_0$ root square operators | $5n_0+5$ $+n_0$ root square operators | $10n_0$ +0 root square operators |

Table 1: Arithmetic Operations. $n_0$ is the number of delays tested, $M$ the number of elements of $s_1$ and $s_2$ used to evaluate the cross-correlation.

| | NCC | Sum-Table Method | Proposed method |
|---|---|---|---|
| **Initialization** | | | |
| Construction of $\sigma_{s_1^2}(k,T)$ | 0 | $M$ data | 0 |
| Construction of $\sigma_{s_2^2}(k)$ | 0 | $M$ data | 0 |
| Construction of $\sigma_{s_1 s_2}(k,T)$ | 0 | $n_0 M$ data | 0 |
| Calculation of $\Sigma_1$ | 0 | 0 | $n_0$ data |
| Calculation of $\Sigma_2$ | 0 | 0 | 0 |
| Calculation of $\Sigma_{12}$ | 0 | 0 | $n_0$ data |
| **Total data**: | 0 | $(n_0+2)M$ data | $2n_0$ data |
| | | | |
| **Main program** | | | |
| Data of $S_1$ | $M$ data | 1 data | $M$ data |
| Data of $S_2$ | $R$ data | $R$ data | $R$ data |
| Construction of $\sigma_{s_1^2}(k,T)$ | 0 | $M$ data | 0 |
| Construction of $\sigma_{s_2^2}(k)$ | 0 | $n_0 M$ data | 0 |
| Construction of $\sigma_{s_1 s_2}(k,T)$ | 0 | $n_0 M$ data | 0 |
| Calculation of $\Sigma_1$ | 1 data | 1 data | $n_0$ data |
| Calculation of $\Sigma_2$ | 1 data | 1 data | 0 |
| Calculation of $\Sigma_{12}$ | 1 data | 1 data | $n_0$ data |
| **Total data**: | $M+R+3$ | $(2n_0+1)M+R+4$ | $M+R+2n_0$ |

Table 2: Number of data to be stored in memory for evaluate the Normalization (18). $n_0$ is the number of time delays tested, $M$ is the number of elements of $s_1$ and $s_2$ used to calculate the cross-correlation. $R = M + \frac{\max(L\tau)}{dt}$ is the total length (in terms of the number of samples) of the signal used in the motion estimation.

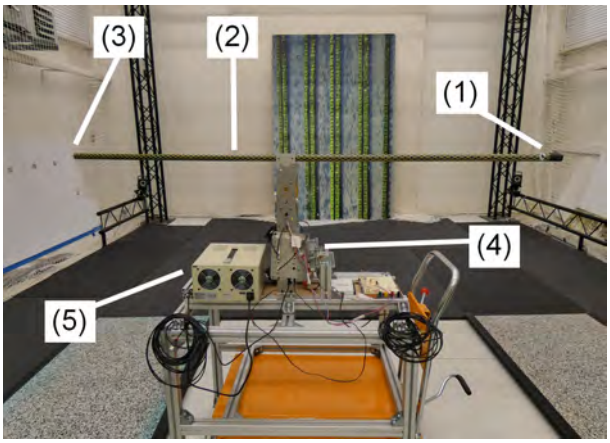


Figure 4: Rotating bench test in the experimental room. (1) sensor + lenses. (2) arm used to hold the sensor. (3) weight for balancing the test bench. (4) motor and reduction gear. (5) power supply. For an arm of $R = 1$ m in length here, the tangential velocity is $V_t = 15.8$ m/s (about 57 km/h).

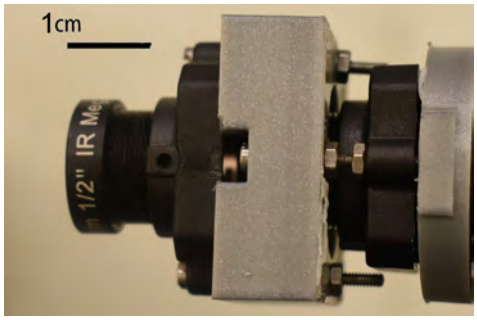


Figure 5: Combination of two lenses with focal lengths of 25 mm and 3.6 mm forming a minimalistic OF sensor.

possible to perform a new distance estimation at each new measurement which means that the maximal number of estimated distances is equal to the number of measurements performed.

Figure 6 and 7 show the estimated distance for $N_t = 36$ and $N_t = 60$, corresponding to $M = 888$ and $M = 533$. It is worth noting that the average value is similar in both case, but with a larger variance for $N_t = 60$ due to a smaller $M$. Indeed, increasing $M$ means that the distances were estimated by taking a larger number of measurements, thus covering a larger part of the room. Although the distance estimation obtained could be smoothed by filtering the false peaks due to errors in the calculations. As a consequence, small obstacles placed in the landscape were not detected accurately.

Figure 6 features a low variance showing that the estimated distance is accurate and repeatable. A peak at 200° was observed, corresponding to the furthest distance estimated, but its related variance is lower or equal to 1 m for a distance of 6 m.

Moreover, one can observe that the mean error is in most case lower than 0.5 m and even 0.25 m. Since the best theoretical accuracy imposed by the list $Ld_{test}$ is 0.25 m for distances smaller than 3 m and 0.5 m for distances smaller than 7 m, we can conclude our method estimates distance with a suitable accuracy. Peaks in the error correspond to the corners of the room, where the visibility is critical and very complex due to the geometrical shape.

The velocity of a large scale helicopter's rotor is around 4 Hz with blade of 4 m. Tangential velocity tested in our experiment is lower compared with a sensor fixed at the tip of a real helicopter blade, but very similar if the sensor is fixed at 1 m from the rotor center. Moreover, these first results are promising and more experimentation with a longer arm are planed.

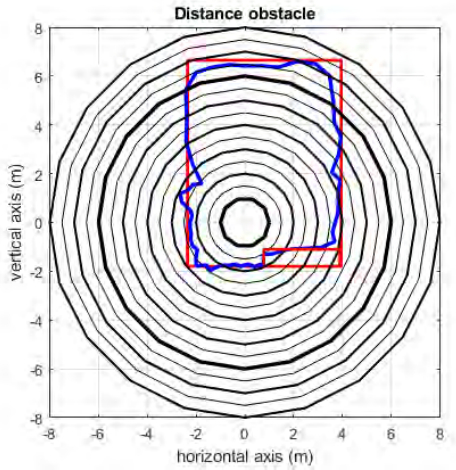


Figure 6: Distance estimated using the proposed method. Bench test is placed inside the central circle. Thick circles are spaced one meter apart, and distance between thick and thin circle correspond to 0.5m. Red : true distance. Blue : $N_t = 60$.

### 6.2. *Real time computation performances*

In this section, algorithms were implemented in realtime onboard a micro-controller (teensy 3.6) in C language. Due to the relatively limited computational resource, an acquisition frequency of 15 kHz was used. An obstacle of size 1 m at a distance of 2.5 m was added (see red rectangle in figure ) to propose a different configuration from the one shown in figure 6. Each angular position of the rotating arm was measured twice and the mean value of the estimated distance is plotted in figure 8. The distances were estimated after an acquisition of 1 s of signal.

We choose $Ld_{test}(i) = \left[\ 0.1,\quad 0.2,\quad \ldots,\quad 6.8,\quad 7\ \right]$m so $n_0 = 36$, $dt = 6.66e-05$ s and $n_2 = M = 215$ sample.

A computing time of 0.06 s was required to process the 1s of measurement, thus making it possible to implement this method in real time. Note since the acquisition frequency is much lower than the one used, this computation time is much lower than the one obtained with Matlab. However, it is worth noting that the obstacle of 1 m was perfectly detected at a distance of 2.5 m, confirming the theory exposed in Section 3.2.

In this study, tests were performed on a 1 m-long arm. During experiments not shown in this paper, we succeeded in obtaining a stable and accurate estimation of the distance for an arm as small as 0.4 m. A smaller arm could not but used due to the degradation of the translational OF when the arm becomes too small. Our method of obstacle detection can not be considered well-suited for MAVs yet, because it will require further exper-

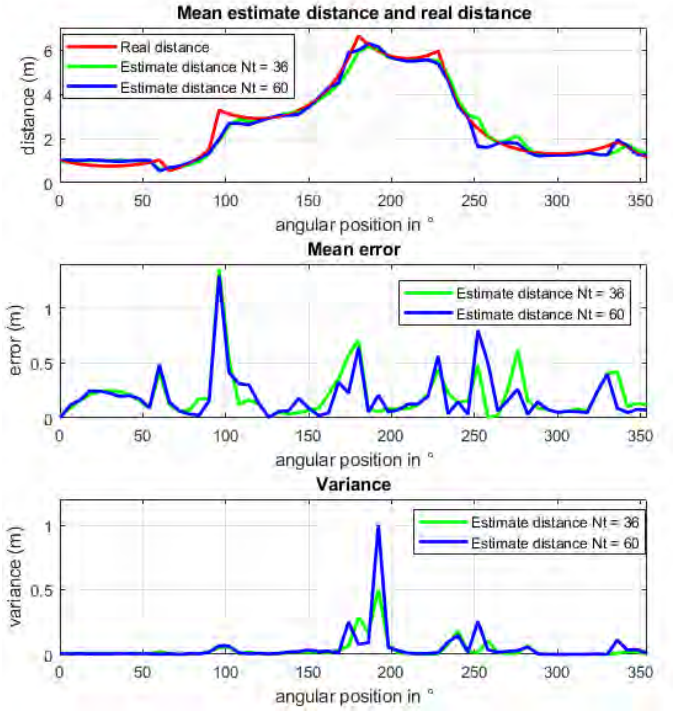


Figure 7: Comparison between real distances and the distances obtained at the velocity $\Omega_r = 907°/s$. Red : true distance. Green: $N_t = 36$. Blue: $N_t = 60$.

iments. However, our method can be considered suitable for helicopters of relatively large size.

## 7. Conclusion

This study proposes an implementation of a minimalist optical flow sensors to fix on a helicopter's blades, in order to estimate the distance to an obstacle using an OF-based method. Our sensor and method required few computational resource and a small number of components to be implemented (only two photodiodes, electronic amplifiers and an adequate lens). In this sense, it is an energy efficient design that makes our parsimonious sensor find its place in technologies for sustainable development goals (see [23]). The OF-based method of distance assessment presented here involves calculating normalized cross-correlations to determine the time delay between two signals. A special method was developed to reduce the time required to compute the correlations calculated in the OF processing without any loss of accuracy, associated with a choice of distances to test. A new algorithm is presented for implementing the method in real time. The results of the experimental tests show that distances to obstacles can be determined much faster with the new method presented here than with the classical method. Moreover, these results show a stable estimation of the distance with a small variance and an estimation error lower than 0.5 m for distances up to 6 m, corresponding to the accuracy required for such system.

In the near future, we will try to reduce the size of the arm on which the sensor is fixed in order to make our non-emissive optical anti-collision system suitable for MAVs.

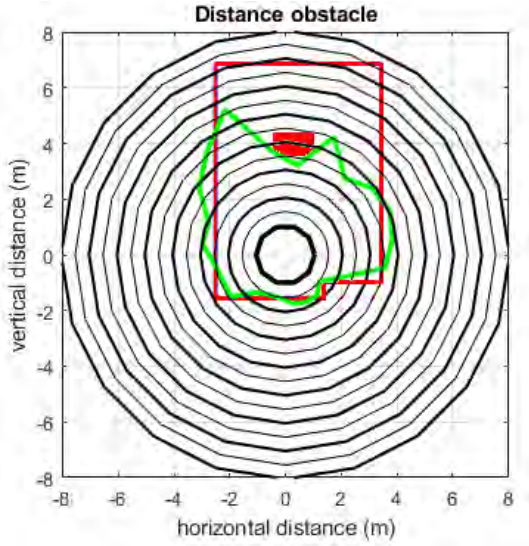


Figure 8: Comparison between real distances and the distances obtained at the velocity $\Omega_r = 1404°/s$ by means of realtime computation implemented onboard a microcontroller (teensy board). An additional obstacle (red rectangle) was placed at a distance of 2.5 m from the center. Red : true distance.

## 8. Acknowledgement

We acknowledge support from the Centre National de la Recherche Scientifique (CNRS), Aix-Marseille Universite and the Agence Nationale de la Recherche (ANR) (in the framework of the IRIS (Intelligent Retina for Innovative Sensing) project ANR-12-INSE-0009, OrigaBot project ANR-18-CE33-0008-01) and French national agency for Defense (DGA, Helistop project). An application of this study is developing in collaboration with Airbus Helicopters, an optical flow sensor fixed at the tip of an helicopter's blade to measure the distance to various obstacles (wall, cliff...) in the azimuthal plane (rotor plane) during hovering flight. The principle of this work has been patented [22].

## References

[1] J. FS Costa-Júnior, G.A. Cortela, Luis E Maggi, T. FD Rocha, W. CA Pereira, R. PB Costa-Felix, and A. V Alvarenga. Measuring uncertainty of ultrasonic longitudinal phase velocity estimation using different time-delay estimation methods based on cross-correlation: Computational simulation and experiments. *Measurement*, 122:45–56, 2018.

[2] PGM De Jong, T. Arts, APG Hoeks, and RS Reneman. Determination of tissue motion velocity by correlation interpolation of pulsed ultrasonic echo signals. *Ultrasonic Imaging*, 12(2):84–98, 1990.

[3] L Deck. Measurements using Fourier-transform phase shifting interferometry. 25:115–118, 2001.

[4] Karel Dudáček. Short time delay measurement: technical report no. dcse/tr-2015-03. 2015.

[5] J. Dupeyroux, V. Boutin, JR Serres, L.U. Perrinet, and S. Viollet. M 2 apix: a bio-inspired auto-adaptive visual sensor for robust ground height estimation. pages 1–4, 2018.

[6] S. Emani, KP S., VV S. Variyar, and S. Adarsh. Obstacle detection and distance estimation for autonomous electric vehicle using stereo vision and dnn. pages 639–648, 2019.

[7] HW Ho, C De Wagter, BDW Remes, and GCHE De Croon. Optical-flow based self-supervised learning of obstacle appearance applied to mav landing. *Robotics and Autonomous Systems*, 100:78–94, 2018.

[8] SM Hosseini and R. Abdollahi. Real-time dsp-based time delay estimation using two sensors. *IJE*, 105(4):598–613, 2018.

[9] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *In IEEE transactions on acoustics, speech, and signal processing*, 24(4):320–327, 1976.

[10] M. Lee, K-S Kim, and S. Kim. Measuring vehicle velocity in real time using modulated motion blur of camera image data. *In Proc IEEE Transactions on Vehicular Technology*, 66(5):3659–3673, 2016.

[11] J. Luo and E Konofagou. A fast normalized cross-correlation calculation method for motion estimation. *In IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 57(6):1347–1357, 2010.

[12] W. Ma and J. Huang. Accurate time delay estimation based on sinc filtering. 2:1621–1624, 2002.

[13] S. Mafrica, A. Servel, and F. Ruffier. Minimalistic optic flow sensors applied to indoor and outdoor visual guidance and odometry on a car-like robot. *Bioinspiration & biomimetics*, 11(6):066007, 2016.

[14] P. Markstein. Software division and square root using goldschmidt's algorithms. In *Proc. 6th Conference on Real Numbers and Computers (RNC'6)*, volume 123, pages 146–157, 2004.

[15] K. McGuire, G. de Croon, C. De Wagter, B. Remes, K. Tuyls, and H. Kappen. Local histogram matching for efficient optical flow computation applied to velocity estimation on pocket drones. pages 3255–3260, 2016.

[16] K. McGuire, G. De Croon, C. De Wagter, K. Tuyls, and H. Kappen. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *In Proc IEEE Robotics and Automation Letters*, 2(2):1070–1076, 2017.

[17] Bas J Pijnacker H., K. YW Scheper, and G. CHE De Croon. Vertical landing for micro air vehicles using event-based optical flow. *Journal of Field Robotics*, 35(1):69–90, 2018.

[18] H. Roggeman, J. Marzat, M. Derome, M. Sanfourche, A. Eudes, and G. Le Besnerais. Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. pages 2090–2099, 2017.

[19] F. L. Roubieu, F. Expert, M. Boyron, B. Fuschlock, S. Viollet, and F. Ruffier. A novel 1-gram insect based device measuring visual motion along 5 optical directions. pages 687–690, 2011.

[20] J. R. Serres and F. Ruffier. Optic flow-based collision-free strategies: From insects to robots. *Arthropod structure & development*, 46(5):703–717, 2017.

[21] E. Vanhoutte, F. Ruffier, and J. Serres. A quasi-panoramic bio-inspired eye for flying parallel to walls. In *2017 IEEE SENSORS*, pages 1–3, 2017.

[22] S. Viollet, F. Colonnier, and E. Vanhoutte. System for measuring the distance by optical flow, patent n°wo2018065737, 2017.

[23] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang. Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives. *IEEE Communications Surveys & Tutorials*, 20(3):2389–2406, 2018.

[24] J-C Yoo and TH Han. Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6):819, 2009.