

Mobile Robot with Movement Detection Controlled by a Real-Time Optical Flow Hermite Transform

Ernesto Moya-Albor, Jorge Brieva and Hiram Eredín Ponce Espinosa

Abstract This chapter presents a new algorithm inspired in the human visual system to compute optical flow in real-time based on the Hermite Transform. This algorithm is applied in a vision-based control system for a mobile robot. Its performance is compared for different texture scenarios with the classical Horn and Schunck algorithm. The design of the nature-inspired controller is based on the agent-environment model and agent's architecture. Moreover, a case study of a robotic system with the proposed real-time Hermite optical flow method was implemented for braking and steering when mobile obstacles are close to the robot. Experimental results showed the controller to be fast enough for real-time applications, be robust to different background textures and colors, and its performance does not depend on inner parameters of the robotic system.

Keywords Optical flow · Hermite transform · Movement detection · Navigation system · Differential method

1 Introduction

Movement detection and characterization in a scene is a relevant task in vision systems used in robotic applications controlled by visual features. The optical flow approach allows to obtain the displacement (magnitude and direction) of each pixel in the image in a given time. The quality and precision of these displacements in real time will determine the performance of the control system. The optical flow

E. Moya-Albor · J. Brieva (✉) · H.E. Ponce Espinosa
Faculty of Engineering, Universidad Panamericana, 03920 Mexico City, Mexico
e-mail: jbriaeva@up.edu.mx

E. Moya-Albor
e-mail: emoya@up.edu.mx

H.E. Ponce Espinosa
e-mail: hponce@up.edu.mx

problem has been treated widely in the literature. Some of them are easily implemented and extended to real time applications but with low accuracy, as [12], and others have fine implementations and high accuracy but most are impossible to use in real time applications by hardware limitations and high processing time, as [25] and [21]. In this chapter, we propose an algorithm in real time, using the optical flow constraint equation of Horn and Schunck [12], as the differential approach most used in real time applications, with improvements in accuracy due to the proposal using the Hermite transform. The Hermite transform is a biologically inspired image model, allowing fully describe the significant visual features in digital images, and together with the Horn and Schunck differential approach provide a real-time optical flow method more robust to noise and to intensity changes in the image sequence.

Computer vision can help to sense control variables in a navigation system. This kind of sensors is not conventional and is less used than others in navigation systems. Particularly, optical flow models can help to measure magnitude and direction displacement in a particular scene. In literature, there are several applications using optical flow especially in navigation problems: for aeronautical applications was proposed an aircraft maneuvering controlled by translational optical flow in [27], in [14] they used optical flow based on block-matching algorithm, image-based lane departure warning (LDW) system using the Lucas-Kanade optical flow and the Hough transform methods was proposed in [28], application to a lunar landing scenario using optical flow sensing was addressed in [30], for Power Assisted Wheelchair in [19] and Optical Flow Based Plane Detection for Mobile Robot Navigation in [32], and in [37] they used a controller design based on real time optical flow and image processing applied in a quadrotor and a Visual Tracking of a Moving Target by a camera mounted on a robot. Other applications not directly related to navigation systems are presented in literature as Counting Traffic [1], real time velocity estimation based on OF and disparity [11], and optical-flow-based altitude sensor and its integration with flapping-wing flying microrobot [7].

This chapter is organized as follows: Sect. 2 gives the basic concepts about the biological inspired model. The principles of optical flow methods are described in Sects. 3 and 4 presents our approach to compute optical flow in real time and some tests to validate our approach. Section 5 describes the implementation of our approach in the navigation of mobile robots. A conclusion is finally reported in Sect. 6.

2 Hermite Transform: A Biologically Inspired Model of Digital Images

This section introduces the Hermite Transform as a bio-inspired model for represent relevant perceptive features in digital images. First, we list the representative bio-inspired image models. Then, the basic concepts of the Hermite Transform and

its relationship with the human vision system is presented. Finally, we show a steered version of the Hermite Transform, that allows describe oriented patterns.

2.1 Bio-inspired Image Models

A large number of image processing algorithms has been inspired on the human visual perception systems. The principal task that these approaches must resolve is to realize a good description of the image to identify, reconstructing and tracking the objects in order to interpret the scene. Different methodologies can be divided into two general categories: those requiring strong prior knowledge and those requiring weak or no prior knowledge. In the second group the most important techniques are based on psychovisual human studies [29] and biological models [34]. The first one proposed by Gestalt Psychologists point out the importance of visual organization or perceptual grouping in the description of the scene, particularly the objects in the scene cannot be processed separately. They define concepts as proximity, continuity, homogeneity as principles to implement perception rules for processing the image. The main difficulty of these approaches is the implementation of the perception rules that must be particular to each application.

Approaches based on biological models seem more adequate for image processing due to their ability to generalize to all kind of scenes. They are inspired directly on the response of the retina and the ganglion cells to light stimuli (see Fig. 1) A first model proposed by Ernst Mach recognize that not just light intensities but intensity changes influence what we see (i.e., derivatives and sum of second derivatives in the space). The Gabor model, proposed by [26], represents the receptive fields of the visual cortex through Gaussian modulated with complex exponentials. Like the receptive fields, the Gabor functions are spatially local and consist of alternating bands of excitation and inhibition in a decaying envelope.

In 1987, Young [36] proposed a receptive field model based on the Gaussian and its derivatives. These functions, like the Gabor, are spatially local and consist of alternating regions of excitation and inhibition within a decaying envelope.

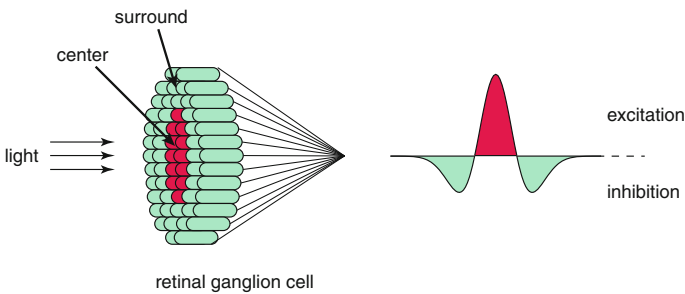


Fig. 1 Diagram of ganglion cells showing the biological inspiration for image processing

showed that Gaussian derivative functions more accurately model the measured receptive field data than the Gabor functions. These functions can be interpreted as the product of Hermite polynomials and a Gaussian window. It allows to decompose the image based on Hermite polynomials by defining a biological model of the measured receptive field data in the Human Vision System (HVS) [34–36].

The Hermite Transform uses a Gaussian window to localize and analyze the visual information where from a perceptual standpoint, the Gaussian window is a good model of the overlapping receptive fields found in physiological experiments [26]. Furthermore, the associated polynomials involve Gaussian derivative operators found in psychophysical models of the HVS [4, 17]. Finally, the operators used in the Hermite Transform provide a natural agreement with the theory scale-space [13] where the Gaussian window minimizes the uncertainty of the product of the spatial and frequency domain [33].

2.2 The Hermite Transform

The Hermite Transform is a special case of polynomial transform and was introduced by Martens in 1990 [18]. First, the original image $L(x, y)$ with coordinates (x, y) is located at various positions multiplying it by a window function $v^2(x - x_0, y - y_0)$ at positions (x_0, y_0) that conform a sampling lattice S . Then, the local information for each analysis window is expanded in terms of a family of orthogonal polynomials $G_{m,n-m}(x, y)$ where m and $(n - m)$ denote the analysis order in x and y direction respectively.

The Hermite Transform of an image can be computed by a convolution of the image $L(x, y)$ with the filter functions $D_{m,n-m}(x, y)$ obtaining the cartesian Hermite coefficients $L_{m,n-m}(x, y)$:

$$L_{m,n-m}(x_0, y_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} L(x, y) D_{m,n-m}(x_0 - x, y_0 - y) dx dy \quad (1)$$

$n = 0, 1, \dots, \infty \quad m = 0, 1, \dots, n$

The filter functions $D_{m,n-m}(x, y)$ are defined by polynomials $G_{m,n-m}(x, y)$ that are orthogonal with respect to an analysis window $v^2(x, y)$

$$D_{m,n-m}(x, y) = G_{m,n-m}(-x, -y) v^2(-x, -y), \quad (2)$$

where $v(x, y) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$ is a Gaussian window with normalization factor for a unitary energy for $v^2(x, y)$.

The polynomials orthonormal with respect to $v(x,y)^2$ can then be written as

$$G_{m,n-m}(x,y) = \frac{1}{\sqrt{2^n m!(n-m)!}} H_m\left(\frac{x}{\sigma}\right) H_{n-m}\left(\frac{y}{\sigma}\right), \tag{3}$$

where $H_n(x)$ are the Hermite polynomials given by Rodrigues' formula:

$$H_n(x) = (-1)^n \exp(x^2) \frac{d^n}{dx^n} \exp(-x^2) \quad n = 0, 1, 2, \dots \tag{4}$$

and $H_n\left(\frac{x}{\sigma}\right)$ represents the generalized Hermite polynomials with respect to the Gaussian function (with variance σ^2).

In Fig. 2, we show the Hermite polynomials $H_n(x)$ for $n = 0, 1, 2, 3, 4, 5$.

The corresponding analysis Hermite filters are separable both spatial and polar and they can be expressed by

$$D_{m,n-m}(x,y) = D_m(x)D_{n-m}(y), \tag{5}$$

where the one-dimensional Hermite filters can be computed by

$$D_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \frac{1}{\sigma\sqrt{\pi}} H_n\left(\frac{x}{\sigma}\right) \exp\left(-\frac{x^2}{\sigma^2}\right). \tag{6}$$

Fig. 2 The Hermite polynomials $H_n(x)$ for $n = 0, 1, 2, 3, 4, 5$

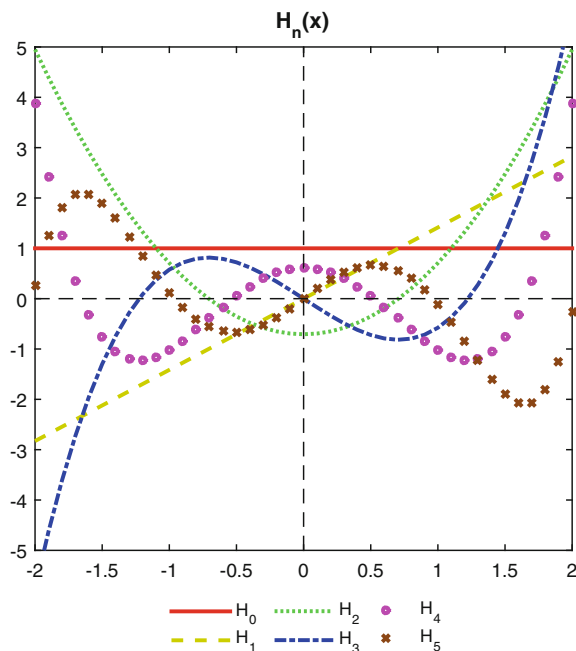


Figure 3a shows the Hermite filters $D_{m,n-m}(x, y)$ for $N = 3$ ($n = 0, 1, \dots, N$ and $m = 0, 1, \dots, n$).

In Fig. 4a, we obtained the cartesian Hermite coefficients of the *House* test image [10] for $N = 3$ ($n = 0, 1, 2, 3$).

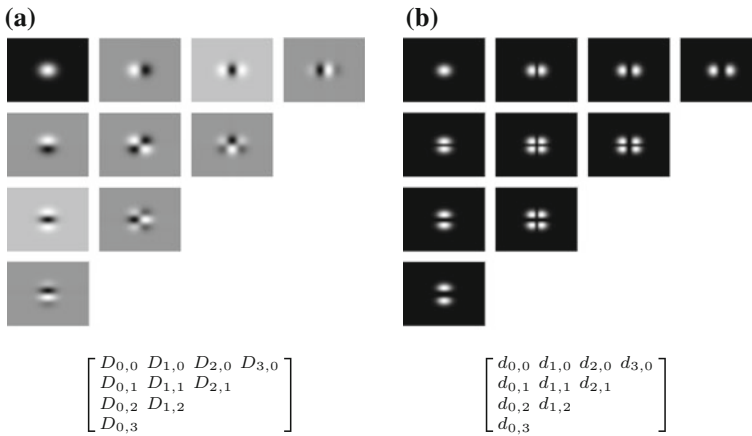


Fig. 3 **a** An ensemble of the Hermite filters $D_{m,n-m}(x, y)$ and **b** their Fourier transform spectrum $d_{m,n-m}(\omega_x, \omega_y)$ (right) for $N = 3$ ($n = 0, 1, \dots, N$ and $m = 0, 1, \dots, n$) [22]

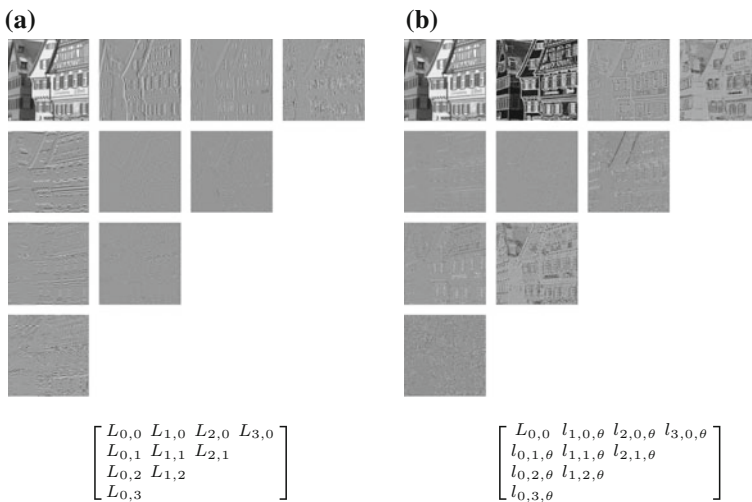


Fig. 4 **a** An ensemble of the cartesian Hermite coefficients and **b** the Steered Hermite coefficients of the *House* test image for $N = 3$ ($n = 0, 1, 2, 3$) [20]

2.3 The Steered Hermite Transform

The Steered Hermite Transform (ST) is a version of the cartesian Hermite coefficients that is obtained by rotating of the cartesian coefficients towards an estimated local orientation, according to a criterion of maximum oriented energy at each window position.

The Hermite filters $D_{m,n-m}(x,y) = D_m(x)D_{n-m}(y)$ are separable in space, and its Fourier transform can be expressed in polar coordinates. If $\omega_x = \omega \cos(\theta)$ and $\omega_y = \omega \sin(\theta)$, then

$$d_m(\omega_x)d_{n-m}(\omega_y) = g_{m,n-m}(\theta) \cdot d_n(\omega), \quad (7)$$

where $d_n(\omega)$ is the Fourier transform of each filter function, which expresses radial frequency selectivity of the n th derivative of the Gaussian but with a radial coordinate r for x , and $g_{m,n-m}(\theta)$ expresses the directional selectivity of the filter

$$g_{m,n-m}(\theta) = \sqrt{\binom{n}{m}} \cos^m \theta \cdot \sin^{n-m} \theta \quad (8)$$

The orientation feature of the Hermite filters explains why they are products of polynomials with a radially symmetric window function (Gaussian function). The $N + 1$ Hermite filters of order n form a steerable basis for each individual filter of order n . Filters of increasing order n analyze successively higher radial frequencies (Fig. 3b), and filters of the same order n and different (directional) index m distinguish between different orientations in the image. Note that the radial frequency selectivity $d_n(\omega)$ is the same for all $N + 1$ filters of order n and that these filters differ only in their orientation selectivity [18]. The resulting filters can be interpreted as directional derivatives of a Gaussian function.

For local 1D patterns, the Steered Hermite Transform provides a very efficient representation. This representation consists of a parameter θ that indicates the orientation of the pattern, and a small number of coefficients that represent the profile of the pattern perpendicular to its orientation. For a 1D pattern with orientation θ , the following relation holds:

$$l_{m,n-m,\theta}(x,y) = \begin{cases} \sum_{k=0}^n (L_{k,n-k}(x,y))(g_{k,n-k}(\theta)), & m = 0 \\ 0, & m > 0, \end{cases} \quad (9)$$

where $l_{m,n-m,\theta}(x,y)$ are Steered Hermite coefficient for the local angle θ .

This means that a Steered Hermite Transform offers a way to describe 1D patterns explicitly on the basis of their orientation and profile [31].

In Fig. 4b, we steer the cartesian Hermite coefficients of the corresponding coefficients of Fig. 4a according to maximum energy direction. It is noticeable that

the energy is concentrated in only three coefficients (first row of Fig. 4b), which represent the orientation of the different structures of the image.

3 Optical Flow Problem

This section describes the optical flow problem focused in the differential approaches used to obtain the displacements of the objects in a sequence of image. We present the seminal differential methods of Horn and Schunck, and Lucas and Kanade to highlight the advantages of global and local proposal. Later, we mention the additional constraints of recent approaches to improve robustness of classical methods.

3.1 Definition

The optical flow (OF) is defined as a two-dimensional distribution of apparent velocities that can be associated with variations of brightness patterns in a sequence of images [9]. The obtained vector field shows the displacement of each pixel in some time in the image sequence and represents the motion of objects in the scene or the relative motion of the sensor.

To illustrate the optical flow concept, we show in Fig. 5 the vector field obtained from the frames 43 and 44 of the *Cameramotion* sequence [15], where the apparent

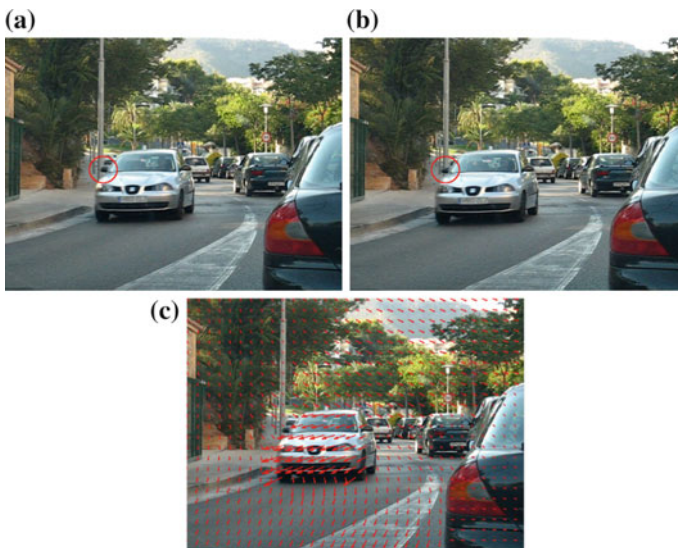


Fig. 5 Optical flow representation in a sequence of images [20]. **a** Frame 43 of the *Cameramotion* sequence. **b** Frame 44 of the *Cameramotion* sequence. **c** Optical flow result

motion of the car of the left is highlighted by the side view mirror within the red circle.

The vector of displacements obtained in two successive images can be observed by:

- The motion of objects in the scene.
- The relative movement between the observer and the scene.
- Variations in scene illumination.

Various methods have been proposed to calculate the optical flow in image sequences since 1980. Barron et al. [2] performed the first classification of the existing optical flow methods into four groups:

- Differential methods
- Region-based matching methods
- Energy-based methods
- Phased-based methods

The benchmarks of Barron et al. [2] in 1994 and Galvin et al. [8] in 1998 showed that the differential methods and phase-based methods were the techniques with better performance. Their study emphasizes the measurement accuracy and concludes that the most accurate methods are the differential proposals, where methods using global smooth constraint appear to produce visually attractive flow fields.

Differential techniques compute image velocity from spatiotemporal derivatives of image intensities or filtered versions of the image. The first approaches use first order derivatives and are based on image translation assuming intensity conservation. Second order differential methods use the Hessian of the intensity function to constrain 2D velocity. There are global and local first- and second-order methods, where global methods use the additional global constraint to compute dense optical flows over large image regions. Local methods use normal velocity information in local neighborhoods to perform a least squares minimization to find the best fit for the vertical and horizontal components of displacement [3].

3.2 Differential Optical Flow Methods

The differential methods are based on the work by Horn and Schunck (HS) [12], which proposes the *Constant Intensity Constraint* assuming that the intensities of the pixels of the objects remain constant:

$$L(X + W) - L(X) = 0, \quad (10)$$

where $L(X, t)$ is an image sequence, with $X = (x, y, t)^T$ representing the pixel location within a rectangular image domain Ω ; $W := (u, v, 1)^T$ is a vector that defined the displacement u and v of each a pixel at position (x, y) within the sequence of images at a time t to a time $(t + 1)$ in the directions x and y ; respectively.

Considering linear displacements (10) is expanded by Taylor's series, obtaining the *Optical Flow Constraint* equation

$$W^T(\nabla_3 L) = 0, \quad (11)$$

where $\nabla_3 L := (L_x, L_y, L_t)^T$ and $L_* := \frac{\partial L}{\partial *}$.

The optical flow constraint Eq. (11) incorporates the *Ill-posed Aperture Problem*, and it is not sufficient to determine the two unknown functions u and v uniquely, i.e., in homogeneous areas where movement is observed locally, only the normal component of the movement can be estimated. To overcome this problem, some additional constraints are required.

In 1981, Horn and Schunck [12] proposed the *Smoothness Constraint* that assumed that the apparent speed of the intensity pattern in the image varies smoothly; that is, neighboring points of the objects have similar velocities. To recover the optical flow the approach of Horn and Schunck minimizes a energy functional of the type

$$E_{HS}(W) = \int_{\Omega} (W^T(\nabla_3 L \nabla_3 L^T) W + \alpha |\nabla W|^2) dX. \quad (12)$$

The functional of (12) is a global differential method which allows to obtain dense vector fields. The first term is a data term requiring that the optical flow constraint equation is fulfilled, while the second term penalizes deviations from smoothness. The smoothness term $|\nabla W|^2 = |\nabla u|^2 + |\nabla v|^2$ is called *regularizer* and the positive smoothness weight α is the regularization parameter. One would expect that the specific choice of the regularizer has a strong influence on the result.

One classic local differential approach that minimizes (11) was proposed by Lucas and Kanade (LK) [16] in 1981. They consider the flow constant within a neighborhood ρ (Gaussian function K_ρ of standard deviation ρ), and determine the two constants u and v at a point (X, t) using a weighted least squares approach: a weighted least squares approach:

$$E_{LK}(W) = W^T J_\rho(\nabla_3 L) W, \quad (13)$$

where

$$J_\rho(\nabla_3 L) := K_\rho \otimes (\nabla_3 L \nabla_3 L^T) \quad (14)$$

and \otimes represents the convolution operator.

The local differential methods have the advantage of being robust against noise, but with the disadvantage that they do not get dense flows and an additional interpolation step is required to recover the displacement vector in each pixel of the scene.

To improve robustness to global differential methods, recent approaches have emerged and have suggested some additional constraints [6].

An immediate problem of (12) is that the intensity does not always remain constant from one image to another, therefore an independent intensity change measure is required. In [23, 24] a *Constant Gradient Constraint* (generally used in local methods to handle the aperture problem) is proposed:

$$\nabla L(X + W) - \nabla L(X) = 0. \quad (15)$$

Different approaches have been proposed to obtain accurate fields of displacements, one such methods is a combination of global (Horn and Schunck) and local (Lucas and Kanade) methods to generate dense flow fields, and robust against noise [5, 6].

Another proposal is a spacetime formulation performing a convolution with a three-dimensional Gaussian function and considering soft flows in the temporal direction. This improves robustness to noise and in those cases where the linearization of (10), and in consequence (11), is not valid for large displacements, multi-resolution strategies are used [5]. One option is to propose functionals that combine local constraints (intensity constraint and gradient constraint), a spatiotemporal smoothness constraint and a multi-scale approach [25].

In this sense, the Hermite Transform can be used to define a functional that performs a polynomial decomposition using the Steered Hermite Transform of two consecutive images in a short period of time. This decomposition represents the local characteristics of images from an perceptual approach within a multi-resolution scheme [21].

4 Real Time Hermite Optical Flow Method

This section defines the real-time optical flow method using the Hermite transform, the local constraints of Horn and Schunck approach are defined using the zero order and steered Hermite coefficients as local descriptors of visual features of the images.

4.1 Optical Flow Using the Hermite Transform

The main disadvantages of the recent differential methods for real time applications are their high computational time and difficult to implement. On the other hand, the Horn and Shunck [12] approach is a fast method with low implementation complexity. The main disadvantage of the Horn and Shunck method is its low accuracy. In those applications that require an approximation of the displacements, such method is sufficient in most cases.

We proposed a modified version of Horn and Shunck approach that allows to increase the accuracy of the optical flow by using the Hermite Transform as a biological image model. An expansion of the *Constant Intensity Constraint* (10),

with the incorporation of the *Steered Hermite Coefficient Constraint* of the Hermite Transform, is defined as follows

$$\left[L_0(X+W) - L_0(X) \right] + \gamma \left[\sum_{n=1}^N l_{n,\theta}(X+W) - \sum_{n=1}^N l_{n,\theta}(X) \right] = 0, \quad (16)$$

where γ is a weight parameter.

Equation (16) includes the local characteristics of the both images that represent the homogenous regions (low frequencies) in the zero order coefficients (L_0) and the edges, textures and complex structures (high frequencies) in the steered Hermite coefficients ($l_{n,\theta}$). In Fig. 6 we show the L_0 and $l_{n,\theta}$ Hermite coefficients from the image 42 and 43 of *Cameramotion* sequence, where there is a displacement (W) of the pixels in the position (X) between an image at time t and another image at time ($t+1$).

Considering linear displacements of (16) and expanded by Taylor series, we obtain the *Optical Flow Hermite Constraint* equation

$$\begin{aligned} & \left[u \frac{\partial L_0(X)}{\partial x} + v \frac{\partial L_0(X)}{\partial y} + \frac{\partial L_0(X)}{\partial t} \right] \\ & + \gamma \sum_{n=1}^N \left[u \frac{\partial l_{n,\theta}(X)}{\partial x} + v \frac{\partial l_{n,\theta}(X)}{\partial y} + \frac{\partial l_{n,\theta}(X)}{\partial t} \right] = 0. \end{aligned} \quad (17)$$

The one-dimensional spatial derivatives can be reduced considering that the one-dimensional Hermite coefficients are achieved by the inner product between the signal located by the Gaussian window and the Hermite polynomials:

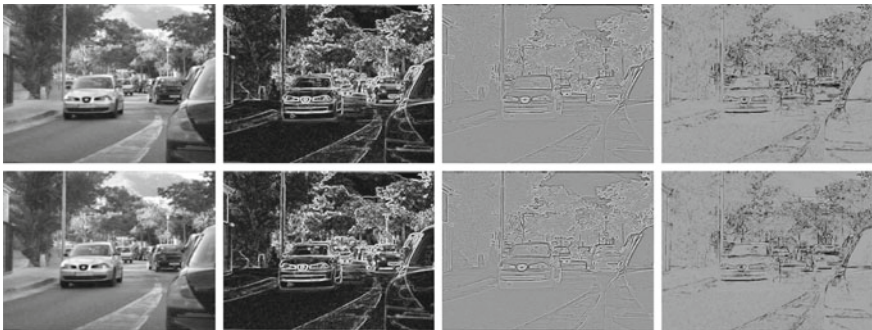


Fig. 6 Hermite coefficients from images 42 (*first row*) and 43 (*second row*) of *Cameramotion* sequence [20]. In first row are shown the coefficients $L_{0,0}(X)$, $l_{1,\theta}(X)$, $l_{2,\theta}(X)$ and $l_{3,\theta}(X)$ at time t . Second row shows the coefficients $L_{0,0}(X+W)$, $l_{1,\theta}(X+W)$, $l_{2,\theta}(X+W)$ and $l_{3,\theta}(X+W)$ at time $t+1$

$$L_k = \left\langle L(x), H_k \left(\frac{x}{\sigma} \right) \right\rangle. \quad (18)$$

It can be demonstrated that the derivative of the Hermite coefficients holds [21]:

$$L_k = L^{(k)}(x) = \frac{\partial^k L(x)}{\partial x^k}, \quad (19)$$

and therefore,

$$\begin{aligned} \frac{\partial L_0(X+W)}{\partial X} &= L_{1,0}(X+W), \\ \frac{\partial}{\partial X} \sum_{n=1}^N l_{n,\theta}(X) &= \sum_{n=1}^N l_{n,\theta_{(m)+1}}(X), \end{aligned} \quad (20)$$

and in a similar way for the derivatives of y .

The *Optical Flow Hermite Constraint* equation can be rewritten as:

$$\begin{aligned} &\left[uL_{0,1}(X) + vL_{1,0}(X) + \frac{\partial L_0(X)}{\partial t} \right] \\ &+ \gamma \sum_{n=1}^N \left[ul_{n,\theta_{(m)+1}}(X) + vl_{n,\theta_{(n)+1}}(X) + \frac{\partial l_{n,\theta}(X)}{\partial t} \right] = 0, \end{aligned} \quad (21)$$

where:

$$l_{n,\theta_{(m)+1}}(X) = \sum_{n=1}^N L_{(m)+1,n-m}(X) \cdot g_{m,n-m}(\theta), \quad (22)$$

$$l_{n,\theta_{(n)+1}}(X) = \sum_{n=1}^N L_{m,(n+1)-m}(X) \cdot g_{m,n-m}(\theta). \quad (23)$$

The original functional of Horn and Schunck [12] (12) can be expressed using (21) to define the real-time Hermite optical flow (RT-HOF) as follows

$$\begin{aligned} E_{RT-HOF}(W) &= \int_{\Omega} \left[\left(uL_{0,1}(X) + vL_{1,0}(X) + \frac{\partial L_0(X)}{\partial t} \right) \right. \\ &\quad \left. + \gamma \sum_{n=1}^N \left(ul_{n,\theta_{(m)+1}}(X) + vl_{n,\theta_{(n)+1}}(X) + \frac{\partial l_{n,\theta}(X)}{\partial t} \right) \right. \\ &\quad \left. + \alpha |\nabla W|^2 \right] dX. \end{aligned} \quad (24)$$

Using variational calculus, the corresponding Euler-Lagrange equations are:

$$\begin{aligned} & \left[L_{0,1}^2(X) + \gamma L_{0,1}(X) \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right] u \\ & + \left[L_{0,1}(X)L_{1,0}(X) + \gamma L_{0,1}(X) \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right] v \quad (25) \\ & = \alpha^2 \nabla^2 u - \left[L_{0,1}(X)L_{0r}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_r}(X)) \right], \end{aligned}$$

$$\begin{aligned} & \left[L_{0,1}(X)L_{1,0}(X) + \gamma L_{1,0}(X) \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right] u \\ & + \left[L_{1,0}^2(X) + \gamma L_{1,0}(X) \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right] v \quad (26) \\ & = \alpha^2 \nabla^2 v - \left[L_{1,0}(X)L_{0r}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_r}(X)) \right], \end{aligned}$$

where $L_{0r}(X) = \frac{\partial L_0(X)}{\partial r}$, $l_{n,\theta_r}(X) = \frac{\partial l_{n,\theta}(X)}{\partial r}$ and $\nabla^2 u$ represents the Laplacian of u .

Finding the solution of (25, 26) for u and v and applying Gauss-seidel iterative method, the final equations hold

$$\begin{aligned} u^{n+1} = \bar{u}^n - L_{0,1}(X) & \left[\left(L_{0,1}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right) \bar{u}^n \right. \\ & + \left(L_{1,0}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right) \bar{v}^n \\ & \left. + L_{0r}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_r}) \right] / \quad (27) \\ & \left[\alpha^2 + \left(L_{0,1}^2(X) + \gamma L_{0,1}(X) \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right) \right. \\ & \left. + \left(L_{1,0}^2(X) + \gamma L_{1,0}(X) \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right) \right], \end{aligned}$$

$$\begin{aligned}
 v^{n+1} = \bar{v}^n - L_{1,0}(X) & \left[\left(L_{0,1}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right) \bar{u}^n \right. \\
 & + \left(L_{1,0}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right) \bar{v}^n \\
 & \left. + L_{0t}(X) + \gamma \sum_{n=1}^N (l_{n,\theta_t}) \right] / \\
 & \left[\alpha^2 + \left(L_{0,1}^2(X) + \gamma L_{0,1}(X) \sum_{n=1}^N (l_{n,\theta_{(n)+1}}(X)) \right) \right. \\
 & \left. + \left(L_{1,0}^2(X) + \gamma L_{1,0}(X) \sum_{n=1}^N (l_{n,\theta_{(m)+1}}(X)) \right) \right], \tag{28}
 \end{aligned}$$

where the Laplacian was approximated by $\nabla u \approx k(\bar{u}_{i,j} - u_{i,j})$ and the local average was defined by performing a convolutions of u with the kernel

$$\begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}.$$

To compare the real-time Hermite optical flow with the proposal of Horn and Schunck, both methods were implemented in MATLAB[®].

For the Horn and Schunck method, the solution of (12) is given by [12]:

$$u^{n+1} = \bar{u}^n - [L_x(X)(L_x(X)\bar{u}^n + L_y(X)\bar{v}^n + L_t(X))] / \tag{29}$$

$$[\alpha^2 + L_x^2(X) + L_y^2(X)]$$

$$v^{n+1} = \bar{v}^n - [L_y(X)(L_x(X)\bar{u}^n + L_y(X)\bar{v}^n + L_t(X))] / \tag{30}$$

$$[\alpha^2 + L_x^2(X) + L_y^2(X)].$$

In Algorithms 1 and 2, we show the pseudocode for the iterative solution by Gauss-Seidel of the Horn and Schunck and real-time Hermite optical flow methods respectively.

Algorithm 1 Horn and Schunck's optical flow estimation.

```

1: procedure HORN AND SCHUNCK OPTICAL FLOW METHOD
2:    $\alpha \leftarrow$  Smoothing parameter
3:    $Num\_of\_itera \leftarrow$  Number of iterations
4:    $img1 \leftarrow$  Imagen in time  $t$ 
5:    $img2 \leftarrow$  Imagen in time  $t + 1$ 
6:    $u \leftarrow$  Matrix of zeros of size of  $img1$ 
7:    $v \leftarrow$  Matrix of zeros of size of  $img1$ 
8:    $G \leftarrow$  Square Gaussian filter of size  $\sigma/6$ 
9:    $img1 = img1 \otimes G$  ▷ Smoothing images  $img1$  and  $img2$ 
10:   $img2 = img2 \otimes G$ 
11:   $L_x = \frac{1}{4} \left( img1 \otimes \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} + img2 \otimes \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \right)$  ▷ Compute spatiotemporal derivatives
12:   $L_y = \frac{1}{4} \left( img1 \otimes \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} + img2 \otimes \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \right)$ 
13:   $L_t = \frac{1}{4} \left( img1 \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + img2 \otimes \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \right)$ 
14:  for  $i = 1$  to  $Num\_of\_itera$  do
15:     $\bar{u} = u \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$  ▷ Compute local averages of the flow
16:     $\bar{v} = v \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$ 
17:    Update  $u$  and  $v$  using the Eq. (29) and (30)
18:  end for
19:  return  $u, v$ 
20: end procedure

```

Algorithm 2 Real-time Hermite optical flow estimation.

```

1: procedure REAL-TIME HERMITE OPTICAL FLOW METHOD
2:    $\alpha \leftarrow$  Smoothing parameter
3:    $\gamma \leftarrow$  Weight parameter
4:    $Num\_of\_itera \leftarrow$  Number of iterations
5:    $N \leftarrow$  Maximum Hermite polynomial degree
6:    $M \leftarrow$  Square Gaussian window size ( $M + 1$ )
7:    $img1 \leftarrow$  Imagen in time  $t$ 
8:    $img2 \leftarrow$  Imagen in time  $t + 1$ 
9:    $u \leftarrow$  Matrix of zeros of size of  $img1$ 
10:   $v \leftarrow$  Matrix of zeros of size of  $img1$ 
11:  Compute the analysis Hermite filters  $D_{m,n-m}$  from Eq. (5) and (6):
12:  for  $n = 0$  to  $N$  do
13:    for  $m = 0$  to  $n$  do
14:       $L_{m,n-m} = img1 \otimes D_{m,n-m}$   $\triangleright$  Cartesian Hermite coefficients  $img1$ 
15:       $Lw_{m,n-m} = img2 \otimes D_{m,n-m}$   $\triangleright$  Cartesian Hermite coefficients  $img2$ 
16:       $\theta = \arctan \frac{L_{0,1}}{L_{1,0}}$   $\triangleright$  Phase of the gradient of Hermite coefficients  $img1$ 
17:       $\theta w = \arctan \frac{Lw_{0,1}}{Lw_{1,0}}$   $\triangleright$  Phase of the gradient of Hermite coefficients  $img2$ 
18:       $g_{m,n-m}(\theta) = \sqrt{\binom{n}{m}} \cos^m \theta \cdot \sin^{n-m} \theta$   $\triangleright$  Angle functions  $img1$ 
19:       $gw_{m,n-m}(\theta w) = \sqrt{\binom{n}{m}} \cos^m \theta w \cdot \sin^{n-m} \theta w$   $\triangleright$  Angle functions  $img2$ 
20:       $l_{n,\theta} = \sum \sum_{n=1}^N \left( L_{m,n-m} \right) \left( g_{m,n-m}(\theta) \right)$   $\triangleright$  ST coefficients  $img1$ 
21:       $l_{n,\theta_{(m)+1}}(X) = \sum_{n=1}^N L_{(m)+1,n-m}(X) \cdot g_{m,n-m}(\theta)$ 
22:       $l_{n,\theta_{(n)+1}}(X) = \sum_{n=1}^N L_{m,(n+1)-m}(X) \cdot g_{m,n-m}(\theta)$ 
23:       $lw_{n,\theta} = \sum \sum_{n=1}^N \left( Lw_{m,n-m} \right) \left( gw_{m,n-m}(\theta) \right)$   $\triangleright$  ST coefficients  $img2$ 
24:    end for
25:  end for
26:   $L_{0t} = Lw_{0,0} - L_{0,0}$   $\triangleright$  Temporal derivatives
27:   $l_{n,\theta_t} = \sum_{n=1}^N lw_{n,\theta w} - \sum_{n=1}^N l_{n,\theta}$ 
28:  for  $i = 1$  to  $Num\_of\_itera$  do
29:     $\bar{u} = u \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$   $\triangleright$  Compute local averages of the flow
30:     $\bar{v} = v \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$ 
31:    Update  $u$  and  $v$  using the Eq. 27 and 28
32:  end for
33:  return  $u, v$ 
34: end procedure

```

5 Implementation on Navigation Mobile Robots

This section describes a case study in which a mobile robot implements a movement detection controller based on the Hermitian optical flow approach presented earlier. For experimental purposes, this controller was developed and implemented in a LEGO[®] robot with a mounted webcam, and the whole system communicates with MATLAB which computes the control system. In that sense, the whole robotic system is introduced in this section as well as the design of the control law.

5.1 Description of the System

In this case study, a mobile robot was implemented with the LEGO[®] Mindstorms EV3 as both the mechanical and the electrical platform because it is easy and practical to use. In particular, the mobile robot was built as in a tank configuration using two direct current (DC) motors and two rubber bands as actuators, and the robot was planned to be in a differential steering configuration. In addition, a webcam was mounted on the robot to be used as the sensor. For experimental purposes, the intelligent brick of the LEGO[®] Mindstorms EV3 set was used as an interface between the mobile robot and a computer with MATLAB[®], as the latter was employed to compute the control of the whole system. Both the webcam and the intelligent brick communicated with the computer via USB. Figure 7 shows a diagram of the robotic system. The technical specifications of the robotic system are summarized in Table 1.

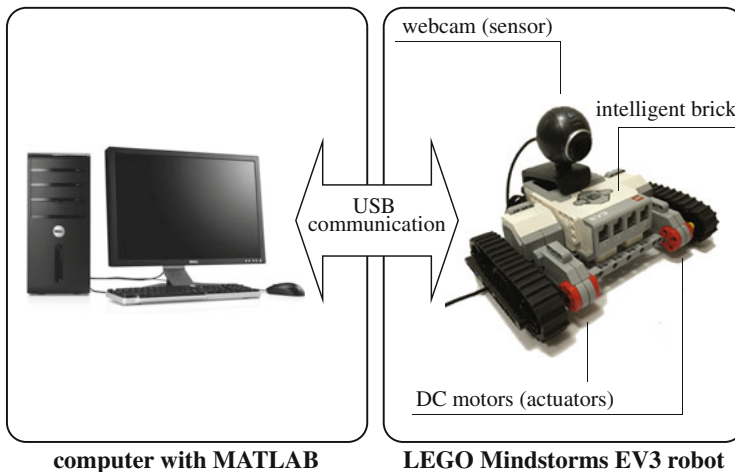


Fig. 7 Diagram of the robotic system employed in the case study

Table 1 Technical specifications of the robotic system

Component	Description
Webcam sensor	1.3 M pixels, 1280 × 1024 max resolution
DC motor actuators	9 V-input, 0.55 A nominal, 2.03 W mechanical power, 4.95 W electrical power, 160 rpm max nominal
USB communication	9600 baud rate
Windows-based workstation	Intel® Xeon® six-core processor 2.6 GHz, 32 GB RAM

5.2 Characterization of Hermite Optical Flow Method in Robot Navigation

In order to develop a nature-inspired control law for robot navigation, a characterization of the real-time Hermite optical flow method was performed. This characterization considered a description of the performance of the RT-HOF in terms of the displacement values measured between two images. Two experiments were conducted: a displacement measurement using RT-HOF without timeout limitations between frames (soft real-time system), and the second experiment measures the displacement values using RT-HOF with hard timeout deadlines between frames (hard real-time system).

5.2.1 Characterization in a Soft Real-Time Hermite Optical Flow System

The first experiment for characterizing the proposed real-time Hermite optical flow algorithm was developed under a soft real-time framework avoiding any deadline times, i.e., the software is free to use unlimited time for computing displacements between two images. In the experiment, the LEGO® robot was located statically in front of a reconfigurable wall. That wall worked as the background of the scene, with four possible configurations: white, dry-leaves, blue-square, and red-stripes backgrounds, as shown in Fig. 8. Then, another LEGO® robot started to move in front of the main robot with a certain speed (also configurable in three possible states: low, medium and hard). A short video was recorded using the same webcam mounted on the main robot.

For analysis purposes, a test point marked in red color was previously located at the middle of the secondary robot. Then, manual segmentation was done to find the test point in every frame of the short video. After that, the RT-HOF algorithm measured the displacement of the test point in the whole video. Figure 9 shows the mean of magnitude and phase of the displacement at the test point for some representative cases, best and worst results, of the secondary robot using all backgrounds. In Tables 2 and 3, we show the results in magnitude and phase of the

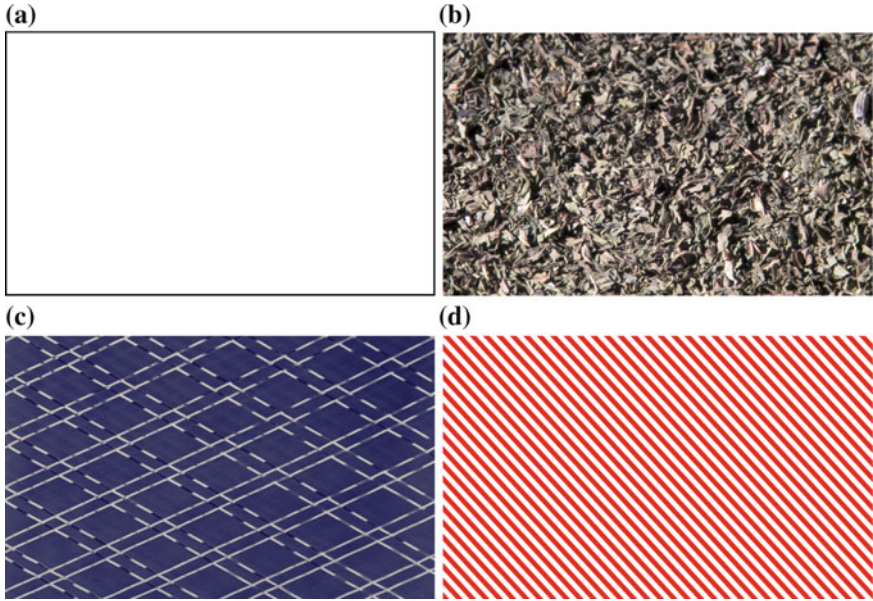


Fig. 8 Examples of backgrounds used in the experiments: **a** white, **b** dry leaves, **c** blue squares, **d** red stripes

vector of displacement at the test point for each speed case and for each background. Notice that the displacement of the test point relatively measures the displacement of the whole secondary robot (e.g., as a dynamic obstacle). It is evident from Fig. 9 that the RT-HOF method can distinguish different levels of speed in mobile obstacles in both plain and texturized backgrounds without any difficulties. For comparison purposes, the Horn method was also computed as seen in Fig. 9.

5.2.2 Characterization in a Hard Real-Time Hermite Optical Flow System

The second experiment for characterizing the proposed RT-HOF method was developed under a hard real-time framework with a time rate of 100 ms. The same scene was occupied with four different backgrounds (see Fig. 8) and three levels of speed in the dynamic obstacle. However, an automated segmentation was employed to extract only the measured displacements of the dynamic obstacle online (secondary robot), and then the mean displacement (magnitude and angle) was obtained for each pair of frames in the webcam streaming. Figure 10 shows some the mean displacement at the mobile obstacle for each speed case using all background configurations, and Tables 4 and 5 summarizes the whole experimental results.

Some conclusions can be drawn from Figs. 9 and 10, and Tables 4 and 5. First, the Horn-based method generates a greater angle than Hermite-based method.

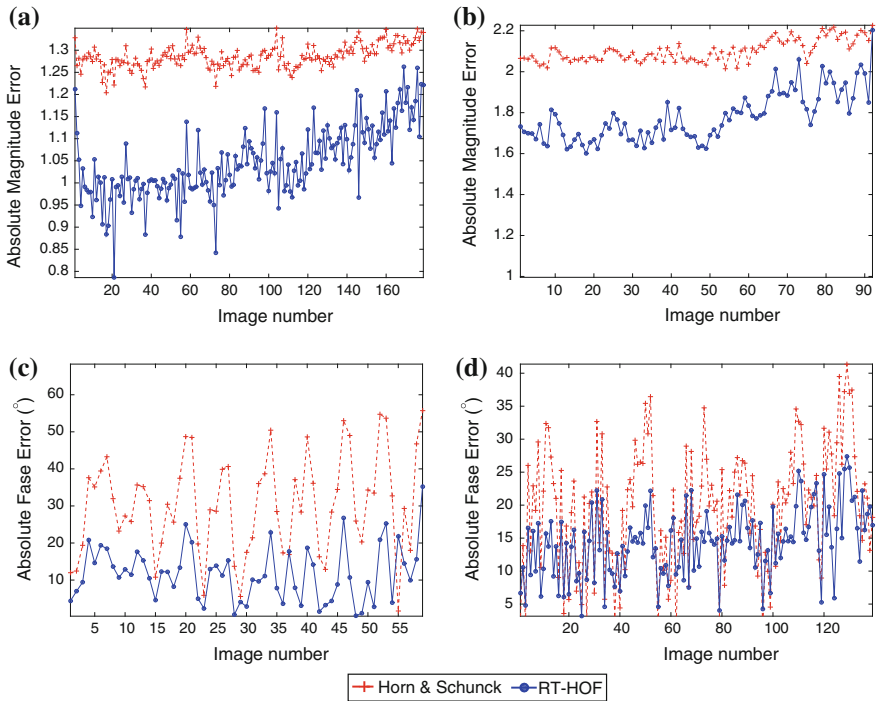


Fig. 9 Displacement characterization for a soft real-time system, best and worst results for the mean of magnitude and phase: **a** best absolute magnitude error with red stripes background at speed one, **b** worst absolute magnitude error with dry leaves background at speed two, **c** best absolute phase error with blue squares background at speed three, **d** worst absolute phase error with squares background at speed three

Table 2 Displacement characterization in magnitude for a soft real-time system

Background	Method	Mean magnitude error $\mu \pm \sigma$ [px]		
		m_1	m_2	m_3
White	Horn	1.34 ± 0.02	1.94 ± 0.44	2.16 ± 0.63
	Hermite	1.10 ± 0.07	1.64 ± 0.36	1.86 ± 0.55
Dry-leaves	Horn	1.28 ± 0.03	1.70 ± 0.41	1.85 ± 0.58
	Hermite	1.09 ± 0.07	1.46 ± 0.33	1.66 ± 0.54
Blue squares	Horn	1.18 ± 0.04	1.62 ± 0.38	1.78 ± 0.53
	Hermite	1.00 ± 0.09	1.34 ± 0.28	1.55 ± 0.47
Red stripes	Horn	1.90 ± 0.56	1.75 ± 0.41	1.90 ± 0.56
	Hermite	1.04 ± 0.08	1.42 ± 0.32	1.60 ± 0.48

While Horn-based method can distinguish large speeds in both white and blue-squares backgrounds, it cannot recognize large speeds in both dry-leaves and red-stripes backgrounds significantly. In addition, Hermite-based method can

Table 3 Displacement characterization in phase for a soft real-time system

Background	Method	Mean angle error $\mu \pm \sigma$ [°]		
		θ_1	θ_2	θ_3
White	Horn	19.61 ± 9.31	30.79 ± 14.45	34.79 ± 15.53
	Hermite	14.39 ± 5.24	13.09 ± 6.70	14.07 ± 7.33
Dry-leaves	Horn	16.61 ± 9.94	30.17 ± 19.23	32.23 ± 20.91
	Hermite	15.55 ± 5.38	17.98 ± 7.47	18.57 ± 10.00
Blue squares	Horn	14.85 ± 16.17	22.93 ± 15.04	24.82 ± 16.05
	Hermite	16.17 ± 5.87	15.26 ± 6.90	14.57 ± 7.84
Red stripes	Horn	15.25 ± 9.94	25.75 ± 16.04	28.32 ± 2.11
	Hermite	13.81 ± 5.23	13.90 ± 6.44	13.82 ± 7.526

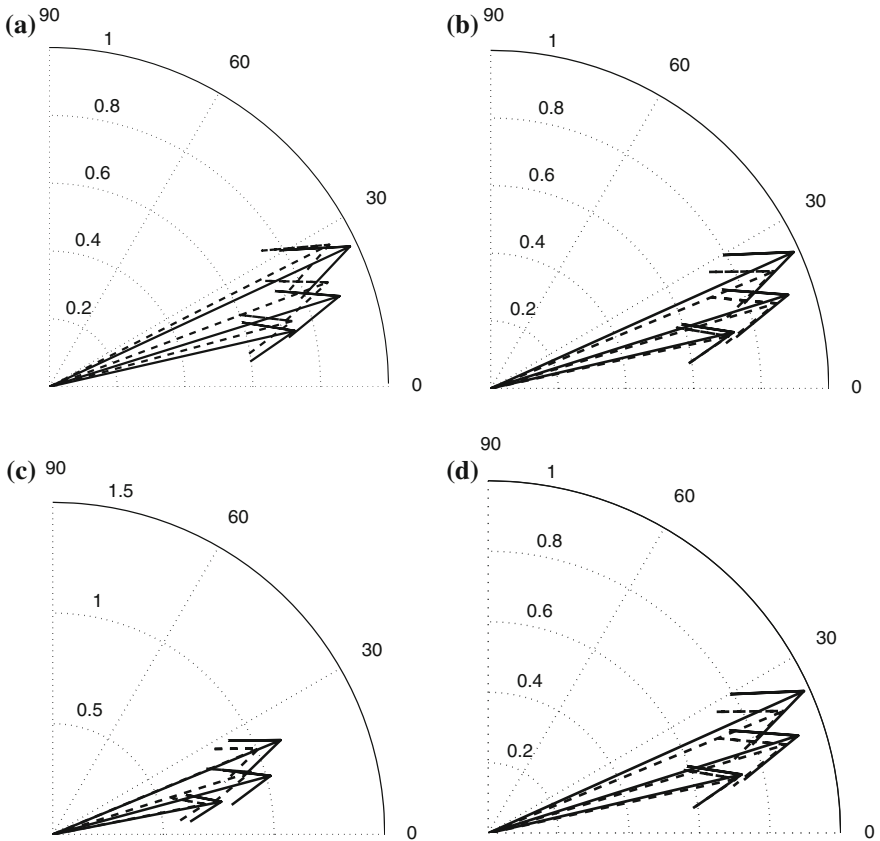


Fig. 10 Examples of the displacement characterization for a hard real-time system: **a** white background, **b** dry-leaves background, **c** blue squares background, **d** red stripes background. (*strong-line*) Hermite-based method, (*dashed-line*) Horn-based method. *Arrows* represent vectors of displacement

Table 4 Displacement characterization in magnitude for a hard real-time system

Background	Method	Magnitude $\mu \pm \sigma$ [px/s]		
		m_1	m_2	m_3
White	Horn	0.74 ± 0.09	0.88 ± 0.08	0.93 ± 0.06
	Hermite	0.74 ± 0.07	0.90 ± 0.09	0.98 ± 0.05
Dry-leaves	Horn	0.72 ± 0.063	0.88 ± 0.05	0.91 ± 0.07
	Hermite	0.74 ± 0.062	0.92 ± 0.04	0.98 ± 0.05
Blue squares	Horn	0.70 ± 0.06	0.94 ± 0.10	1.01 ± 0.05
	Hermite	0.78 ± 0.07	1.02 ± 0.04	1.11 ± 0.02
Red stripes	Horn	0.69 ± 0.07	0.86 ± 0.08	0.92 ± 0.040
	Hermite	0.74 ± 0.08	0.90 ± 0.11	1.00 ± 0.042

Table 5 Displacement characterization in phase for a hard real-time system

Background	Method	Angle $\mu \pm \sigma$ [°]		
		θ_1	θ_2	θ_3
White	Horn	15.10 ± 3.19	20.56 ± 4.27	26.90 ± 5.21
	Hermite	12.84 ± 1.89	17.33 ± 3.96	25.02 ± 4.43
Dry-leaves	Horn	12.38 ± 3.20	16.60 ± 3.47	22.47 ± 6.56
	Hermite	13.04 ± 2.34	17.48 ± 3.09	24.23 ± 2.38
Blue squares	Horn	11.22 ± 2.82	17.47 ± 4.16	22.65 ± 4.38
	Hermite	10.99 ± 2.11	15.05 ± 1.94	22.47 ± 4.08
Red stripes	Horn	15.33 ± 3.57	20.68 ± 2.97	27.41 ± 4.96
	Hermite	14.87 ± 2.66	19.06 ± 2.48	27.54 ± 3.59

distinguish between speeds in any background. The overall results of these experiments are that real-time Hermite optical flow can be used as a nature-inspired control law because its performance can distinguish between different levels of speeds in dynamic obstacles, and it responds well in hard real-time framework.

5.3 Design of the Nature-Inspired Control Law

The LEGO[®] robot was controlled using a nature-inspired control law, based on the real-time Hermite optical flow method. Following, we describe the methodology to design the nature-inspired controller.

This work adopted the agent-environment cycle methodology for designing the control law in the robot, as shown in Fig. 11. This aims that the robot perceives the state of the environment (e.g., the scene in which the robot is interacting), then a specific robot architecture defines how this perception is interpreted to finally get a decision of the type of action it will perform in the environment. Once the action is selected, the robot executes it and the agent-environment cycle assumes that the

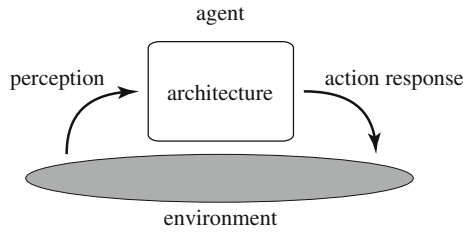
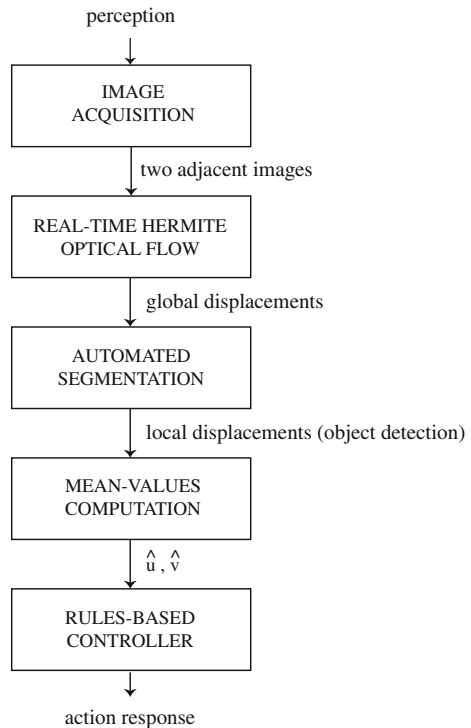


Fig. 11 Agent-environment cycle

state of the environment changes, giving the opportunity to the robot to start the process again.

Thus, the design of the nature-inspired control law in this work refers to find a suitable robot architecture. In Fig. 12 the final design of the robot architecture used in this case study is depicted. As shown, the robot starts the process acquiring images at 100 ms of frame rate. Then, a pair of two adjacent images are analyzed using the real-time Hermite optical flow. A map of the displacements is then obtained showing where objects in images are moving. Next, an automated segmentation is done in order to get only the displacements that reaches an empirical threshold, assuming that dynamic obstacles perform displacements greater than this

Fig. 12 Diagram of the nature-inspired controller used in the case study



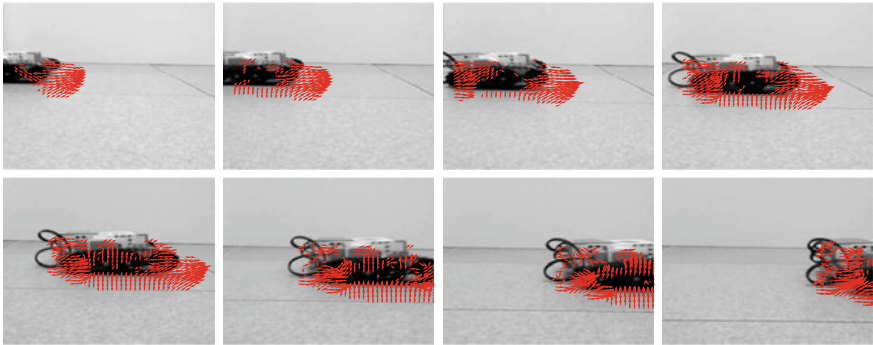


Fig. 13 An example of the field observed by the robot

threshold. For this case study, a threshold value of $T_{seg} = 0.5$ was selected (using a trial-and-error approach). In addition, this case study supposes that there is one dynamic obstacle in front of the LEGO[®] robot at most. Thus, the automated segmentation finds the dynamic obstacle and computes its local displacements.

In Fig. 13, we show an example of the field of displacements segmented observed by the LEGO[®] robot when it moves and other robot approximates.

Later on, the robot computes the mean value \hat{u} of the horizontal component of displacements and the mean value \hat{v} of the vertical component of the displacements. Also, the resultant angle $\hat{\theta}$ of the displacements is computed. All these values give to the robot some advice about the movement of the dynamic object in the two dimensions projected on the image. Then, a rule-based controller was designed in terms of the mean values \hat{u} and \hat{v} and the angle $\hat{\theta}$. Algorithm 3 shows the prototype of the rules-based controller. As noted, it requires two thresholds T_u and T_v associated to the horizontal and vertical components of the dynamic obstacle's displacement.

Algorithm 3 Rules-based controller used in the case study.

```

1: if  $\|\hat{u}\| > T_u$  and  $\|\hat{v}\| > T_v$  then
2:   if  $\|\hat{\theta}\| > T_\theta$  then
3:     braking or steering-to-the-right
4:   else if  $\|\hat{\theta}\| < T_\theta$  and  $\|\hat{\theta}\| \neq 0$  then
5:     braking or steering-to-the-left
6:   else
7:     go-forward
8:   end if
9: else
10:  go-forward
11: end if

```

Two experiments were conducted to find the thresholds T_u and T_v for the rules-based controller. The first experiment consisted on having a fixed obstacle in front of the LEGO[®] robot, and this one moves straight until reached the fixed

obstacle. The webcam mounted in the robot acquired images at 100 ms of frame rate. The real-time Hermite optical flow with automated segmentation obtained the mean values \hat{u} and \hat{v} during the translation. Then both mean values were plotted as shown in Fig. 14. As noted, when the robot is far from the obstacle the \hat{v} value is small, when the robot is close to the obstacle the \hat{v} value is large (slope of 0.02 px/s). The \hat{u} value is not affected (slope of 7.1×10^{-4} px/s). The same experiment was repeated but now the fixed obstacle was replaced with a dynamic obstacle with constant speed. Results of this experiment are shown in Fig. 15. Notice that when \hat{u} is minimum and \hat{v} is maximum, the mobile obstacle is placed in front of the robot. From the latter figures it can be seen that the most appropriate threshold values are $T_u = 0.6$ px and $T_v = 0.5$ px.

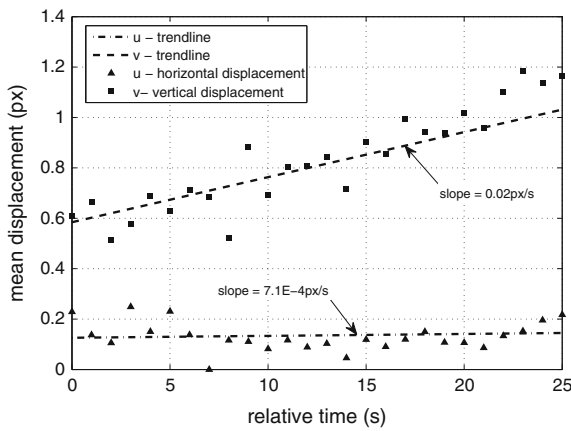


Fig. 14 Experiment 1 for extracting the threshold values of the components in displacements

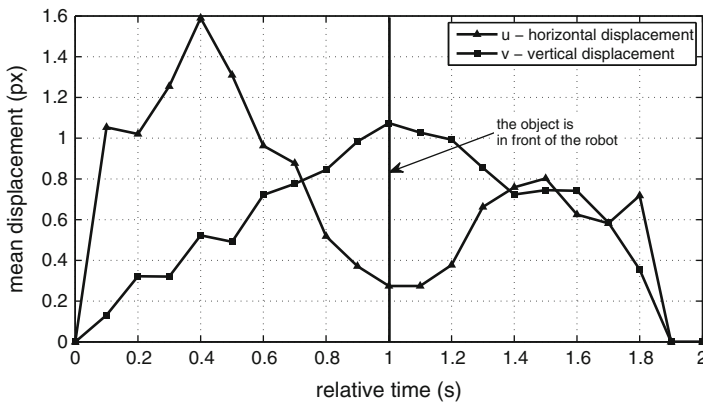


Fig. 15 Experiment 2 for extracting the threshold values of the components in displacements

In addition, a threshold T_θ is needed for determining the direction of the mobile obstacle. In this case study, a value of $T_\theta = 40^\circ$ was used (obtained by trial-and-error). At last, the robot takes a decision from the rules-based controller, picking the more suitable action. For this case study three possible actions were designed: *go-forward*, *braking*, or *steering*. Once the LEGO[®] robot performs the action, the whole process is repeated again.

5.4 Experimental Results for Dynamic Obstacles Avoidance

This case study implemented the proposed real-time Hermite optical flow method as a nature-inspired controller for dynamic obstacles avoidance in robots (Fig. 12). In particular, a LEGO[®] robot with a webcam, as the unique sensor, was employed. Two experiments were performed as follows: a braking action and a steering action.

5.4.1 Braking Action

This experiment consisted on braking the LEGO[®] robot when a mobile obstacle is presented in front of it. Once the obstacle moves on, the robot can go straight again. Figure 16 show the trajectories of the robot in that situation with three possible constant speeds of the mobile obstacle: *low*, *medium* and *high*. It is evident from Fig. 16 that in all cases the LEGO[®] robot can stop without colliding with the obstacle.

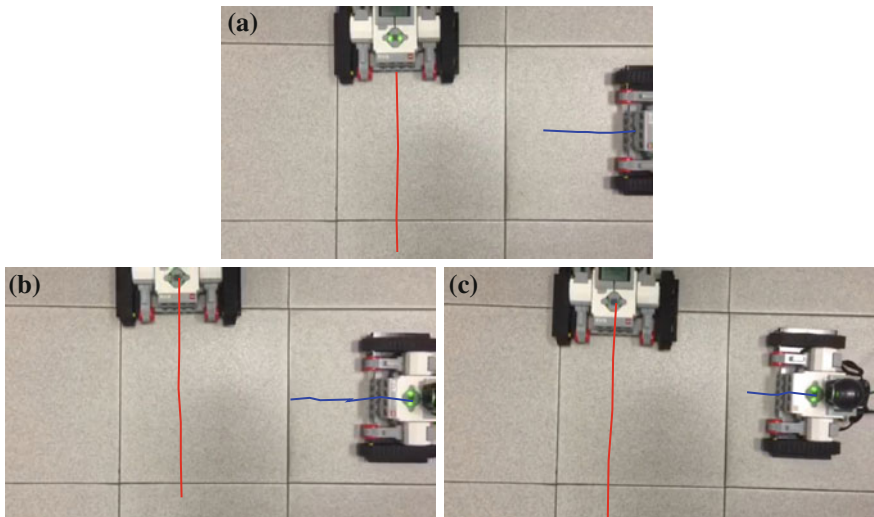


Fig. 16 An example of the trajectories and evidence of the braking actions in the robot, using a mobile obstacle with **a** *low*, **b** *medium* and **c** *high* speed

In Fig. 17 we show the image sequence of the trajectory followed for the LEGO® robot for the *medium* speed, where the braking action is observed in the three images of the central row.

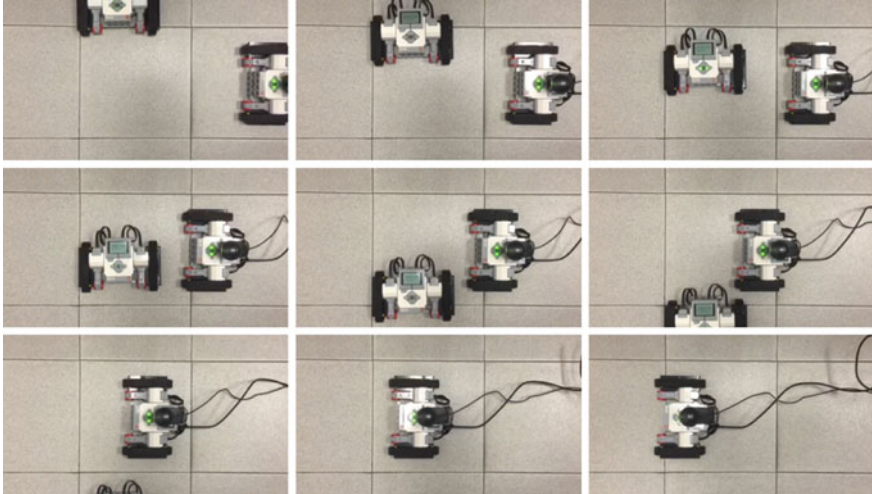


Fig. 17 Image sequence of the trajectory of robot for braking action in *medium* speed

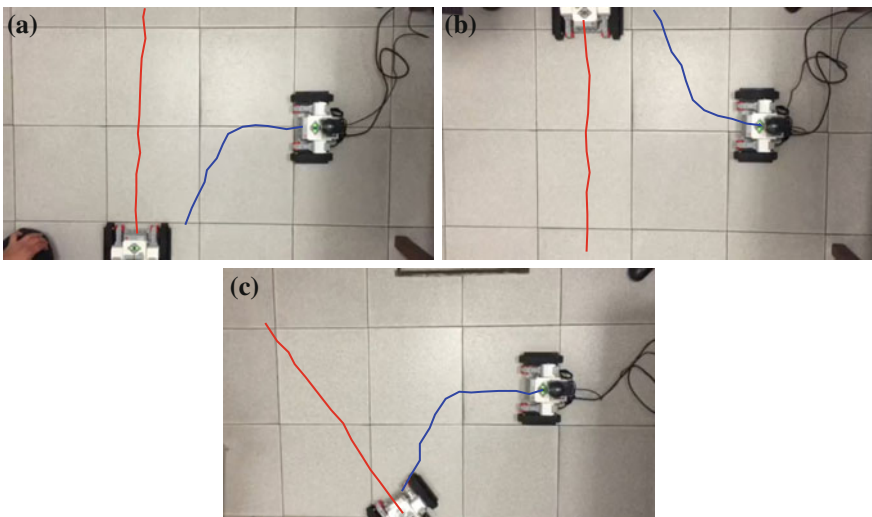


Fig. 18 An example of the trajectory and evidence of the steering action in the robot: **a** turns left, **b** turns right and **c** avoidance behavior when a mobile obstacle moves diagonally

5.4.2 Steering Action

This experiment consisted on steering the LEGO[®] robot when a mobile obstacle is presented in front of it. In particular, the robot steers in the opposite direction of the obstacle. The trajectory obtained from the experiment when the robot turns left and when it turns right are shown in Fig. 18a–b, and when it avoids a mobile obstacle

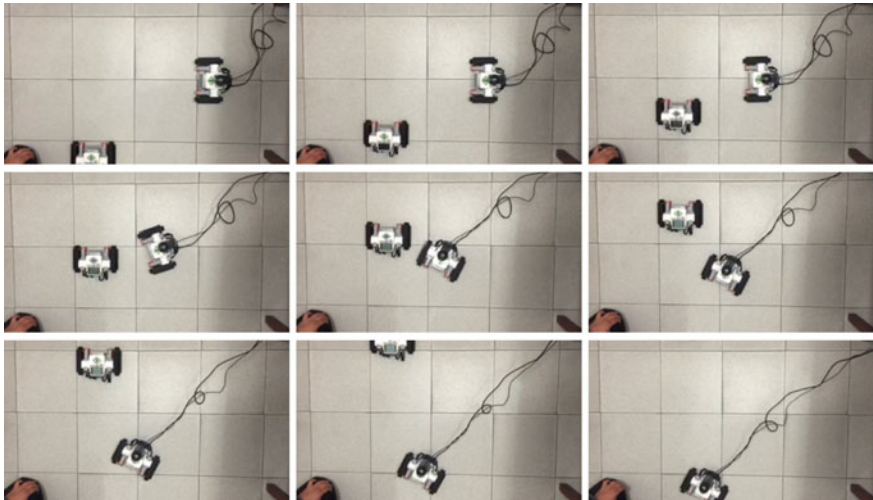


Fig. 19 Image sequence of the trajectory of robot for the left turn action



Fig. 20 Image sequence of the trajectory of robot for the right turn action

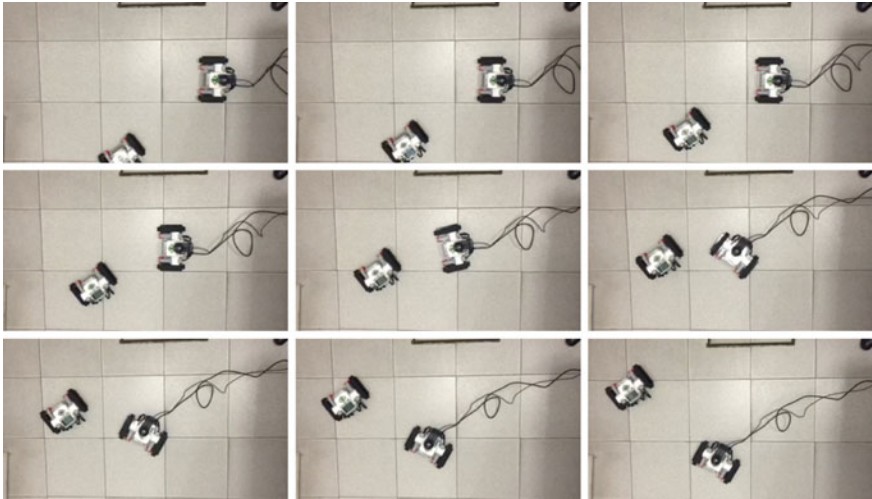


Fig. 21 Image sequence of the path followed for the robot when a mobile obstacle moves diagonally

that moves diagonal is shown in Fig. 18c. Notice that the LEGO[®] robot can avoid mobile obstacles in all situations.

The paths followed for the LEGO[®] robot are shown in the image sequences of Figs. 19 and 20 respectively, where the turn's choice of robot in both cases is opposite to direction of the obstacle in motion. In Fig. 21 we show the image sequence for the avoidance behavior of a mobile obstacle that moves diagonal.

6 Conclusions and Future Work

This work presented and described a new Hermite optical flow (RT-HOF) method for real-time purposes as a nature-inspired technique for computer vision. As described, this new real-time Hermite optical flow method is easier to implement and faster than the original Hermite optical flow method. In particular, this work focuses on using that approach for controlling a mobile robot.

As a case study, a LEGO robot with a webcam was occupied to design a controller for mobile obstacles avoidance. A detailed description of the design methodology of the controller was included, from an agent-environment based model to the robot's architecture. Different tests were run for characterizing and validating the controller. In fact, several comparisons between the proposed RT-HOF method with Horn and Schunck-based algorithm were developed in terms of background textures and velocities of mobile obstacles, in which RT-HOF method resulted to be very effective for real-time purposes. On the other hand, the design of the nature-inspired controller was described such that the robot can

perform actions based on its perception of the environment. Some guidelines were also presented to find empirical parameters in the controller.

Results confirm that the nature-inspired controller based on the new real-time Hermite optical flow method can avoid mobile obstacles in two different approaches: braking and steering when mobile obstacles are close to the robot. In addition, background textures and colours did not affect the performance of the controller. To this end, it is remarkable to say that this nature-inspired approach allows robots avoid mobile obstacles even though the parameters of the whole robotics system are unknown. In contrast with other works that perform obstacle avoidance with optical flow methods, the proposed RT-HOF: (i) is fast enough to compute an approximate solution of displacements between images that can be used as a visual perception in robotic systems, (ii) is robust to different background textures and colours, (iii) allows designing a nature-inspired controller, and (iv) its performance does not depend on parameters of a robotic system.

In future work, other improvements to the real-time Hermite optical flow method will be attended to increase the accuracy in the approximation of displacements and multi-resolution approaches will also be considered. Hardware independence is also considered for autonomous navigation of robots. In addition, other environmental characteristics will be evaluated, e.g. variation of light intensity. Also, the set of actions in the robot will be increased to perform better obstacle avoidance. Lastly, applications of this approach in the real-world will be investigated.

References

1. Abdagic, A., Tanovic, O., Aksamovic, A., Huseinbegovic, S.: Counting traffic using optical flow algorithm on video footage of a complex crossroad. In: *ELMAR, 2010 PROCEEDINGS*, pp. 41–45 (2010)
2. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. *Int. J. Comput. Vis.* **12**(1), 43–77 (1994). doi:[10.1007/BF01420984](https://doi.org/10.1007/BF01420984), <http://dx.doi.org/10.1007/BF01420984>
3. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. *ACM Comput. Surv.* **27**(3), 433–467 (1996)
4. Bloom, J., Reed, T.: A gaussian derivative-based transform. *IEEE Trans. Image Process.* **5**(3), 551–553 (1996)
5. Bruhn, A., Weickert, J., Schnörr, C.: Combining the advantages of local and global optic flow methods. In: *Proceedings of the Twenty-fourth DAGM Symposium on Pattern Recognition*, Springer, London, UK, pp. 454–462 (2002)
6. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vis.* **61**(3), 211–231 (2005)
7. Duhamel, P.E., Perez-Arancibia, N., Barrows, G., Wood, R.: Biologically inspired optical-flow sensing for altitude control of flapping-wing microrobots. *Mechatron. IEEE/ASME Trans.* **18**(2), 556–568 (2013). doi:[10.1109/TMECH.2012.2225635](https://doi.org/10.1109/TMECH.2012.2225635)
8. Galvin, B., Mccane, B., Novins, K., Mason, D., Mills, S.: Recovering motion fields: an evaluation of eight optical flow algorithms. In: *British Machine Vision Conference*, pp. 195–204 (1998)
9. Gibson, J.J.: The perception of the visual world. *Am. J. Psychol.* **64**, 622–625 (1951)

10. Henry Guennadi Levkin.: Imageprocessing/videocodexs/programming. <http://www.hlevkin.com/TestImages/additional.htm> (2004). Accessed 20 Jan 2013
11. Honegger, D., Greisen, P., Meier, L., Tanskanen, P., Pollefeys, M.: Real-time velocity estimation based on optical flow and disparity matching. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5177–5182 (2012). doi:[10.1109/IROS.2012.6385530](https://doi.org/10.1109/IROS.2012.6385530)
12. Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artif. Intell.* **17**(1–3), 185–203 (1981)
13. Koenderink, J., van Doorn, A.: Generic neighborhood operators. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(6), 597–605 (1992)
14. Lan, H., Mei, S.J., Hua, C.P., Hua, C.G.: Visual navigation for UAV using optical flow estimation. In: Control Conference (CCC), 2014 33rd Chinese, pp. 816–821 (2014). doi:[10.1109/ChiCC.2014.6896732](https://doi.org/10.1109/ChiCC.2014.6896732)
15. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. <http://people.csail.mit.edu/celiu/motionAnnotation/database/cameramotion.zip> (2008). Accessed 12 Oct 2011
16. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI '81), pp. 674–679 (1981)
17. Marr, D., Hildreth, E.: A theory of edge detection. *R. Soc. Lond. B* **207**, 187–217 (1980)
18. Martens, J.B.: The hermite transform-theory. *IEEE Trans. Acoust. Speech Signal Process.* **38** (9), 1595–1606 (1990)
19. Motokucho, T., Oda, N.: Vision-based human-following control using optical flow field for power assisted wheelchair. In: 2014 IEEE 13th International Workshop on Advanced Motion Control (AMC), pp. 266–271 (2014). doi:[10.1109/AMC.2014.6823293](https://doi.org/10.1109/AMC.2014.6823293)
20. Moya-Albor, E.: Optical flow estimation using the hermite transform. Ph.D. thesis, Universidad Nacional Autónoma de México, México City, México (2013)
21. Moya-Albor, E., Escalante-Ramírez, B., Vallejo, E.: Optical flow estimation in cardiac CT images using the steered Hermite transform. *Signal Process. Image Commun.* **28**(3), 267–291 (2013a). doi:[10.1016/j.image.2012.11.005](https://doi.org/10.1016/j.image.2012.11.005), <http://www.sciencedirect.com/science/article/pii/S092359651200207X>
22. Moya-Albor, E., Escalante-Ramírez, B., Vallejo, E.: Optical flow estimation of the heart's short axis view using a perceptual approach, vol. 8922, pp. 892,206–892,206–12 (2013b). doi:[10.1117/12.2041960](https://doi.org/10.1117/12.2041960), <http://dx.doi.org/10.1117/12.2041960>
23. Nagel, H.H.: Constraints for the estimation of displacement vector fields from image sequences. In: Proceedings of the Eighth International Joint Conference on Artificial Intelligence, vol. 2, pp. 945–951 (1983)
24. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (5), 565–593 (1986)
25. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vis.* **67**(2), 141–158 (2006)
26. Sakitt, B., Barlow, H.B.: A model for the economical encoding of the visual image in cerebral cortex. *Biol. Cybern.* **43**(2), 97–108 (1982)
27. Serra, P., Cunha, R., Hamel, T., Silvestre, C., Le Bras, F.: Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow. *IEEE Trans. Control Syst. Technol.* **23**(2), 570–583 (2015). doi:[10.1109/TCST.2014.2330996](https://doi.org/10.1109/TCST.2014.2330996)
28. Sharma, R., Taubel, G., Yang, J.S.: An optical flow and hough transform based approach to a lane departure warning system. In: 11th IEEE International Conference on Control Automation (ICCA), pp 688–693 (2014). doi:[10.1109/ICCA.2014.6871003](https://doi.org/10.1109/ICCA.2014.6871003)
29. Treisman, A.: Perceptual grouping and attention in visual search for features and for objects. *J. Exp. Psychol. Hum. Percept. Perform.* **8**(2), 194 (1982)
30. Valette, F., Ruffier, F., Viollet, S., Seidl, T.: Biomimetic optic flow sensing applied to a lunar landing scenario. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2253–2260 (2010). doi:[10.1109/ROBOT.2010.5509364](https://doi.org/10.1109/ROBOT.2010.5509364)

31. van Dijk, A.M., Martens, J.B.: Image representation and compression with steered Hermite transforms. *Sig. Process.* **56**(1), 1–16 (1997)
32. Wang, Z., Zhao, J.: Optical flow based plane detection for mobile robot navigation. In: 2011 9th World Congress on Intelligent Control and Automation (WCICA), pp. 1156–1160 (2011). doi:[10.1109/WCICA.2011.5970697](https://doi.org/10.1109/WCICA.2011.5970697)
33. Wilson, R., Granlund, G.H.: The uncertainty principle in image processing. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI.* **6**(6), 758–767 (1984)
34. Young, R.A.: The Gaussian derivative theory of spatial vision: analysis of cortical cell receptive field line-weighting profiles. Tech. Rep. GMR-4920, General Motors Research Laboratories, Detroit, Mich, USA (1985)
35. Young, R.A.: Simulation of human retinal function with the Gaussian derivative model. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 564–569 (1986)
36. Young, R.A.: The Gaussian derivative model for spatial vision: I. Retinal Mech. *Spat. Vis.* **2** (4), 273–293 (1987)
37. Zamudio, Z., Lozano, R., Torres, J., Campos, E.: Stabilization of a helicopter using optical flow. In: 2011 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), pp. 1–6 (2011). doi:[10.1109/ICEEE.2011.6106690](https://doi.org/10.1109/ICEEE.2011.6106690)