This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

IEEE Transactions on Cognitive and Developmental Systems, Volume 12, March 2020, pp. 1 - 11
DOI: 10.1109/TCDS.2018.2858742

A Furcated Visual Collision Avoidance System for an Autonomous Micro Robot

Hamid Isakhani, *Member, IEEE*, Nabil Aouf, Odysseas Kechagias-Stamatis, and James F. Whidborne, *Senior Member, IEEE*

***Abstract*—This paper proposes a secondary reactive collision avoidance system for micro class of robots based on a novel approach known as the Furcated Luminance-Difference Processing (FLDP) inspired by the Lobula Giant Movement Detector, a wide-field visual neuron located in the lobula layer of a *locust* nervous system. This paper addresses some of the major collision avoidance challenges; obstacle proximity & direction estimation, and operation in GPS-denied environment with irregular lighting. Additionally, it has proven effective in detecting edges independent of background color, size, and contour. The FLDP executes a series of image enhancement and edge detection algorithms to estimate collision threat-level which further determines whether or not the robot's field of view must be dissected where each section's response is compared against the others to generate a simple collision-free maneuver. Ultimately, the computation load and the performance of the model is assessed against an eclectic set of off-line as well as real-time real-world collision scenarios validating the proposed model's asserted capability to avoid obstacles at more than 670 mm prior to collision, moving at 1.2 ms^{-1} with a successful avoidance rate of 90% processing at 120 Hz on a simple single core microcontroller, sufficient to conclude the system's feasibility for real-time real-world applications that possess fail-safe collision avoidance system.**

***Index Terms*—Autonomous robots biologically-inspired collision avoidance furcated luminance-difference processing (FLDP) direction and proximity estimation.**

I. Introduction

COLLISION avoidance is an intricate and vital block within an autonomous system for which, various hardware and software solutions are being tested and studied in order to develop a reliable and robust system capable of generating collision-free control commands independent of human supervision and control. Limited payload delivery of the smaller robots hinders the development of their autonomous control systems, though certain algorithms implemented on micro-sized processors deliver a reasonably efficient collision-free maneuvering capability [1]–[5]. However, substantial hardware advances are emerging in development of larger aerial robots (<20kg) bolstered by their greater payload capacity accommodating sophisticated sensors and computers to perform fully autonomous take-off and landing [6]–[8], stabilization and localization [9]–[11], collision avoidance [12], [13], and aerobatic flight [14], [15]. Hence, the software developments for micro class of mobile robots must be further explored to compensate for hardware deficiencies.

Contemporary hardware solutions for collision sensing and avoidance, for instance, a planar laser range finder (LIDAR) is much heavier than the available payload on a small category unmanned aircraft. On the other hand, active range finders such as the Microsoft Kinect 2 are incapable of operating outdoors, hence one of the most feasible solutions remaining, could be the fusion of a lightweight camera and a robust computer vision algorithm. Salient features of vision-based solutions specifically for aerial robotic applications include its simplicity, reduced weight, and cost, justifying researchers' focus on this technique. Ground robot application of a vision-based machine learning algorithm to avoid obstacles autonomously at a processing frequency of 7 Hz and a forward velocity of 5 ms^{-1} in a complex cluttered environment produced satisfactory results (<2% errors) [16] which was improved further in aerial applications of on-board depth computation and learning range classifiers in real-time, facilitating autonomous collision-free flight over 100 m [17]. Although, these monocular vision-based solutions satisfy the hardware limitations, they fail to cope with the curbed processing power as they are computationally expensive, thus a simple biologically inspired vision-based model is proposed, whose principle of operation is based on difference frame processing, a rather appealing behavior observed in the lobula layer of the *locust* visual neuron that performs an extremely simple process of image segmentation by computing absolute gray scale image difference over two consecutive frames [18]–[22].

Biologically-inspired robotics is burgeoning as the neuroscientific study of biological beings develop [23]. As an example, Lambrinos' polarized light compass exhibited by a desert ant called *Cataglyphis*, has been implemented on a ground robot (Sahabot) [24] where a triple light-polarizing sensor inspired by the polarization-opponent interneurons of a cricket's optic lobe provides the input to the compass. However, they fail to yield a comprehensive computational model as a result of ambiguously identified neural networks. A fly's motion detection capability on the other hand, has been implicitly applied in robotic navigations based on a Reichardt correlation-type elementary motion detector that is inspired by fly optic lobe [25]. Furthermore, the control algorithm based on optic flow methods maneuvers the robot through complex environments successfully. Nevertheless, these techniques are computationally expensive, hence succumb to the challenging

H. Isakhani is with the Computational Intelligence Laboratory, School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK. (e-mail: hIsakhani@lincoln.ac.uk)

N. Aouf and O. Kechagias-Stamatis are with the Signals and Autonomy group, Centre for Electronic Warfare, Information and Cyber, Cranfield University, Defence Academy, Swindon SN6 8LA, UK. (e-mail: n.aouf@cranfield.ac.uk, o.kechagiasstamatis@cranfield.ac.uk)

J.F. Whidborne is with the Centre for Aeronautics, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford MK43 0AL, UK. (e-mail:j.f.whidborne@cranfield.ac.uk)

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

requirement of this research that is an extreme limitation on computational power. Therefore, a successfully tested, simple, and comprehensive computational model of a Lobula Giant Movement Detector (LGMD) is considered here to provide a source of inspiration for this study.

An LGMD responds decisively to an approaching object on collision course, facilitating collision-free flights in vast and dense swarms [26]. It is a bilaterally paired motion sensitive neuron that responds robustly to images of objects approaching on a collision course by integrating input signals from the photoreceptors. The computational model of an LGMD was pioneered by Rind et al. [27], and has continuously evolved over the years, with the current version being implemented in ground robot navigations and autonomous cars involving diverse set of adaptations and modifications with respect to their applied field. One of the outstanding features of LGMD is the neuron's capability to detect direction of an obstacle's motion involving approach, translation, and recession [28], [29], which is achieved by the integration of a feed-forward inhibition, and ON/OFF channels. However, greater robustness, precision, and computational simplicity could be achieved by enhancing these connectomes with complementary modules that emulate the near-exact intricate behavior of an effective insect-vision suitable for mobile robotic applications.

Continuous attempts are being made to contribute towards modification of an LGMD model, including the pioneers of the original model, Sztarker and Rind [30] whose work found that, apart from the conventional LGMD and DCMD (Descending Contra-lateral Movement Detector), there are additional neurons in the *locust* optic lobe that respond to expanding stimuli and have been implicated in triggering evasive responses, naming the additional neural network as LGMD2. Furthermore they provide evidence to prove that the two share many key features including direction selectivity, but neither its role in overall behavior nor its post-synaptic target neurons have been studied in detail yet, offering a potential subject of research for neuroscientists.

Silva et al. [31], [32] proposed a modified LGMD architecture integrating two previous LGMD models to implement features such as noise immunity proposed in [18], [33] and direction detection proposed in [19], the optimized model is validated against a set of test cases (collision scenarios), where a successful filtering of isolated excitations is performed to prevent the perturbations from contributing to the excitation of the LGMD cell. Silva goes on to demonstrate that the neural architecture introduced in [18] was unsatisfactory when tested for shrinking stimuli whereas the model in [19] detected obstacle motion direction in depth. However, the latter is not immune to signal noise, which could cause faulty collision alarms in case of a perturbed input signal. Hence, fusing the advantages of the two models, a modified model is achieved that can distinguish an approaching from a receding object while ignoring input signal noise.

Cuadri et al. [26], among others, introduced two complementary modules called the Attention Focusing Mechanism and Topological Feature Estimator to augment their model's versatility and efficiency, where the first module aims to optimize the use of computational power by limiting the processing core to focus only on the frame zones that exhibit maximum activity, and the second module aims to extract further information about the current alarming status, particularly making a quick categorization of the approaching obstacle. Similar to the afore mentioned findings, in this paper, a biologically-inspired computational model is designed and tested to perform an effective collision avoidance onboard a mobile robot based on a novel approach called the Furcated Luminance-Difference Processing (FLDP). This solution addresses some of the major collision avoidance challenges using complementary modules such as Image Stabilizer, Contrast Rectifier, and Proximity Estimator, to reduce motion-induced vibrations, rectify irregular lighting, and distinguish between a large faraway object and a small imminent one, respectively. Experiments, results and performance assessments are presented systematically to bolster the asserted capabilities. Although the pure engineering contributions of this research might not be comparable to the current literature, we mainly aim to appreciate and inspire study on the biological solutions to address engineering problems.

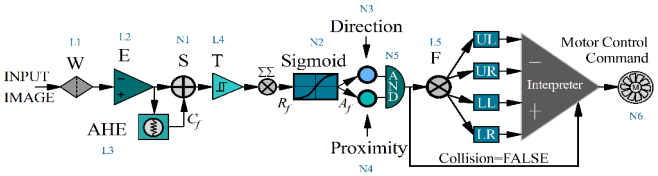


Fig. 1. Schematic illustration of Furcated Luminance-Difference Processing (FLDP) model. The proposed architecture consists of 5 layers (L1,L2,...,L5) of processing namely, linear time-invariant filter (W), excitation (E), adaptive histogram equalization (AHE), threshold (T), and furcation (F). Followed by 6 single processing nodes (N1,N2,...,N6) namely, the direction detector, depth estimator, summation, sigmoid transformer, AND logic gate, and motor control command generator node.

The organization of this paper is as follows, Section II introduces the structure of the proposed model and its processing layers. Section III demonstrates the off-line experiments and testing of the algorithm. Section IV illustrates real-world test setup and results. Section V assesses the results and compares the performance of the model with the current literature. And ultimately, section VI offers concluding remarks.

II. Model Design

The reactive collision avoidance model proposed here is based on the luminance difference processing exhibited by an LGMD that is sensitive to looming objects causing changes in luminance projected on the photoreceptor cells of *locust* compound eye, creating edges that help the insect distinguish a receding object from an approaching one by estimating these edges grow and move as described by the neuroscientists in related research [22], [27], [33], [34]. The architecture of our algorithm called the Furcated Luminance-Difference Processing (FLDP) is illustrated in Figure 1 which is carefully structured and enhanced to attain maximum robustness required on-board a mobile robot. The proposed architecture consists of five layers of processing (L1,L2,...,L5) namely, linear time-invariant filter layer (W), excitation layer (E),

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

adaptive histogram equalization layer (AHE), threshold layer (T), and furcation layer (F), preceded by six single processing nodes (N1,N2,...,N6) namely, the direction detector, depth estimator, summation, sigmoid transformer, AND logic gate, and motor control command generator node.

As shown in Figure 2(a), the input images a_f, a_{f-1} and so on, are a sequence of f number of 2D arrays with a dimension kxl pixels, where each pixel value $(u, v) \in [0-255]$ distributed panoramically along u(horizontal axis) and v(vertical axis) representing gray scale image frames of the input signal illustrated in Figure 1.

A. Layer-W

Since aerial robots exhibit vibrations over various frequencies, a linear time-invariant filter such as a Wiener filter [35] is introduced as the first layer to eliminate noise (undesirable blur) adaptively within the input image exerting a minute computation load (∼4 milliseconds to process 1 frame) by estimating the local mean (μ_f) and variance (σ_f) around each pixel as,

$$\mu_f = \frac{1}{NM} \sum_{u=1}^{N} \sum_{v=1}^{M} a_f(u, v) \tag{1}$$

and

$$\sigma_f^2 = \frac{1}{NM} \sum_{u=1}^{N} \sum_{v=1}^{M} a_f^2(u, v) - \mu_f^2, \tag{2}$$

where N and M are the local neighborhood of each pixel along horizontal and vertical axis respectively in the input image, which further creates a pixel-wise filter using,

$$b_f(u, v) = \mu_f + \frac{\sigma_f^2 - w^2}{\sigma_f^2} [a_f(u, v) - \mu_f] \tag{3}$$

where b_f is the filtered input signal and w^2 is the noise variance (additive noise) which is not specified for our application, hence considered as the average of all the locally estimated variances.

B. Layer-E

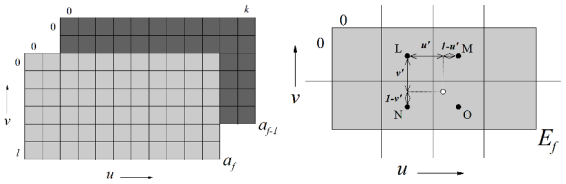
The filtered signal is then fed to the excitation layer that estimates the luminance difference of two consecutive input image frames by computing the absolute difference of the value of each pixel within the 2D array of an image with respect to its previous time-step, mathematically,

$$E_f(u, v) = |b_f(u, v) - b_{f-1}(u, v)| \tag{4}$$

where E_f is the output of the excitation layer at frame-f, b_f and b_{f-1} are the filtered luminance at current and previous frames f and $f - 1$, respectively.

C. Layer-AHE

The next processing layer is the adaptive histogram equalizer (AHE) [36] that enhances the excitation frame's contrast, facilitating operation in an environment with an irregular lighting by defining a fine boundary separation along obstacle edges through subdivision and interpolation scheme. As shown in Figure 2(b), gray-level attribute (denoted by a white dot) is determined by the gray-value distribution in its neighboring contextual regions (2x2 array) of a sample point in an image frame with centers denoted as L, M, N, and O where local gray-level mappings ($g_L(c)$, $g_M(c)$, $g_N(c)$, and $g_O(c)$) are based on the histogram of the contained pixels in the input frame. Considering c as the original pixel intensity for the sample point, we compute its new value by bilinear interpolation of the gray-level mappings that were calculated for each of the neighboring contextual zones as,



(a) (b)

Fig. 2. Input data representation. (a) Schematic illustration of a sample input data that consists of a sequence of image frames denoted by f, each containing kxl pixels (k and $l \in$ [0-255]) distributed panoramically over (u,v) plane. (b) Schematic illustration of contextual regions (2x2 array) of a sample point in an image frame with centers denoted as L, M, N, and O where local gray-level mappings (g_L(i), g_M(i), g_N(i), and g_O(i)) are based on the histogram of the contained pixels in the input frame, i being the original pixel intensity for the sample point. The gray-level attribute (denoted by a white dot) is determined by the gray-value distribution in its neighboring contextual regions.

$$\begin{aligned} c'(u', v') = \{&(1 - v')((1 - u')g_L(c) + u'g_M(c)) \\ &+ v'((1 - u')g_N(c) + u'g_O(c))\} \end{aligned} \tag{5}$$

Here u' and v' are the normalized distances with respect to the point L. The optimal contrast is thus calculated by dividing the entire image frame into such rectangular contextual elements shown in Figure 2(b) (a sample zone), and then tile-mapped to attain a whole contrast-equalized image as,

$$C_f(u, v) = \begin{bmatrix} c'_{11} & c'_{12} & \cdots & c'_{1k} \\ c'_{21} & c'_{22} & \cdots & c_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ c'_{l1} & c'_{l2} & \cdots & c'_{lk} \end{bmatrix} \tag{6}$$

where C_f is the concatenation of all the contextual regions along u and v forming a 2D array of pixels with the original dimension of the input image, kxl.

D. Layer-S

The obtained contrast-corrected image C_f is added to the excitation image E_f to get,

$$S_f = \rho_s \times |E_f + C_f| \tag{7}$$

The summed output is treated with a sensitivity factor, $\rho_s \in [0.2 - 0.5]$, that is an empirically determined coefficient (based on operation environment) to help mitigate the undesirable excitations due to perturbations (environmental buffeting).

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

E. Layer-T

Furthermore, the resulting summed data S_f is passed through a simple binary thresholding image segmentation process,

$$T_f(u,v) = \begin{cases} 1 \text{ (White)}, & \text{if } S_f(u,v) \geq T_r \\ 0 \text{ (Black)}, & \text{if } S_f(u,v) < T_r \end{cases} \tag{8}$$

Here the output T_f converts the input matrix S_f into binary dataset and replaces all pixels with luminance greater than the defined threshold (T_r) with the value 1 (white) and replaces all other pixels with the value 0 (black). T_r, a global image threshold is a scalar value determined using Otsu's method [37]. It is a function of zeroth- and the first-order cumulative moments of the gray-level histogram of the input image.

F. Response Generation Node

Ultimately, at this node the excited (white) pixels that have passed the threshold (T_r) are summed along both dimensions of the array to generate a system response as,

$$R_f(u,v) = \sum_{u=1}^{k} \sum_{v=1}^{l} |T_f(u,v)| \tag{9}$$

which is then fed to the spike generator to interpret the model response (collision alarms) as spikes. This is accomplished by transforming the response signal into a sigmoid function as,

$$A_f = (1 + e^{-R_f/s_{cell}})^{-1} \tag{10}$$

where s_{cell} is the total number of the excited pixels that have passed the threshold, and since R_f is greater than zero, the normalized spiking response, $A_f \in [0.5 - 1.0]$.

G. Direction and Depth Estimation Nodes

The generated spiking response is then interpreted to estimate the collision threat-level and nature of the threat, that is its course and proximity. First the direction of obstacle's motion relative to the robot is determined by comparing average spiking response over four time-steps using a small amount of memory that records consecutive spike values (accurate to the thousandths place) to check for any persistent increase or decrease implying obstacle approach or recession, respectively. This is achieved by assigning the direction of an obstacle's motion relative to the robot with a binary value $\in$ [0,1] using the condition,

$$Direction = \begin{cases} \text{Approaching} = 1, & \text{if avg}A_f > \text{avg}A_{f-1} \\ \text{Receding} = 0, & \text{if avg}A_f < \text{avg}A_{f-1} \\ \text{Stagnant} = 0, & \text{otherwise} \end{cases} \tag{11}$$

and then estimating the obstacle's proximity relative to the robot using the condition,

$$Depth = \begin{cases} \text{Near} = 1, & \text{if avg}A_f > (\delta \times \text{avg}A_{f-1}) \\ \text{Far} = 0, & \text{otherwise} \end{cases} \tag{12}$$

where avgA_f is an average value of A_f over 4 time-steps, and $\delta = 1.35$, is an empirically estimated coefficient that determines the nature of generated spikes implying if a threat is distant or imminent which in turn facilitates a prudent avoidance-decision making. The logic bolstering this function is

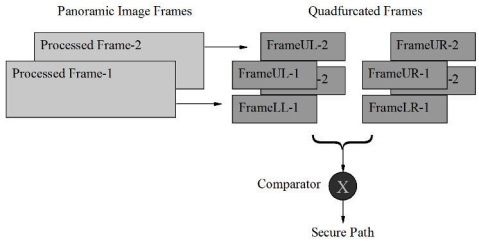


Fig. 3. Schematic illustration of furcation process, where the input image frames are dissected to form individual image frames (four in this case) each representing one section of the robot's field of view namely, upper-left (UL), upper-right (UR), lower-left (LL), and lower-right (LR).

H. Furcation Node

Direction and depth estimation nodes lay the foundation of the furcation process which performs a simple dissection of the input image frame into any number of symmetric quarters depending on the application. However, to further simplify the computation, an AND logic gate is introduced to decide whether or not to initiate the process of furcation by boolean multiplication expressed as,

$$\text{Furcate} = \text{Direction AND Depth} \tag{13}$$

Following the AND gate truth table, the furcation process is not initiated if either of the parameters 'direction' or 'depth' is OFF, whereas if both parameters are turned ON exhibiting state 1, the algorithm furcates the output of thresholding layer to form individual image frames each representing one section of the observer's field of view, illustrated in Figure 3. This node is of great significance to this research as it provides a simple yet effective solution for the challenging task of collision-free trajectory planning. It must be noted that the robot's degree of freedom (DOF) decides the number of sections FOV is split during the furcation process. Further processing is similar to the previous section but performed on four quarter images in parallel. Furcation of T_f array is represented as,

$$T_{f_1}(x_1, y_1) = T_f(\frac{-u}{2}, \frac{v}{2}) \tag{14}$$

$$T_{f_2}(x_2, y_2) = T_f(\frac{u}{2}, \frac{v}{2}) \tag{15}$$

Similarly T_{f_3} and T_{f_4} are computed to be fed simultaneously to the spike generator as,

$$R_{f_1}(x_1, y_1) = \sum_{x_1=1}^{k_1} \sum_{y_1=1}^{l_1} |T_{f_1}(x_1, y_1)| \tag{16}$$

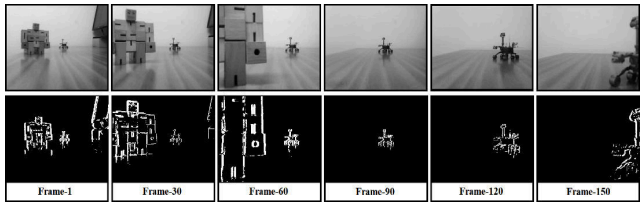
and

$$A_{f_1} = (1 + e^{-R_{f_1}/m_{cell}})^{-1} \tag{17}$$

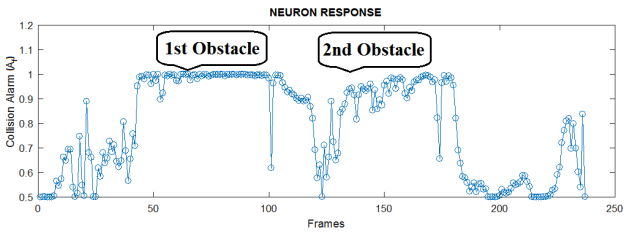
where m_{cell} is the total number of the excited pixels that have passed the threshold, and since R_{f_1} is greater than zero, the normalized spikes, $A_{f_1} \in [0.5 - 1.0]$, where 1.0 (spike) corresponds to an ON signal and every other value is OFF.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

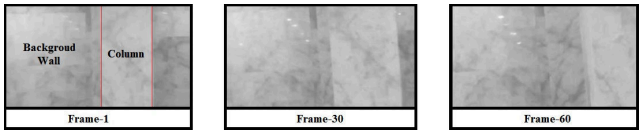


(a)

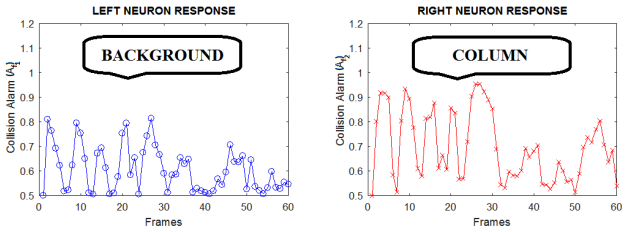


(b)

Fig. 4. Typical spiking response for an approaching obstacle. (a) Processed image frames (lower row) and raw input image frames (upper row) extracted from the input collision movie where the robot is moving at a linear velocity of 1.2 ms^{-1} towards two static obstacles of 0.5 m and 1 m wide on the left and right side of its field of view, respectively. (b) Spiking response for the illustrated collision movie, where the spikes exceed 0.95 border from 40th and 135th frame for the first and second obstacle, respectively, implying that the obstacles are detected more than 3 seconds prior to contact.



(a)



(b)

Fig. 5. Model response to evaluate edge detection capability. (a) represents 1st, 30th and 60th image frames extracted from the column detection movie (90fps), respectively. This scenario involves an obstacle (column) spanning ground to ceiling built with the same granite used in the background wall, approach the robot at a linear velocity of 1.4 ms^{-1} traveling a distance of 5.6 m in 4 seconds towards the right half of robot's field of view. (b) demonstrates the output spiking response of the model for the left and right half of the robot's field of view where the detected obstacle (column) causes elevated spike levels indicated in right neuron.

I. Motor Command Generation

Further the average response over four time-steps (4 frames) is computed as,

$$\text{avg}A_{f_1} = \frac{\sum_{j=0}^{4} A_{(f-j)_1}}{4} \tag{18}$$

Similarly, the average response of the remaining quadrants $(A_{f_2}, A_{f_3}, A_{f_4})$ are computed in parallel which are further compared against each other to estimate the most secure path (quadrant with the least average excitation) as,

$$\text{MotorCmd} = \text{avg}A_{f_1} - \text{avg}A_{f_2} - \text{avg}A_{f_3} - \text{avg}A_{f_4} \tag{19}$$

where the minus sign $(-)$ represents 'comparing' of quadrants, and 'MotorCmd' represents the quadrant with the least collision-threat level for which a Pulse-Width Modulation (PWM) signal is generated to control the turning radii (servo position) of the robot depending on the collision-threat level to navigate the robot through the estimated secure quadrant. Detailed experimental results and analysis of the proposed algorithm for different scenarios, velocities, and test environments are described in further sections.

III. Off-line Testing of Collision Avoidance Model

A comprehensive off-line analysis of the proposed collision avoidance model is conducted and briefly demonstrated along with the inferences drawn for a wide range of collision scenarios. The collision scenarios emulated in this section assess specific modules introduced as the proposed model's novelties including; static obstacle detection, obstacle direction detection, obstacle proximity estimation, and operation in complex backgrounds. In order to conduct a prudent off-line analysis, a complete set of input data are gathered to test the model for various possible scenarios preparing the algorithm for a successful real-time real-world application. Input dataset collection is performed systematically considering every vital parameter such as field of view (FOV=90°), data acquisition frequency (30Hz), image resolution, dimension, and format (320x240 Pixels in gray-scale). Similarly, the collision scenarios are orchestrated precisely by defining sample trajectories, various constraints, obstacles, and backgrounds to emulate real-world conditions.

Further, the model in its simplest form is fed with the collected input data to analyze the role of each processing node individually and calibrate them consistently. Once the model produces the desired response, modulation and enhancement of the algorithm is performed by introducing complementary modules to fulfill the task specific objectives described previously.

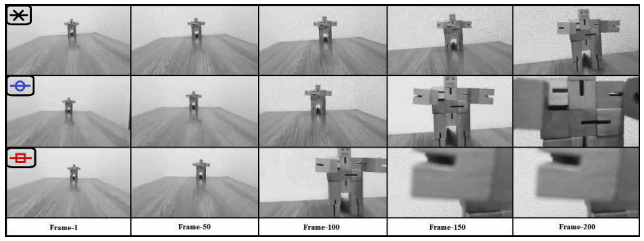
A. Collision Avoidance System

The input data presented here are a sample representation of the collision scenarios orchestrated to test individual features of the proposed model. This sample involves an observer (robot) moving at a linear velocity of 1.2 ms^{-1} towards two static obstacles of 0.5 m and 1 m wide on the left and right side of the robot's field of view (90°), respectively. These obstacles are placed clearly on the collision course and must eventually be avoided. Figure 4 illustrate sample image frames extracted from the input collision movie.

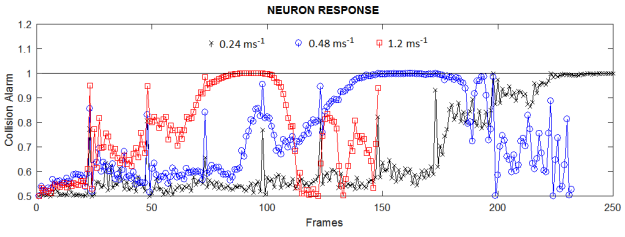
1) Complex Indoor Scenario: To evaluate the proposed model's capability to operate indoors, a complex-background scenario was orchestrated involving a brightly colored column (pillar) surfaced with the same granite used on the background wall. The robot approaches the granite column at a linear velocity of 1.4 ms^{-1} traveling a distance of 5.6 m in 4 seconds.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems



(a)



(b)

Fig. 6. Model response to evaluate model consistency. (a) Represents 1st to 200th image frames at 50 frame intervals extracted from the consistency-evaluation movies (recorded at 90fps for three linear velocities 0.24, 0.48, and 1.2 ms^{-1} ordered from top to bottom respectively). This scenario involves a brightly colored static obstacle with a 1.2 m separation looming in order to trigger collision avoidance. (b) Demonstrates the output spiking response of the model over 250 frames where the black cross, blue circles, and red squares correspond to three linear velocities 0.24, 0.48, and 1.2 ms^{-1}, respectively. When the collision alarm elevation reaches 1.0, it indicates a definite potential threat.

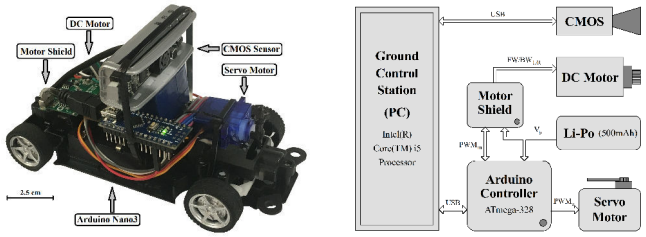
The model responds to the looming obstacle by increasing the collision alarm (spikes) shown in Figure 5, as the column nears, gradually the right half of the robot's field of view is activated, elevating the spike levels in the right neuron response colored in red crosses. Further, the spike interpreter generates left-steering motor control commands that maneuver the robot away from the pillar. It is evident that the model successfully detects like-colored obstacles and differentiates edges independent of their size, color and contour.

2) Model Consistency: A rather simple scenario was designed to evaluate the proposed model's response agility for various forward linear velocities. Here the robot approaches a brightly colored static obstacle with an initial separation of 1.2 m at 0.24, 0.48, and 1.2 ms^{-1} corresponding to black cross, blue circle and red square, shown from top to bottom in Figure 6(a), respectively. The spiking response for all three cases are presented in Figure 6(b), which illustrates the black cross, blue circles, and red squares spike level rise to 1.0 from the 200th, 130th, and 70th frame. Using the equation 20, we compute distances prior to a potential collision as 660, 510, and 300 mm, for the velocities in ascending order, respectively.

IV. Real-time Real-World Experimentation

In order to validate the proposed algorithm, we demonstrate a real-time implementation of the model on a ground robot tested against a number of real-world collision scenarios involving multiple obstacles of varying color, background, dimension, and contour.

The proposed model is implemented on a 3-DOF (degree of freedom) ground robot designed and fabricated at the Unmanned Autonomous Systems Laboratory (UASL), Cranfield University. It exhibits the necessary agility to accomplish successful collision avoidance using a DC motor, 9g servo motor, Arduino nano development board, motor shield, and a 2-Megapixel CMOS sensor. The Atmel ATmega328 8-bit AVR micro-controller with a maximum of 20 MHz operating frequency built into an Arduino development board is implemented to interface the robot with the proposed algorithm developed in MATLAB. Image frames captured by the CMOS sensor are transmitted through a USB cable to the ground control station (Intel Core i5-6500, 8GB RAM) where the images are processed and motor control commands generated. These commands are transmitted through the micro-controller to the servo and motor shield to control the robot's locomotion.



(a) (b)

Fig. 7. Robotic platform. (a) Assembled platform used for real-time real-world experimentation. (b) Simplified circuit connection depicting every component involving a basic 2-Megapixel CMOS sensor, an Atmel ATmega328 8-bit AVR micro-controller, basic servos and motors, and a ground control station PC (Intel Core i5-6500, 8GB RAM).

The conventional robotic differential steering was substituted with an Ackerman steering to replicate near-exact real-world four wheel vehicle dynamics posing greater maneuvering challenges as a result of underactuation. The schematic illustration of the designed robot, test platform and circuit connections are shown in Figure 7.

A. Real-World Collision Scenario

As mentioned in Section II-H, the robot's field of view is bifurcated here (3-DOF Robot). Initially the algorithm is executed in MATLAB and the robot launched along the trajectory. The robot is assigned to travel along a specified path through two waypoints 2 m apart steered with the help of the FLDP's servo commands interfaced through an Arduino micro-controller. The path of the robot is impeded with two obstacles, 0.15 m and 0.2 m high, laid 0.7 m apart. The robot moves with a linear forward velocity of 1.5 ms^{-1} traversing the assigned 2 m distance. However, this mission is accomplished without a collision at a success rate of ∼90% for 10 trials shown in Figure 8(a). The robot's field of view is bifurcated and processed to create a comparison between either directions (left-right) facilitating collision-free motor control command generation. The spiking response of the processed images for either half of robot's field of view is illustrated in Figure 8(c), where the first obstacle detected on the right half leads the robot towards left side. Further, the robot compensates the maneuver to return to the assigned trajectory which is again obstructed with another obstacle on the left causing generation

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

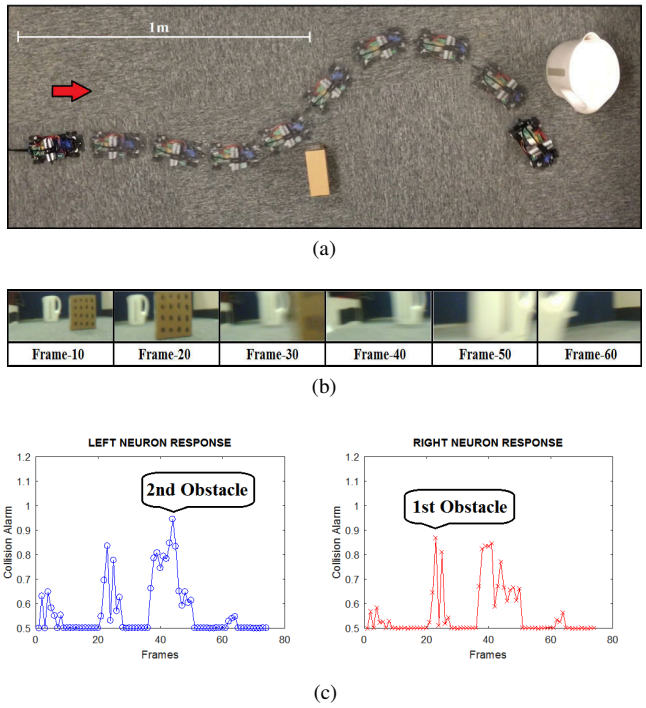


Fig. 8. Real world experimentation. (a) Blended frames of the top-view movie captured while robot travels the trajectory, involving a drift from left to right (indicated by the red arrow) through two obstacles of height, 0.15 m and 0.2 m laid 0.7 m apart, respectively. The linear velocity of the robot is 1.5 ms^{-1} traveling a total distance of 2 m without collision. (b) Robot's field of view where random obstacles are detected and avoided as a result of furcated luminance-difference processing. (c) Presents the spiking response of the algorithm with respect to potential collision threats where the spikes for obstacles 1 and 2 are annotated at frames-20 and 40, respectively. This implies that the system initiates avoidance at approximately 670 mm prior to collision.

of a right steering command shown in Figure 8(b), the entire process of consecutive obstacle avoidance is completed in 20 frames (0.6 seconds), initiated at frame-20 and completed by frame-40. Substituting the equation 20 with, initial obstacle-robot separation (I =1m), frame number at which the avoidance is initiated (x =20th)(Figure 8(c)), velocity of the robot (v =1.5ms^{-1}), processing frequency (f =90Hz), we obtain the proposed model's capability to avoid arbitrary real-world obstacles at 670 millimeters prior to collision.

B. Arbitrary Standard Collision Scenarios

In order to realize the ultimate objective of our proposed algorithm, an arbitrary set of conventional collision cases encountered by a generic robot was orchestrated. Figure 9(a-d) illustrates four individual scenarios tested for at least five trials generating more than ∼80% successful collision-free maneuvering of the robot through obstacles laid in various formations. For these tests, the robot was assigned to traverse a straight trajectory connecting two waypoints from extreme left to extreme right hand side. The velocity of the robot remained linear at ∼1 ms^{-1} traveling an approximately 1.5 m distance crammed with 0.1 m high obstacles (∼40 g crisp packets) whose flexible profile proved that the algorithm remains independent of obstacle contour, dimension, and color.

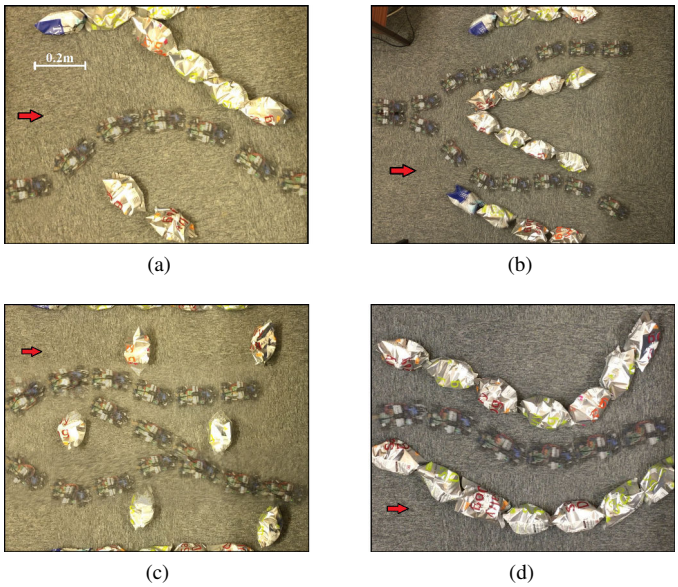


Fig. 9. Standard collision scenarios, involve a ground robot drifting at a linear velocity of ∼1 ms^{-1} traveling an approximately 1.5 m long trajectory connecting two waypoints from left to right saturated with 0.1 m high obstacles (∼40 g crisps packets). The standard scenarios orchestrated here are, (a) Right-angle inclined path, (b) Obstacles splitting path ahead, (c) Obstacles laid in checkered pattern, and (d) A challenging steep turn necessitating a correct set of consecutive motor control commands to maneuver without a collision.

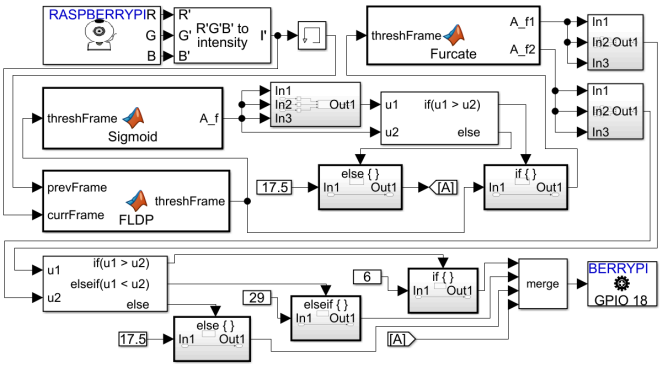
The illustrated test cases involve conventional scenarios and obstacle setup such as, right-angled path necessitating sharp maneuvers (Figure 9(a)), a 45° junction splitting trajectory (Figure 9(b)), checkered-pattern-laid obstacles crammed to convolute the test (Figure 9(c)), and ultimately the most challenging scenario considering the robot's underactuation (Ackerman steering) to make a steep turn demanding a correct sequence of consecutive motor control commands (Figure 9(d)).

C. On-board Implementation and Testing

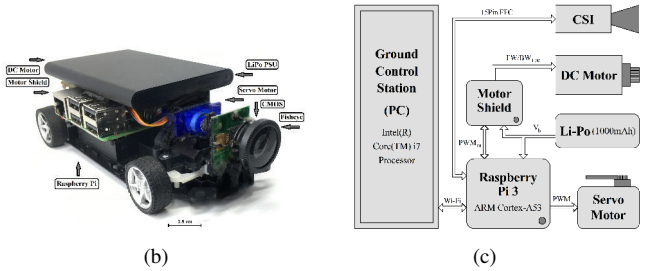
As a result of successful off-board testing of the FLDP described in previous sections, the algorithm was further reprogrammed as a Simulink model shown in Figure 10(a) to facilitate binary code generation for deployment on an ARM processor built into a Raspberry Pi 3 model B development board shown in Figure 10(b). The Simulink model is mainly composed of 3 MATLAB function blocks that execute the FLDP algorithm with C/C++ code generation support. The robot on the other hand, is equipped with a basic 2-Megapixel CSI sensor, a fish-eye lens, servos and motors, a Raspberry Pi 3 Model B board with an ARM Cortex-A53 processor and 1GB RAM. The Embedded Coder generates the binary code and deploys it for standalone operation on the robot through a Wi-Fi connection shown in Figure 10(c). Further, the image frames from the CSI sensor are passed to the processor for motor command generation in the form of PWM signals fed to the servo motor controlling the steering of the robot leading it towards the most secure path (quadrant of the robot's Field of View with the least average excitations).

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems



(a)



(b) (c)

Fig. 10. Embedded robotic platform. (a) Block diagram of the FLDP Simulink Model. (b) Assembled robot used for onboard implementation and testing of the FLDP. (c) Simplified circuit connection depicting every component involving a basic 2-Megapixel CIS sensor, a Raspberry Pi 3 Model B single board computer, servos and motors, and a ground control station PC (Intel Core i7-3520, 12GB RAM).

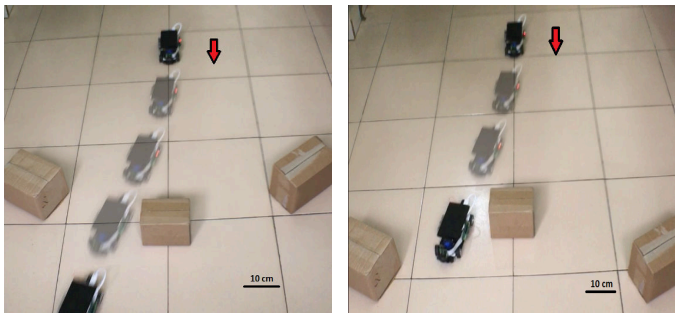
Ultimately, the algorithm is validated on-board the FLDP robot for real-world embedded application. The robot was assigned to traverse a straight trajectory connecting two way-points 1.5 m apart from top to bottom shown in Figure 11. The velocity of the robot remained linear at $\sim$1.5 ms^{-1} traveling an approximately 1.5 m distance that is obstructed with three 0.1 m high obstacles (cardboard boxes) laid in checkered pattern 0.25 m apart. The robot successfully avoids the obstacles and rides through passages that it finds with the least collision-threat level. Thus, the algorithm was successfully validated for applications on-board embedded ARM processors, however, further VLSI implementation of the algorithm is suggested as a future work.

V. System Performance Assessment

Five to ten test trials were executed for each scenario depending on their complexity to obtain a detailed and comprehensive conclusion on the model's performance and capabilities. In order to draw valid inferences, a set of established reference parameters such as, (1) Computation load, (2) Detection time and distance prior to collision, and (3) Detection error, were systematically studied in this section to assess and validate performance of the proposed model in off-line as well as real-world real-time scenarios.

A. Computational Complexity

The input signal of the dimension kxl pixels, considered as 'N' is fed to the FLDP algorithm that consists of 5 layers and



(a) (b)

Fig. 11. Simple Real-World Collision Scenario, involve the FLDP ground robot drifting at a linear velocity of $\sim$1.5 ms^{-1} traveling an approximately 1.5 m long trajectory connecting two waypoints from top to bottom with three 0.1 m high obstacles (cardboard boxes). The standard scenarios orchestrated here are, (a) Straight path laid with similar obstacles laid in an arc, (b) Straight path with obstacles laid in checkered pattern.

TABLE I
Computation Load Distribution

Process	Time (ms)	Percentage
Noise Filtering	4	47%
Contrast Correction	3	35%
Luminance Disparity	0.4	5%
Binary Thresholding	0.3	4%
Others	0.8	9%

6 processing nodes. This section presents the computational complexity (CC) of the entire as well as each layer and node of the FLDP. The CC of the; Wiener filter, excitation layer, adaptive histogram equalisation, summing, and thresholding are linear, dominated by O(N). Whereas the CC of the; sigmoid transformation, direction & depth detection, furcation (divisions of FOV= n), furcated sigmoid transformation, and motor command generation are dominated by O(1), O(2N), O(nN), O(n) and O(N), respectively. Hence according to Big-O notation, it can be concluded that the computational complexity of the entire FLDP algorithm is dominated by O(N).

Further, altering fundamental parameters such as furcation factor (n), does not affect the computational complexity as these parameters may only cause a constant change in the 'size' and not the 'nature' of the processed data. Also, switching the algorithm from parallel to serial processing shan't cause any change in the computational complexity as it only multiplies the processing time by the constant furcation factor (n), thus, the complexity yet remains as O(N).

B. Computation Load

On average, the total time required to process 1 image frame of 320 $\times$ 240 pixels is $\sim$8.5 milliseconds, which implies that the model can operate at $\sim$120 Hz in real-time tabulated in Table I. One of the most important performance assessment parameters in this paper includes the algorithm's processing power and memory requirements, since the designed system's primary target applications are micro robots, it is crucial to assess the system's required processing power and memory.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

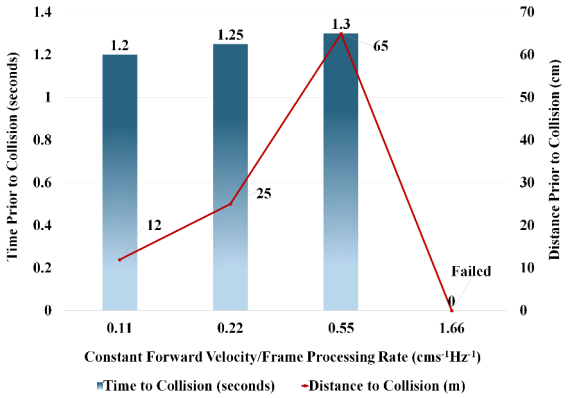


Fig. 12. Detection distance and time prior to collision. The graph presents time (primary vertical axis) and distance (secondary vertical axis) prior to a collision for different ratios of forward velocities to frame processing rate, to illustrate results obtained from tests using related methodologies as in [20], [22] (empirical parameters set according to our proposed model).

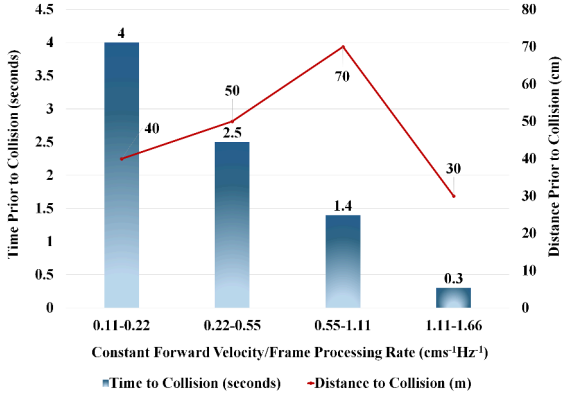


Fig. 13. Detection distance and time prior to collision. The graph presents time (primary vertical axis) and distance (secondary vertical axis) prior to a collision for different ratios of forward velocities to frame processing rate, to bolster model's consistency and agility. It is evident that the system remains consistent with slight drop in performance as the constant forward velocity increases.

For the computation load assessment, all the visualization elements such as response plot, playback object, processed frame display etc. are eliminated to minimize unnecessary load. The load test configuration presented in Table I was performed using MATLAB's built-in stopwatch to time every individual processing layer of the algorithm deducing a time distribution table. It is concluded that the total required time to process a single image frame of 320x240 pixels is 8.5 milliseconds, which implies that a maximum of 120 Hz processing speed can be achieved. However, greater frequencies are feasible by eliminating processing layers with a slightly lesser significance, such as contrast correction, which introduces a rather potential future research topic to conduct a prudent performance trade-off by analyzing these layers' significance with respect to their computation cost.

C. Distance and Time to Collision

One of the most common reference parameters for examining the performance of a collision avoidance algorithm is the distance and time before which a potential collision threat is successfully detected and avoided. This is particularly important for aerial applications as flying robots exhibit under-actuation that requires quicker detections providing sufficient time to initiate successful avoidance maneuvers. The related current literature provides performance assessments of an LGMD model on ground robots which we compare with the capabilities and performance of the FLDP model proposed here. These assessments involve results from [18], [20] and [31] which slightly contradict each other, as they implement different approaches to summarize their results by demonstrating that an increase in robot's forward velocity results in greater distances before which an avoidance maneuver can be initiated. However, the velocity ranges involved in these experiments are too low (0.2-0.5 ms^{-1}) compared to our tests where the velocities exceed a minimum of 1 ms^{-1} almost in all cases. However, their rate of correct detections are slightly higher (95%) due to the low forward velocities used in their experimentations. Silva et al. [31] on the other hand present their model performance based on simulated collision movies involving approach of a rectangular box at linear velocities in the range of 1 ms^{-1} whose distance prior to collision is estimated to remain between 250 to 350 mm with minimum consistency caused by the simulation methodologies and test environment. However, for a real video analysis the obstacle approach velocity is much smaller (0.15 ms^{-1}) where the distance to detection remains almost the same as in simulations.

Further to draw a contrast with respect to the related methodologies, our collected input dataset was supplied to the collision avoidance model with inhibition as a main processing block described in [18] to obtain the set of results illustrated in Figure 12. In line with the published results [18], [20], the time to collision remains more than 1 second when compared to our >3 seconds response shown in Figures 12 and 13. Hence the distance prior to collision is almost $1/3^{rd}$ of the corresponding results deduced in our proposed model for the same scenario and settings explained in detail below. Similarly, the nature of the response is also different, where rise in forward velocity up to approximately 0.5 ms^{-1} causes increase in distance-prior-to-collision, however, at higher velocities, that is post 1 ms^{-1}, the model fails to cope due to increasing false collision alarms.

Now to compare the assessment results of our proposed model with the results shown in Figure 12, we approximate the growth of our model's spiking response to an exponential curve whose slope directly depends on, (1) processing power, (2) memory, (3) data acquisition frequency (Hz), (4) robot's forward velocity, and (5) obstacle's distance from robot. Further, this curve is used to obtain the results shown in Figure 13. To maintain fair testing standards, the above parameters (1)-(5) are approximated to the nearest values with respect to the literature, and tests are performed at a similar laboratory conditions. The distance (D) and time (T) at which the model successfully detects a potential threat prior to collision and fires spiking response, is computed using the robot's linear velocity, initial obstacle distance from robot, and the frame at

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

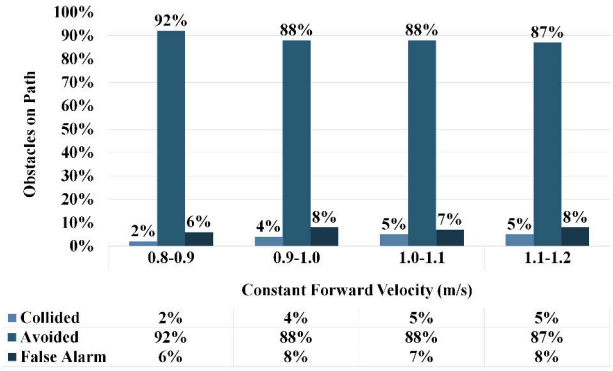


Fig. 14. Schematic illustration of detection errors. A graph of percentage obstacles detected versus linear forward velocity of the robot (8 tests per velocity range) is plotted to represent successful/unsuccessful collision alarms against false alarms for the afore described collision scenarios.

which the first avoidance command is generated, expressed as,

$$D = I - (x.v.f^{-1}) \tag{20}$$

$$T = D.v^{-1} \tag{21}$$

where, I is the initial obstacle distance from robot at 0^{th} frame, v is the robot's linear velocity, f is the data acquisition frequency, and x is the frame number at which avoidance is initiated. Using the above relations, time and distances were computed and plotted on a graph of robot's linear velocity illustrated in Figure 13, Avoidance maneuvers are initiated prior to at least five times the detected obstacle's size providing sufficient distance and time for a successful avoidance maneuver which is extremely favorable in aerial robotics where underactuation is a major concern. Although our maneuvering and control command generation is performed on an entirely different principle when compared to the methodologies implemented by the afore mentioned literature, nevertheless, the required objectives are achieved successfully. Figure 13 and 8 represent the time before collision for higher forward velocities as 670 mm for real-time real-world tests, which evidently bolsters the briskness of our proposed model for indoor micro robot collision avoidance systems. Hence it can be claimed that using this system, steering commands may be generated well prior to the occurrence of a collision providing secure path at a very low computation load.

D. Detection Errors

The proposed model was analyzed for its consistency and performance robustness by testing it against at least 8 test trials for each different scenario involving static obstacles of sizes within a range of 0.5 to 1 m wide sequentially obstructing the robot's path moving at an approximate forward velocity of 0.8 to 1.2 ms^{-1}. To draw a clear inference in this section, a graph of the percentage of obstacles; missed (collided with), avoided, and false detections were plotted against the robot's (observer) forward velocity in Figure 14. A detailed assessment result for different off-line scenarios is illustrated in Figure 13.

The model remains highly effective at lower velocities (0.10-0.20 ms^{-1}), although efficiency drops causing slight rise (2%) in generation of false collision alarms at higher forward velocities (1-1.2 ms^{-1}), yet it remains highly feasible for indoor applications since robots do not operate at elevated velocities within confined environments. Also considering the model's computational simplicity, the delivered efficiency is reasonably high compared to the associated research [20], [22], [31].

VI. Conclusion

This paper demonstrates a secondary vision-based collision avoidance system for autonomous micro robots, whose performance is validated against a diverse set of off-line (recorded) as well as real-world collision scenarios. Contribution of every individual processing layer is illustrated to bolster the claimed capabilities of the system such as, (1) Brisk response, (2) Irregular contrast correction, (3) Direction and proximity estimation, and (4) Optimized performance at minimal computation load. Ultimately, the performance assessment results validate the proposed model's capability to detect obstacles at more than 670 mm (real-world) prior to collision, moving at 1.2 ms^{-1} with a successful avoidance rate of greater than 90% processing at 120 Hz independent of obstacle color, dimension, and contour, sufficient to conclude the system success to effectively avoid obstacles in real-time and real-world applications that possess onboard fail-safe collision avoidance systems. It should be noted that the contributions of this research are not necessarily comparable to the current literature, as it mainly aims to appreciate and inspire further study on the biological solutions to address engineering problems.

For the future developments of this model, integration of an advanced image segmentation methodology, an efficient dynamic-obstacle tracking module, further optimization, complex cluttered environment compatibility, and real-world real-time implementation onboard an autonomous nano aerial robot is recommended.

References

[1] D. Floreano, J. Zufferey, and A. Klaptocz, "Aerial locomotion in cluttered environments," in *Proc. Int. Symp. Robotics Research*, 2011, pp. 1–19.
[2] A. J. Barry, "Flying between obstacles with an autonomous knife-edge maneuver," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2014, p. 2559.
[3] J. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Trans. Robotics*, vol. 22, no. 1, pp. 137–146, 2006.
[4] A. Klaptocz, L. Daler, A. Briod, J. Zufferey, and D. Floreano, "An active uprighting mechanism for flying robots," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1152–1157, 2012.
[5] C. D. Wagter, "Autonomous flight of a 20-gram flapping wing MAV with a 4-gram onboard stereo vision system," in *Proc. IEEE Int. Conf. Robotics and Automation*, Hong Kong, 2014, pp. 4982–4987.
[6] A. Johnson, J. Montgomery, and L. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 4470–4475.
[7] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002, pp. 2799–2804.
[8] S. Lin, M. A. Garratt, and A. J. Lambert, "Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment," *J. Autonomous Robots*, vol. 41, no. 4, pp. 881–901, 2017.
[9] T. Kanade, O. Amidi, and Q. Ke, "Real-time and 3D vision for autonomous small and micro air vehicles," in *Proc. IEEE Int. Conf. Decision and Control*, 2004, pp. 1655–1662.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2858742, IEEE Transactions on Cognitive and Developmental Systems

[10] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *J. Autonomous Robots*, vol. 41, no. 4, pp. 919–944, 2017.

[11] S. B. Fuller, Z. E. Teoh, P. Chirarattananon, N. O. Pérez-Arancibia, J. Greenberg, and R. J. Wood, "Stabilizing air dampers for hovering aerial robotics: design, insect-scale flight tests, and scaling," *J. Autonomous Robots*, vol. 41, no. 8, pp. 1555–1573, 2017.

[12] S. Scherer, "Flying fast and low among obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp. 2023–2029.

[13] D. Bareiss, J. R. Bourne, and K. K. Leang, "On-board model-based automatic collision avoidance: application in remotely-piloted unmanned aerial vehicles," *J. Autonomous Robots*, vol. 41, no. 7, pp. 1539–1554, 2017.

[14] V. Gavrilets, "Control logic for automated aerobatic flight of a miniature helicopter," in *Proc. AIAA Guidance, Navigation and Control Conf.*, 2002, pp. 1–8.

[15] W. E. Green and P. Y. Oh, "A hybrid MAV for ingress and egress of urban environments," *IEEE Trans. Robotics*, vol. 25, no. 2, pp. 253–263, 2009.

[16] J. Michels, A. Saxena, and Y. A. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proc. Int. Conf. Machine Learning*, 2004, pp. 593–600.

[17] D. Dey, K. S. Shankar, S. Zeng, R. Mehta, M. T. Agcayazi, C. Eriksen, S. Daftry, M. Hebert, and J. A. Bagnell, "Vision and learning for deliberative monocular cluttered flight," *J. Field and Service Robotics*, pp. 392–400, 2015.

[18] S. Yue and F. C. Rind, "Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement." *IEEE Trans. Neural Networks*, vol. 17, no. 3, pp. 705–716, 2006.

[19] H. Meng, S. Yue, A. Hunter, K. Appiah, M. Hobden, N. Priestley, P. Hobden, and C. Pettit, "A modified neural network model for lobula giant movement detector with additional depth movement feature," in *Proc. Int. Joint Conf. Neural Networks*, 2009, pp. 2078–2083.

[20] C. Hu, F. Arvin, C. Xiong, and S. Yue, "A bio-inspired embedded vision system for autonomous micro-robots: the LGMD Case," *IEEE Trans. Cognitive and Developmental Systems*, vol. 9, no. 3, pp. 241–254, 2017.

[21] G. Zhang, C. Zhang, and S. Yue, "LGMD and DSNs neural networks integration for collision predication," in *Proc. Int. Joint Conf. Neural Networks*, 2016, pp. 1174–1179.

[22] S. Yue and F. C. Rind, "Redundant neural vision systems-competing for collision recognition roles," *IEEE Trans. Autonomous Mental Development*, vol. 5, no. 2, pp. 173–186, 2013.

[23] A. H. Gaede, B. Goller, J. P. Lam, D. R. Wylie, and D. L. Altshuler, "Neurons responsive to global visual motion have unique tuning properties in hummingbirds," *Current Biology*, vol. 27, no. 2, pp. 279–285, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cub.2016.11.041

[24] D. Lambrinos, M. Maris, H. Kobayashi, T. Labhart, R. Pfeifer, and R. Wehner, "An autonomous agent navigating with a polarized light compass," *J. Adaptive Behaviour*, vol. 6, no. 1, pp. 131–161, 1997.

[25] W. Reichardt and M. Egelhaaf, "Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly," *J. Biological Cybernetics*, vol. 58, no. 5, pp. 287–294, 1988.

[26] J. Cuadri, G. Linan, R. Stafford, M. S. Keil, and E. Roca, "A bioinspired collision detection algorithm for VLSI implementation," in *Proc. SPIE Conf. Bioengineered and Bioinspired System*, vol. 5839, 2005, pp. 238–248.

[27] F. C. Rind and D. I. Bramwell, "Neural network based on the input organization of an identified neuron signaling impending collision." *J. Neurophysiology*, vol. 75, no. 3, pp. 967–985, 1996.

[28] M. Blanchard, F. C. Rind, and P. F. M. J. Verschure, "Collision avoidance using a model of the locust LGMD neuron," *J. Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 17–38, 2000.

[29] S. Bermudez i Badia and P. F. M. J. Verschure, "A collision avoidance model based on the lobula giant movement detector (LGMD) neuron of the locust," in *Proc. Int. Joint Conf. Neural Networks*, vol. 3, 2004, pp. 1757–1761.

[30] J. Sztarker and F. C. Rind, "A look into the cockpit of the developing locust: looming detectors and predator avoidance," *J. Developmental Neurobiology*, vol. 74, no. 11, pp. 1078–1095, 2014.

[31] A. C. Silva, J. Silva, and C. P. Santos, "A modified LGMD based neural network for automatic collision detection," in *Informatics in Control, Automation and Robotics*, 2014, vol. 283, pp. 217–233.

[32] A. Silva and C. P. Santos, "Modeling disinhibition within a layered structure of the LGMD neuron," in *Proc. Int. Joint Conf. Neural Networks*, 2013, pp. 1–7.

[33] S. Yue and F. C. Rind, "Near range path navigation using LGMD visual neural networks," in *Proc. Int. Conf. Computer Science and Information Technology*, 2009, pp. 105–109.

[34] F. C. Rind and P. J. Simmons, "Seeing what is coming: building collision-sensitive neurones," *J. Trends in Neurosciences*, vol. 22, no. 5, pp. 215–220, 1999.

[35] J. S. Lim, "2-D adaptive noise-removal filtering," in *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1990, p. 548.

[36] K. Zuiderveld, "Contrast limited adaptive histograph equalization," in *Graphic Gems IV*, 1994, pp. 474–485.

[37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, 1979.

Hamid Isakhani (M'18) received the B.Eng. degree in aeronautical engineering from Visvesvaraya Technological University, and the MSc by Research in Aerospace from the Cranfield University in 2015 and 2017, respectively. He was an intern engineer at the Indian Space Research Organisation and Hindustan Aeronautics Limited during the years 2012 and 2014 respectively. He is currently a PhD scholar at the School of Computer Science, University of Lincoln, UK, and visiting Tsinghua University as a Marie Curie Research Assistant since 2018. His research interests include application of bio-inspired solutions to design and prototype adaptive multi-mode mobile robots.

Nabil Aouf is currently the professor and Head of the Signals and Autonomy group, Centre for Electronic Warfare information and Cyber, Cranfield University, U.K. He has authored over 100 publications in high calibre in his domains of interest. His research interests are aerospace and defense systems, information fusion and vision systems, guidance and navigation, tracking, and control and autonomy of systems. He is an Associate Editor of the Imaging Science Journal.

Odysseas Kechagias-Stamatis received the M.Sc. degree in guided weapon systems and Ph.D. in 3D ATR for time-critical missile system applications from the Cranfield University, United Kingdom, in 2011 and 2017, respectively. His research interests include 2D and 3D object recognition, computer vision, robotics, and navigation systems.

James F. Whidborne (M'95-SM'10) received the B.A. degree in engineering from Cambridge University, and the M.Sc. and Ph.D. degrees in systems and control from the University of Manchester Institute of Science and Technology. From 1991 to 1994, he was a Research Associate with the Department of Engineering, University of Leicester. From 1994 to 2003, he was a Lecturer, then Senior Lecturer, with the Department of Mechanical Engineering, Kings College London. He is currently a Reader and Head of the Dynamics Simulation and Control Group, Cranfield University. His research interests include the theory and application of advanced control, including flight control and guidance, multiobjective robust control design, fluid flow control and finite precision controller implementation problems.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

https://dspace.org/

A furcated visual collision avoidance system for an autonomous micro robot

Isakhani, Hamid; Aouf, Nabil; Kechagias-Stamatis, Odysseas; Whidborne, James F.

2018-07-23
IEEE
https://doi.org/10.1109/TCDS.2018.2858742
Isakhani H, Aouf N, Kechagias-Stamatis O, Whidborne J. (2020) A furcated visual collision avoidance system for an autonomous micro robot. IEEE Transactions on Cognitive and Developmental Systems, Volume 12, March 2020, pp. 1-11

Downloaded from a DSpace repository. https://dspace.org/