

Real-time velocity measurement to linear motion of a rigid object with monocular image sequence analyses

Danilo Filitto · Júlio Kiyoshi Hasegawa ·
Airtón Marco Polidório · Nardênio Almeida Martins ·
Franklin César Flores

Received: 2 July 2014 / Accepted: 3 November 2014 / Published online: 28 November 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract This paper presents a methodology and all procedures used to validate it, which were executed in a physics laboratory under controlled and known conditions. The validation was based on the analyses of registered data in an image sequence and the measurements acquired by high precision sensors. This methodology intended to measure the velocity of a rigid object in linear motion with the use of an image sequence acquired by commercial digital video camera. The proposed methodology does not need a stereo pair of images to calculate the object position in the 3D space: it needs only images sequence acquired for one, only one, angle view (monocular vision). To do so, these objects need to be detected while in movement, which is conducted by the application of a segmentation technique based on the temporal average values of each pixel registered in N consecutive image frames. After detecting and framing these objects, specific points belonging to the object (pixels), on the plane image (2D

coordinates or space image), are automatically chosen, which are then transformed into corresponding points in the space object (3D coordinates) by the application of collinearity equations or rational functions (proposed in this work). After obtaining the coordinates of these points in the space object that are registered in the sequence of images, the distance, in meters, covered by the object in a particular time interval may be measured and, consequently, its velocity can be calculated. The system is low cost, use only a computer (architecture Intel I3), and a webcam used to acquire the images (640×480 , 30 fps). The complexity of the algorithm is linear, fact that allows the system to operate in real time. The results of the analyses are discussed and the advantages and disadvantages of the method are presented.

Keywords Moving objects · Image segmentation · Geometric transformation · Velocity measurement · Rational polynomials · Collinearity equations · Monocular image sequence

D. Filitto · A. M. Polidório · N. A. Martins (✉) · F. C. Flores
Department of Informatics, UEM, Maringá, PR, Brazil
e-mail: nardenio@din.uem.br

D. Filitto
e-mail: dfilitto@gmail.com

A. M. Polidório
e-mail: ampolidorio@gmail.com

F. C. Flores
e-mail: fcflores@din.uem.br

J. K. Hasegawa
Cartography Department, UNESP, Presidente Prudente,
SP, Brazil
e-mail: hasegawa@fct.unesp.br

1 Introduction

Object movement in any environment is a common research field in Digital Image Processing (DIP). Among the several types of systems capable of monitoring movement in any specific environment, there are those that detect and count moving vehicles [2, 12, 21]; and those that measure the velocity of vehicles [27, 28].

In [2], the monitoring system of traffic flow and road traffic analysis is developed. This system uses methods of image processing and pattern recognition to measure the velocity and recognize the license plate number of the vehicles.

In [12], the system of detection and classification of vehicles is proposed, which uses the size of vehicles identified in segmentation process to calculate the height and the actual length of the same, allowing to classify them as vehicles and not vehicles (Vans, Pick Ups, Trucks).

In [21], a vehicle counting system is proposed, which calculates the total number of vehicles that are traveling on the highway through a tracking zone, in which is performed the extraction of the highway structure, the detection of the moving vehicles and finally the count of vehicles identified in the detection process.

Already the system developed by [27] uses the camera calibration method of Tsai in two phases [23] to define the intrinsic and extrinsic parameters of the camera and a mathematical model to define the geometric relationship between plane image and space object, aiming to convert the coordinates of the plane image (2D), in which the vehicle is, into the coordinates in the world (3D). Finally velocity in linear motion is measured on basis of the difference of time between two sequential frames.

In [28], the system developed is aimed to calculate the vehicle velocity. The main idea of this system is the creation of several control lines in the image. The control lines introduced on the image are used to define the traffic ranges and create points of correspondence with the real world. Based on these points is calculated actual position of each pixel to control the vehicle in the world. With the coordinates in hand, the vehicle velocity is calculated through the covered distance between two points in a given time interval.

To develop a monitoring system based exclusively on the use of DIP techniques, several problems need to be overcome. First of all, such a system must be able to detect and segment the element being monitored in the scene. The segmentation method must be robust, as it must be able to circumvent several adverse conditions such as those inherent to the element of interest itself (color, size, geometric shape, differences in texture patterns, and its own movement in the scene), as well as those problems relative to the environment (variations in the intensity of solar illumination, rain, shadows, interferences caused by other objects present in the scene, etc.). In other words, a monitoring system based only on images has to overcome not only the problems caused by the environment, but also those caused by the object of interest itself.

These problems are difficult to be treated. Robust methods that detect movement of an object in an image sequence are constantly improved or proposed, as: based on optical flow [4, 6–8, 18]; motion history image [9, 10]; background segmentation by codebook model [15].

An additional problem exists when elements belonging to the imaged scene need to be reconstructed tridimensionally using, only, the data from the image. One possible

way to perform this reconstruction is through the computing of stereo pairs images and reconstruct the space object by the application of photogrammetric methods [1, 17].

To proceed this reconstruction, it is necessary, at first, to carry out internal and external orientation of the camera and after performing the rectification and registration of such images acquired in stereo vision, then you should search for homologous pixels between images of the stereo pair for reconstructing these points in the space object [1, 6, 24].

As can be seen, only performing a study on the 3D reconstruction based on images analyses is a problem of significant value. The insertion of other problems can introduce severe errors in this study. How to consider, for example, the motion of a body in an uncontrolled environment introduces difficulties and uncertainties in the 3D reconstruction process.

Thus, this work has only the aim to propose a method for 3D reconstruction of interest points observed in a sequence of images acquired under known conditions, and use those points to measure the velocity of moving objects.

The innovation introduced by this work is to consider only images acquired by a monocular vision system, and this way, to eliminate the photogrammetric procedures (internal and external camera orientation, images rectification and registration and the search for homologous pixels).

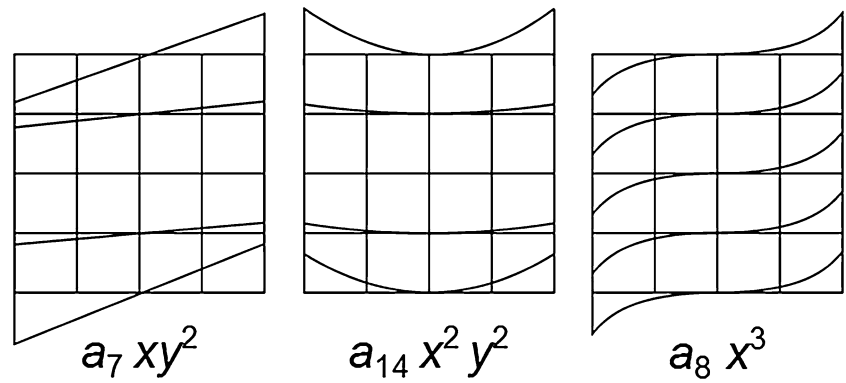
So, a new method to solve this problem is proposed and to validate this method an experiment was executed in a physics laboratory under controlled and known conditions. This study involves the reconstruction of points belonging to the plane image (2D) into the corresponding points in the space object (3D) using monocular image sequences acquired by a commercially available, low-cost, digital video camera [3, 13].

To measure the velocity of a rigid object in linear motion, the covered distances have to be computed. When it is performed using images or points of interest on the image, the image must be geometrically corrected. This geometric correction is required due to the sphericity of the camera lens; all acquired images must be managed by the central perspective projection (as can be seen in Fig. 1), resulting in all points in the object image (3D) within an image being registered on the plane image (2D) leading to losses of data and geometric distortions.

The goal of this study was to present a method enable to determine the velocity of a rigid object in linear motion in real time with the use of monocular image sequence analysis.

This study presents the application of a technique to image sequence segmentation that considers the history of the variation of values associated with the homologue pixels registered in successive image frames [15], and the analysis of geometric relations between points in real space

Fig. 1 Distortion patterns modeled by specific polynomial terms [20]



(space object) and the corresponding points on the plane image, obtained by collinearity equations [5, 26] and rational functions [19, 20, 22].

This paper has been organized as follows: Sect. 2 presents the method applied during the segmentation phase. Section 3 describes how the conversion of plane image coordinates into space object coordinates was conducted. Section 4 describes the proposed system, as well as the experimental results. Finally, in Sect. 5, conclusions are presented and the final considerations are made.

2 Image segmentation based on the history of the values associated with pixels

Kim et al. [15] developed a technique for image segmentation that permits capturing background variations and dealing with scenes that contain moving objects or differences in lighting. This technique quantizes samples of each pixel of an image in codebooks, which represents, in a compact manner, the background model of image sequences in a particular period of time.

To apply this technique, it is necessary to acquire, a priori, a training value sequence \mathfrak{X} for each pixel from each image frame. For N image frames, N vectors x , belonging to the color space RGB, $\mathfrak{X} = \{x_1, x_2, \dots, x_N\}$, are necessary to enable the historical register of the color value variation (or any other attribute) occurred in a determined period of time associated with each pixel. Each pixel has a codebook $\mathfrak{C} = \{c_1, c_2, \dots, c_L\}$ composed by L codewords. Each codeword c_i , $i = 1..L$, is composed by an RGB vector $\mathbf{V}_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a tuple $aux_i = \{\tilde{I} min_i, \hat{I} max_i, f_i, \lambda_i, p_i, q_i\}$ that contains brightness intensity and the temporal variables described as follow:

- $\tilde{I} min_i, \hat{I} max_i$: represent, respectively, the smallest and the largest brightness value observed among all brightness values associated with the variation occurred in the historical register of the pixel i ;

- f_i : represents the frequency a codeword occurs;
- λ_i : represents the longest time interval (during the training period) that one codeword was not recovered;
- p_i, q_i : represent, respectively, the first and the last access time that occurred in the codeword.

2.1 Reference model training

To cope with changing global and local lighting, Kim et al. [15] developed a model for dealing with color distortion and brightness distortion.

Consider the input pixel $X_t = (R, G, B)$ and a codeword c_i , where $\mathbf{V}_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$, and

$$\|X_t\|^2 = R^2 + G^2 + B^2 \tag{1}$$

$$\|\mathbf{V}_i\|^2 = (\bar{R}_i, \bar{G}_i, \bar{B}_i) \tag{2}$$

$$\langle X_t, \mathbf{V}_i \rangle^2 = \mathbf{V}_i = (\bar{R}_i R_i + \bar{G}_i G_i + \bar{B}_i B_i) \tag{3}$$

The color distortion can be calculated by:

$$p^2 = \|X_t\|^2 \cos^2 \theta = \frac{\langle X_t, \mathbf{V}_i \rangle^2}{\|\mathbf{V}_i\|^2} \tag{4}$$

This calculation is represented by the function *colordist*, i.e.,

$$colordist(X_t, \mathbf{V}_i) = \delta = \sqrt{\left(\|X_t\|^2 - p^2\right)} \tag{5}$$

The color distortion can be interpreted as a brightness-weighted version in the normalized color space. This is equivalent to normalizing a codeword vector to the brightness of an input pixel. This way, the brightness is taken into consideration for measuring the color distortion, and to avoid the instability of normalized colors.

Consider also the *brightness* function, given by:

$$brightness(I, \tilde{I} min_m, \hat{I} max_m) = \begin{cases} true, & \text{if } I_{low} \leq X_t \leq I_{hi} \\ false, & \text{otherwise} \end{cases} \tag{6}$$

Table 1 Codebook construction algorithm [15]

Codebook construction algorithm	
I.	$L \leftarrow 0$,
II.	$\mathfrak{C} \leftarrow \emptyset$ (create empty sets)
III.	for $t = 1$ to N do
i.	$X_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$
ii.	/Seek codeword c_m in $\mathfrak{C} = \{c_i 1 \leq i \leq L\}$ that combines with X_t based on two conditions (a) and (b)
a)	$colordist(X_t, \mathbf{V}_m) \leq \varepsilon_1$
b)	$Brightness(I, \langle \hat{I}min_m, \hat{I}max_m \rangle) = true$
iii.	If $\mathfrak{C} = \emptyset$, or if the codeword is not found, then $L \leftarrow L + 1$. Create codeword c_L , with the following attribution:
•	$\mathbf{V}_L \leftarrow (R, G, B)$
•	$aux_L \leftarrow \langle I, I, 1, t - 1, t, t \rangle$
iv.	Otherwise, update codeword c_m , consisting of
•	$\mathbf{V}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $aux_m = \langle \hat{I}min_m, \hat{I}max_m, f_m, \lambda_m, p_m, q_m \rangle$
•	$\mathbf{V}_m \leftarrow \left(\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m}{f_m + 1}, \frac{f_m \bar{B}_m}{f_m + 1} \right)$
•	$aux_m \leftarrow \langle \min\{I, \hat{I}min_m\}, \max\{\hat{I}max_m, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$
	end for
IV.	For each codeword c_i , $i = 1, \dots, L$, alter the value of λ_i to $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$.

In the training period, each value X_t sampled at the time t is compared to values on the codebook to determine if a codeword c_m that corresponds to the sampled value exists. To determine if a codeword exists, an average value for the distortion on the limits of the color and brightness in the codeword of index m is applied.

In the algorithm presented in Table 1, both conditions (a) and (b) present in step III—(ii) are satisfied when the color of X_t and c_m are similar, and the brightness of X_t is among the acceptable limits of the brightness of c_m .

Condition (a) verifies the distortion (δ) of the values associated with an entry pixel $X_t = (R, G, B)$ relative to a codeword c_i in which $\mathbf{V}_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$. This condition compares the result of the *colordist* function with a threshold value ε_1 .

Condition (b) verifies in the tuple codeword c_m , if the brightness value I of X_t belongs to the interval defined by the largest (I_{hi}) and by the smallest (I_{low}) brightness values of each codeword c_m . In other words, $I \in [I_{low}, I_{hi}]$, as:

$$I_{low} = \min \left\{ \beta \hat{I}max, \frac{Imin}{\alpha} \right\} \quad (7)$$

$$I_{hi} = \alpha \hat{I}max \quad (8)$$

for $\alpha < 1$ and $\beta > 1$. These values are working to reduce the range of the interval $[I_{low}, I_{hi}]$.

According to Kim et al. [15], α is between 0.4 and 0.7 and β is between 1.1 and 1.5. This range $[I_{low}, I_{hi}]$ becomes a stable range during codebook updating.

The α value is obtained through experiments. The value 0.4 allows large brightness bounds, but 0.7 gives tight bounds. The β is additionally used for limiting I_{hi} since shadows (rather than highlights) are observed in most cases. In the case of the experiment, the values $\alpha = 0.5$ and $\beta = 1.2$ were set empirically.

The training for background modeling generates a codebook for each *pixel*, formed by codewords that represent the history of values associated with each image *pixel* in a determined training period. However, many of these values belonging to a codebook of each pixel represent invalid entries, such as noises or moving objects. Such entries are eliminated through the creation of a reference model.

2.2 Reference model

The reference model of an imaged scene (Eq. 9) is generated based on the data filtered from the codebook for all the *pixels*. This filtering process removes from the codebook all the codewords that represent noises or moving objects, keeping only the entries that represent the image background.

$$m = \{c_m | c_m \in \mathfrak{C} \text{ and } \lambda_m \leq Tm\} \quad (9)$$

The term Tm in Eq. 9 represents a threshold value used to remove codewords that are supposedly associated with noises and moving objects. According to [15], it is advisable that the most adequate value for this threshold is the amount of frames used in the reference model training

Table 2 Algorithm for background subtraction (BGS) [15]

Algorithm for background subtraction	
I.	$X = (R, G, B), I \leftarrow \sqrt{R^2 + G^2 + B^2}$
II.	For all the codewords contained in m (Equation 9), find the codeword c_m that corresponds to the value X , based on the following conditions: <ul style="list-style-type: none"> • $colordist(X, c_m) \leq \varepsilon_2$ (threshold value) • $Brightness(I, \langle \hat{I}min_m, \hat{I}max_m \rangle) = true$ Update the codeword found, as described in the codebook construction algorithm, step II item iv (Table 1).
III.	$BGS(x) = \begin{cases} \textbf{Object:} & \text{in case corresponding codewords are not found} \\ \textbf{Background:} & \text{in case corresponding codewords are found} \end{cases}$

process divided by 2. In outdoor environments it is advisable [27], to use a reference model training period of over 5 min.

2.3 Background detection

Once the reference model is obtained, it is possible to subtract from a current image the *pixels* referring to static objects (image background). The algorithm presented in Table 2 verifies if any *pixel* in the image belongs to the background or to the moving objects.

3 Coordinate conversion

This section describes the techniques used to conduct coordinate conversions from plane image (2D) to space object (3D).

Two models will be presented for this conversion, the polynomial one, suitable for images when you do not know the parameters of interior orientation of the acquisition sensor and collinearity equations, based on geometrical relationship of the perspective projection and knowledge of the parameters of interior orientation camera.

3.1 Polynomial models

It is possible to relate the plane image and the space object through Polynomial models [22]. The short form of a simple polynomial (non-rational) may be given by the Eqs. 10 and 11:

$$l = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^p a_{ijk} x^i y^j z^k \tag{10}$$

$$c = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^p b_{ijk} x^i y^j z^k \tag{11}$$

where (l, c) represent the coordinate of a determined pixel in the matrix image (row and column, respectively); x, y and z are the tridimensional Cartesian coordinates of the point on the ground; a_{ijk} , and b_{ijk} are the polynomial coefficients; and m, n, p are integer values, that belong to the interval $[0, 3]$, with $m + n(+p)$ being the order of the polynomial functions, generally three. Equations 12 and 13 show in more detail an example of an expansion of the notations referring to Eqs. 10 and 11; in this case, a polynomial is of the third order in the 3D space.

$$l = a_1 + a_2y + a_3x + a_4z + a_5yx + a_6yz + a_7xz + a_8y^2 + a_9x^2 + a_{10}z^2 + a_{11}xyz + a_{12}y^3 + a_{13}yx^2 + a_{14}yz^2 + a_{15}y^2x + a_{16}x^3 + a_{17}xz^2 + a_{18}y^2z + a_{19}x^2z + a_{20}z^3 \tag{12}$$

$$c = b_1 + b_2y + b_3x + b_4z + b_5yx + b_6yz + b_7xz + b_8y^2 + b_9x^2 + b_{10}z^2 + b_{11}xyz + b_{12}y^3 + b_{13}yx^2 + b_{14}yz^2 + b_{15}y^2x + b_{16}x^3 + b_{17}xz^2 + b_{18}y^2z + b_{19}x^2z + b_{20}z^3 \tag{13}$$

According to [20], there are distortions or movement patterns that may be modeled or corrected by specific polynomial terms (Fig. 1). However, the main advantage of using the polynomial model is the correction of all sources of distortions simultaneously [19].

The parameters of the Eqs. 12 and 13 are determined knowing of at least 21 points (control points) in three-dimensional cartesian coordinates (space object) and his record in the image.

The same Eqs. (12 and 13), can be used to determine the three-dimensional coordinates of points, setting the Z coordinate and calculating the X and Y coordinates.

3.2 Collinearity equations

Collinearity equations permit to relate plane image to space object transforming the space object coordinates into plane

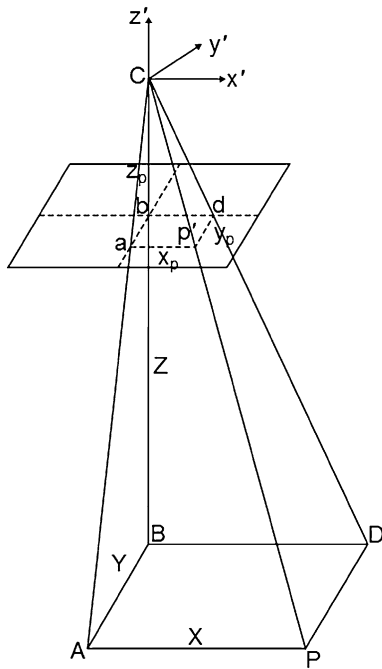


Fig. 2 Coordinate system, parallel image and object, originating in the perspective center

image coordinates (and vice versa). In photogrammetric process, collinearity equations [16] reproduce mathematically the process of image formation, linking the coordinates in the space object (3D) to their corresponding coordinates in the plane image (2D).

The basic principle to establish collinearity is based on the condition that points C (perspective center), p' (image point) and P (object point) belong to the same line [5, 26].

The geometry shown in Fig. 2 presents the following similar triangles $\Delta CDB \sim \Delta Cdb$ and $\Delta CBA \sim \Delta Cba$ and, therefore, permits to establish the following geometric relationships between the measurements:

$$\frac{x_p}{X} = \frac{z_p}{Z} \quad \text{and} \quad \frac{y_p}{Y} = \frac{z_p}{Z} \tag{14}$$

which permits to determine the projective equation systems:

$$x_p = z_p \frac{X}{Z} \quad \text{and} \quad y_p = z_p \frac{Y}{Z} \tag{15}$$

where (x_p, y_p, z_p) is the coordinates of point p' in the plane image, corrected for systematic errors (carried out interior orientation);

(X, Y, Z) is the coordinates of point P in the space object.

In case of the coordinates in the space object are translated, rotated or with a different scale from the plane

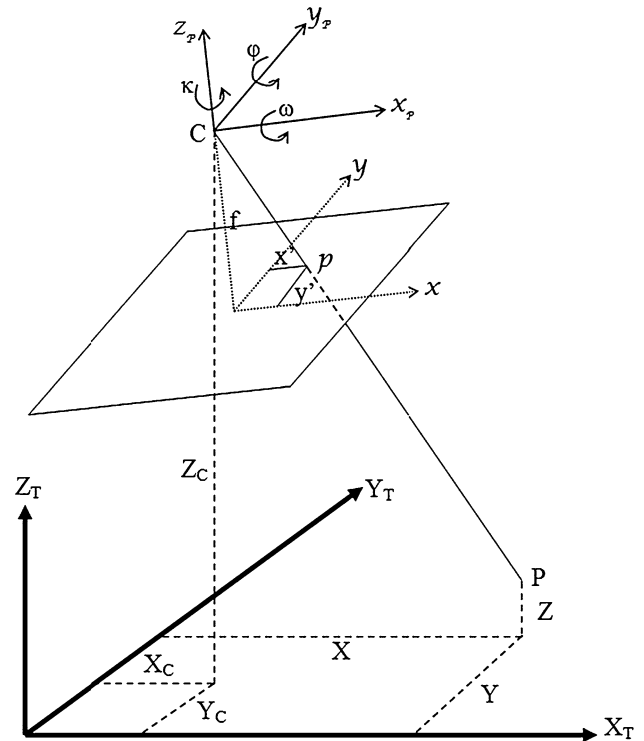


Fig. 3 Coordinate systems of the image (x_p, y_p, z_p) and the translated and rotated object (X_T, Y_T, Z_T) [16]

image (Fig. 3), similarity transformation must be applied on the referential system coordinates of the space object (point P) to the referential system of the plane image (point p'). This transformation may be conducted in three steps: (a) applying three translations in order to compensate the spatial difference between the origins; (b) applying rotation to compensate angular differences; and (c) execute the correction of the difference in scale between the referential systems.

The rotation movements occur counterclockwise, considering that the referential system of the image gyrates while the object remains fixed, whose rotation matrices M_κ , M_σ , and M_ω are given by:

$$\begin{aligned} M_\kappa &= \begin{bmatrix} \cos \kappa & \sin \kappa & 0 \\ -\sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}, M_\sigma \\ &= \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix}, M_\omega \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & \sin \omega \\ 0 & -\sin \omega & \cos \omega \end{bmatrix} \end{aligned} \tag{16}$$

The matrices M_κ , M_σ e M_ω are rotation matrices around the axes x , y and z , respectively.

$$M = \begin{bmatrix} \cos \varphi \cos \kappa & \cos \omega \sin \kappa - \sin \omega \sin \varphi \cos \kappa & \sin \omega \sin \kappa - \cos \omega \sin \varphi \cos \kappa \\ -\cos \varphi \sin \kappa & \cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa & \sin \omega \cos \kappa - \cos \omega \sin \varphi \sin \kappa \\ \sin \varphi & -\sin \omega \cos \varphi & \cos \omega \cos \varphi \end{bmatrix} \tag{17}$$

The rotation matrix M ($M = (M_\kappa \cdot M_\sigma \cdot M_\omega)$) is given by: and the transformation will happen through:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda M \begin{bmatrix} X_P - X_c \\ Y_P - Y_c \\ Z_P - Z_c \end{bmatrix} \tag{18}$$

in which

- λ is a scale factor;
- X_c, Y_c and Z_c are the coordinates of the perspective center in the object reference system;
- X, Y and Z are the point coordinates in the system XYZ .

The equation system described in Eq. 18 may be written as:

$$\begin{aligned} X &= \lambda[m_{11}(X_P - X_c) + m_{12}(Y_P - Y_c) + m_{13}(Z_P - Z_c)] \\ Y &= \lambda[m_{21}(X_P - X_c) + m_{22}(Y_P - Y_c) + m_{23}(Z_P - Z_c)] \\ Z &= \lambda[m_{31}(X_P - X_c) + m_{32}(Y_P - Y_c) + m_{33}(Z_P - Z_c)] \end{aligned} \tag{19}$$

By replacing Eq. 19 in the projective Eq. 15, λ is eliminated from the equation system, resulting in the collinearity equations, Eq. 20:

$$\begin{aligned} x_p &= z_p \cdot \frac{m_{11}(X - X_c) + m_{12}(Y - Y_c) + m_{13}(Z - Z_c)}{m_{31}(X - X_c) + m_{32}(Y - Y_c) + m_{33}(Z - Z_c)} \\ y_p &= z_p \cdot \frac{m_{21}(X - X_c) + m_{22}(Y - Y_c) + m_{23}(Z - Z_c)}{m_{31}(X - X_c) + m_{32}(Y - Y_c) + m_{33}(Z - Z_c)} \end{aligned} \tag{20}$$

The equation system, Eq. 20, is used in the negative film image conception [19]; when applied on the reversal film, due to the fact that $z_p = -f$ (focal distance), it becomes:

$$\begin{aligned} x_p &= -f \cdot \frac{m_{11}(X_P - X_c) + m_{12}(Y_P - Y_c) + m_{13}(Z_P - Z_c)}{m_{31}(X_P - X_c) + m_{32}(Y_P - Y_c) + m_{33}(Z_P - Z_c)} \\ y_p &= -f \cdot \frac{m_{21}(X_P - X_c) + m_{22}(Y_P - Y_c) + m_{23}(Z_P - Z_c)}{m_{31}(X_P - X_c) + m_{32}(Y_P - Y_c) + m_{33}(Z_P - Z_c)} \end{aligned} \tag{21}$$

The Eq. 21 can be used to determine the exterior orientation parameters of the camera. Thus, it is necessary that at least four points with known coordinates in the three-dimensional Cartesian coordinate system to be imaged.

The inverse form of the collinearity equations may be obtained by applying the inverse transformed in the equation system, Eq. 18, which results in:

$$\begin{bmatrix} X_P - X_c \\ Y_P - Y_c \\ Z_P - Z_c \end{bmatrix} = \lambda^{-1} M^T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{22}$$

and

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda_p \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \tag{23}$$

By replacing Eq. 23 in the Eq. 22, resulting in:

$$\begin{bmatrix} X_P - X_c \\ Y_P - Y_c \\ Z_P - Z_c \end{bmatrix} = \lambda^{-1} M^T \lambda_p \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \tag{24}$$

The Eq. 24 may be written as:

$$\begin{aligned} X_P - X_c &= \lambda^{-1} \lambda_p (m_{11}x_p + m_{21}y_p + m_{31}z_p) \\ Y_P - Y_c &= \lambda^{-1} \lambda_p (m_{12}x_p + m_{22}y_p + m_{32}z_p) \\ Z_P - Z_c &= \lambda^{-1} \lambda_p (m_{13}x_p + m_{23}y_p + m_{33}z_p) \end{aligned} \tag{25}$$

By isolating the terms X_P and Y_P from Eq. 25, the inverse form of the collinearity equations is obtained (Eq. 26).

$$\begin{aligned} X_P &= X_c + (Z_P - Z_c) \frac{m_{11}x_p + m_{21}y_p + m_{31}z_p}{m_{13}x_p + m_{23}y_p + m_{33}z_p} \\ Y_P &= Y_c + (Z_P - Z_c) \frac{m_{12}x_p + m_{22}y_p + m_{32}z_p}{m_{13}x_p + m_{23}y_p + m_{33}z_p} \end{aligned} \tag{26}$$

4 The system

A computational system capable of measuring the velocity of a rigid object in linear motion was developed. This system is composed by five modules:

1. interface module, responsible for performing the interface between the computational system and the video camera, capturing the images acquired by the camera and making them available for processing;
2. the module responsible for conducting the detection of moving objects in a sequence of acquired images, using the segmentation module based on the Bradski's Mean Shift algorithm;
3. the coordinate conversion module, responsible for performing the coordinate conversion from plane image to space object by using the collinearity equations or the polynomial model. This module has as the entry parameter the knowledge a priori of 21

Fig. 4 Flowchart of the system

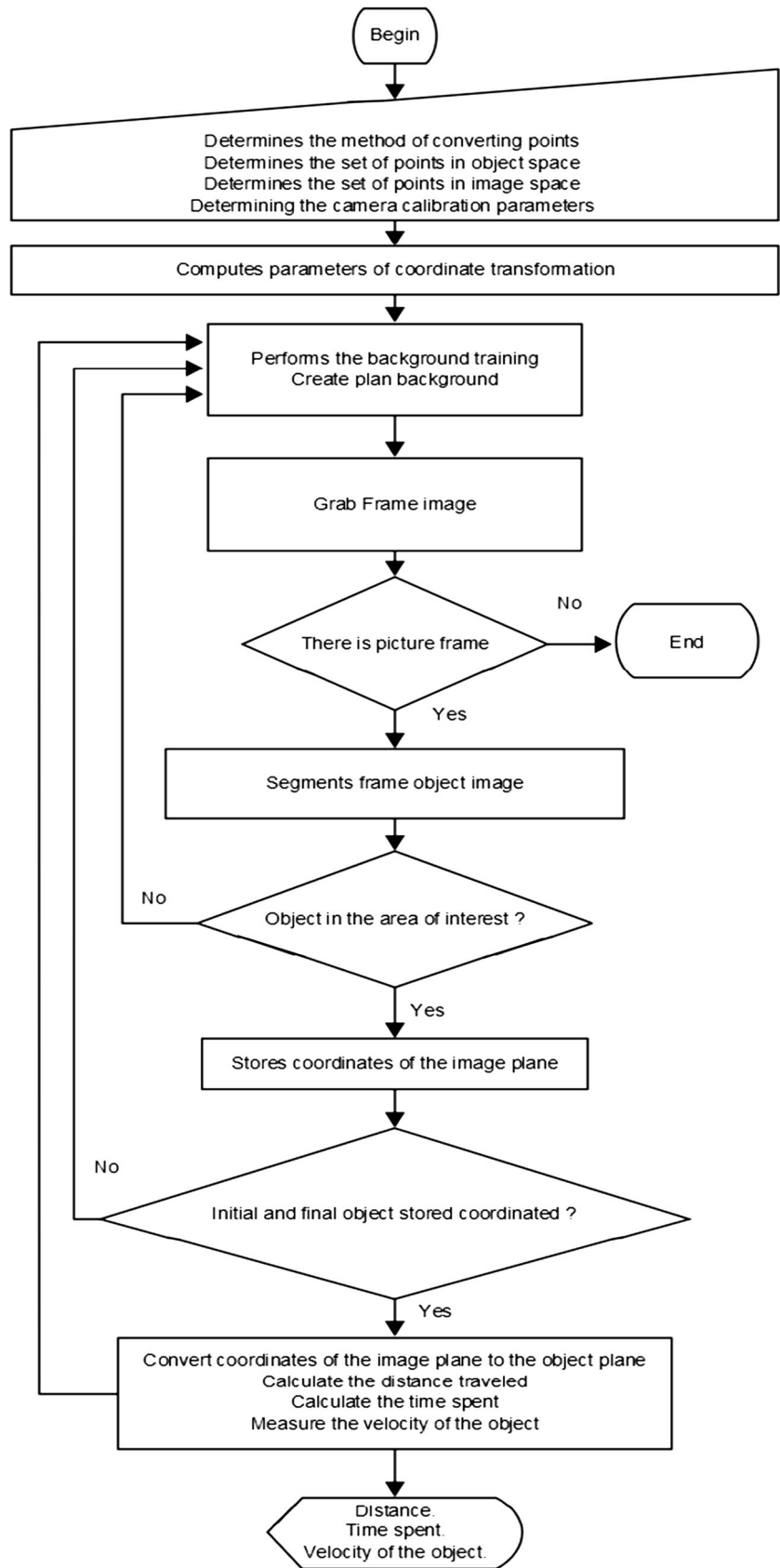
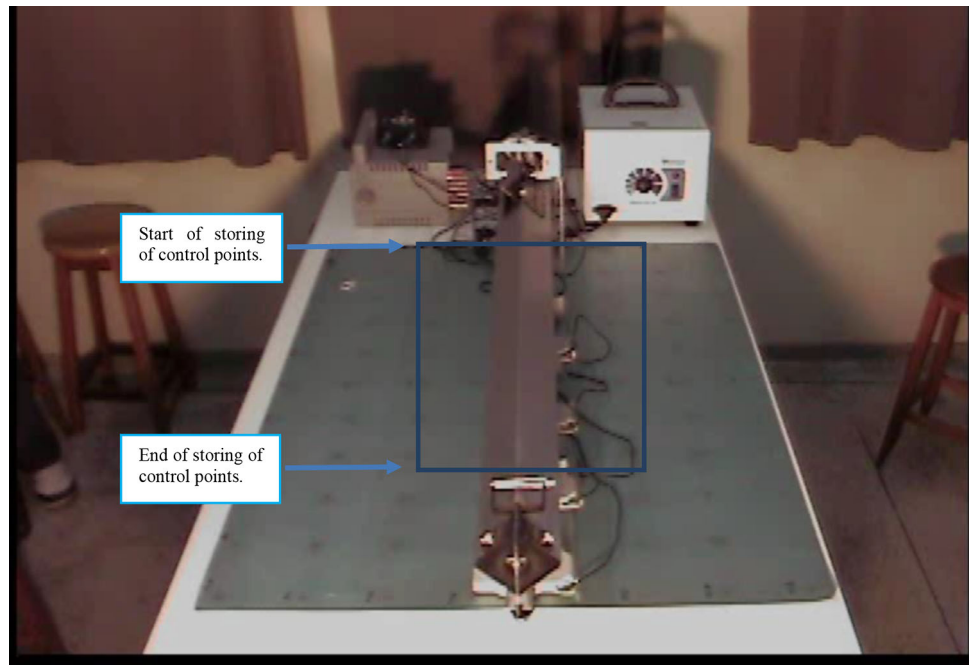


Fig. 5 Bounding box implemented in the detection module for the storing of control points



points of the space object with their respective points on the plane image. In case the conversion method chosen is the collinearity equations, the information about camera calibration is also inserted;

4. the module responsible for measuring the velocity of an object and conducting the analyses relative to this value;
5. the module responsible for exhibiting the sequences of images with the superposition of the computed details.

The implementation of the system (i.e., system modules) is represented in the flowchart in Fig. 4, which is explained in the sequence.

All modules were implemented in C⁺⁺. The OpenCV library was used to handle and to process the captured images.

The implementation starts with the initialization of the interface module, where the source of images is set (webcam or video file). In the following, it is possible to pick an image frame from the video source and to store it in the RAM. This image is represented by the structure Ipl-Image, provided by the OpenCV library.

The image orientation process is done in a unique image, because the camera stands in a still position during the acquisition of the images. After the observation of the controlling points from the image (in the reference plate, to see Fig. 6), the coordinate conversion module is activated, receiving as a parameter, the mathematical model (Eqs. 12 and 13 to the polynomial method or Eq. 26 to the collinearity equations). The module also receives, as parameters, the intrinsic and extrinsic camera data, with

the control points from the image space and their respective coordinates in the object space (21 control points). In this module, it is applied a method to determine the orientation parameters that will be used to convert the points from the space image to the space object.

Following the initialization of the coordinate conversion module, the object detection module is started, and the background image is computed from captured environment images for, at least, 40 s. The background image will be used to segment the moving bodies.

In the following, the monitoring task is started, by activation of the velocity measurement module. Another module is responsible to exhibit the results on screen.

The detection module captures a frame from the image sequence and segments it to verify if there exists an object inside a region of interest given by a bounding box (Fig. 5). If the object is found in the bounding box, the object control point is determined (lower left coordinate) and sent to the measurement module. The segmentation module recalculates the background image taking into account the last captured frame. After this recalculation, the process iterates and another frame is captured from the image sequence.

If the detected object in the segmentation step is located after the region of interest, the measurement module converts the control points by the application of the coordinates conversion module and the velocity is measured as well. The exhibition module takes the computed data and displays it on screen.

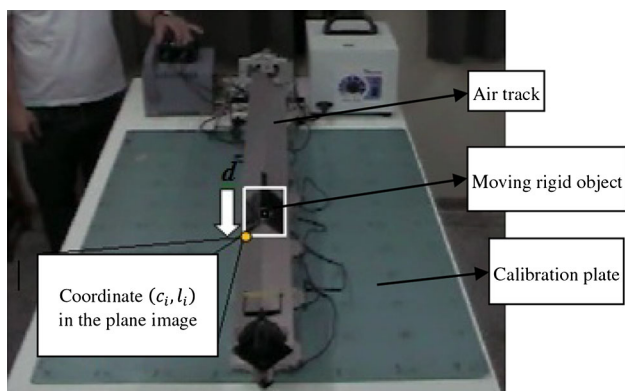


Fig. 6 Image sequence, frame i . The movement of a rigid object is detected (vector d points to the lower left-hand corner of the image). The object is surrounded by a minimal rectangle (in white). The coordinate (c_i, l_i) of the point (in yellow) on the left-hand side of the lower corner of the rectangle is extracted

4.1 The experiment

To validate the system and the methodology developed, an experiment was performed in an Applied Physics Laboratory, with the use of a device named “Air Track” (Fig. 6) to measure the velocity of a rigid object in linear motion on a surface. The Air Track possesses small orifices along its sides. When compressed air is injected inside the track, air leaves these small orifices and forms an air mattress on the surface of the track. This system permits that an object slides over the track with no (or insignificant) friction between the surfaces.

Five photoelectric sensors are placed along the track, each sensor is connected to a chronometer (precision of 5×10^{-4} s). The distances (Δs) that separate the sensors one from the other are known. When an object slides over the track, passing by the first sensor, all chronometers are activated from zero second display. When the object passes by the following sensors, the chronometer connected to that sensor is stopped. Thus, the distance traveled by the object between each sensor and the time (Δt) needed to cover each of the distances are obtained. With these data, the velocity value, $v = \Delta s / \Delta t$, is calculated at each of the measured points.

The experiment was totally registered in a sequence of images, with the use of a low-cost digital camera. Each image frame registered was submitted to the movement detection method and the segmentation of moving objects developed by [15].

After segmenting the moving object belonging to i -th frame registered at time t_i , the coordinate (c_i, l_i) relative to the plane image was extracted (Fig. 6), but significant in relation to the movement performed by the object. After obtaining all coordinates of two consecutive frames (c_i, l_i) and (c_{i+1}, l_{i+1}) , acquired at the moments t_i and t_{i+1} , respectively, they are analyzed to detect if the object

altered its position. In case some alteration in the object position takes place, a rectangle is drawn surrounding the object and the coordinates of the lower left-hand corner are used in the calculations (Fig. 6). To calculate the velocity of the object, two parameters need to be determined: (1) the time interval in which the movement is observed (Δt); and (2) the covered distance (Δs), in metric units, by the moving object during this observation.

The time interval of the observation is obtained by the amount of frames that the camera is capable of imaging per second. For each acquired frame i , the time t_i in which that happened is also registered. Thus, for two consecutive image frames:

$$\Delta t = t_{i+1} - t_i \quad (27)$$

The determination of the covered distance (Δs) must be performed in space object (3D—real world). The point measured in the plane image (2D) needs to be transformed (Eqs. 12 and 13 – for polynomial models or Eq. 22 – for collinearity equations), so that velocity may be effectively calculated.

In order that this transformation may be used in the determination of the Cartesian 3D coordinates (space object), the parameters a_i and b_i ($i = 1, \dots, 20$) must be determined (Eqs. 12 and 13). These 40 parameters are determined using at least 21 photo-identifiable points (support 3D coordinates known in the space object), referencing the image to the space object coordinates. Similarly, in the case of collinearity equations (Eq. 21), the minimum requirement is four control points. Thus, the covered distance may be determined calculating the rectangle surrounding the object in the two moments.

For the performance of the experiment in the laboratory, a steel plate containing 88 visible points, distributed in a matrix arrangement, regularly spaced in 100 mm was used to establish a 3D Cartesian coordinate system, attributing coordinates X, Y and Z to the points on the plate (Fig. 7). When images were acquired, some of the points on the plate were chosen as control points, to be measured in the plane image, enabling the determination of the transformation parameters of Eq. 22 [14].

When the transformation parameters and the calibration information of the camera are known, any point in the image may have its coordinate in the space object estimated quite precisely. By applying the equation for the Euclidian distance in the 3D coordinates in the space object at two moments, the covered distance (Δs) is determined, which is used to determine the object velocity.

4.2 Procedures for conducting the experiment

Figure 8 illustrates the environment setup for the execution of the experiment. Under the air track a steel plate was

Fig. 7 Air track system used to calculate the velocity of a rigid object in controlled conditions

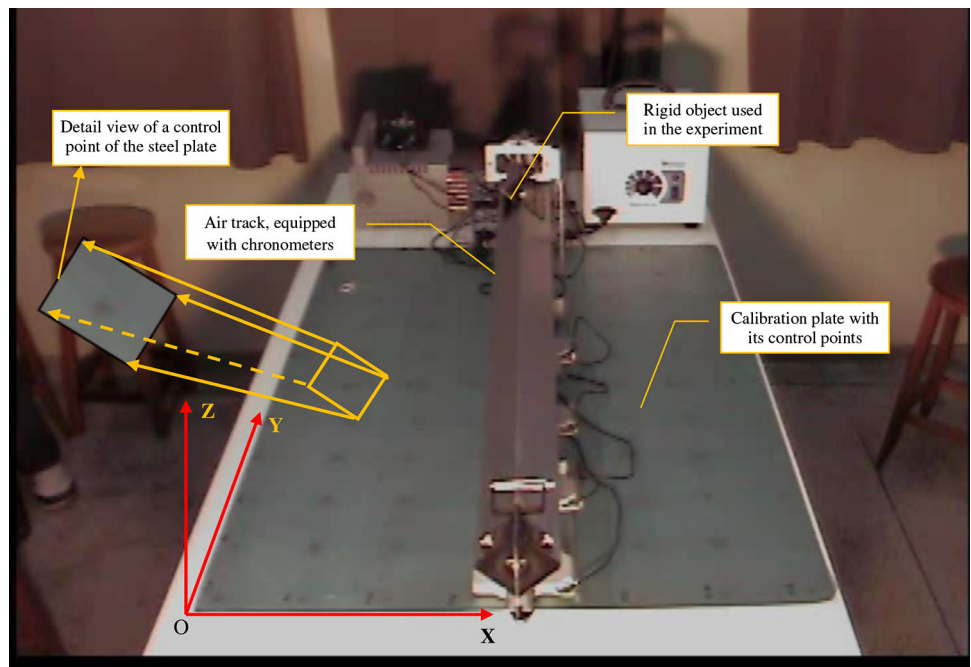
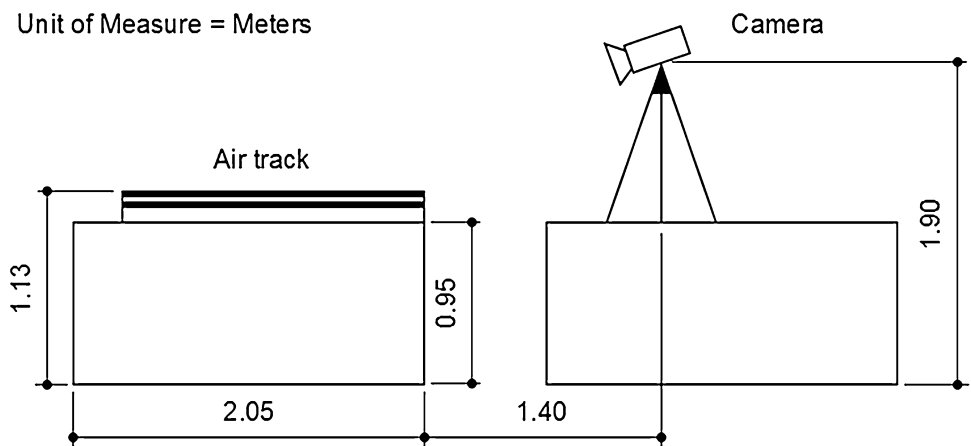


Fig. 8 Environment setup of the experiment



installed (Fig. 7), which permits to determine the points that are related between the space object and the plane image. The experiment, with the object moving over the air track, was filmed and the images acquired submitted to the proposed system, through which the velocity was calculated by image analysis and photogrammetric calculations. The value of the calculated velocity was compared with the real velocity value computed by the sensors on the air track.

4.3 Results

This section describes the results concerning the calculation of the velocity in a straight line of a rigid object with null acceleration. To do so, a linear regression was

performed, using the least squares method (LSM) [11, 25] to enable velocity interpolation. This was performed using a linear model $y = b + ax$, in which the linear b and angular a coefficients values may be estimated with the use of the following expressions:

$$a = \frac{n \sum_1^n x_i y_i - (\sum_1^n x_i)(\sum_1^n y_i)}{n \sum_1^n x_i^2 - (\sum_1^n x_i)^2} \tag{28}$$

$$b = \frac{\sum_1^n y_i - a \sum_1^n x_i}{n} \tag{29}$$

The following expressions permit to determine the deviations occurred in the calculation of the slope of the fit of the data for the linear model, a , Δa , as well as determining the linear coefficient b , Δb .

Table 3 Data obtained in the experiments performed in the laboratory and their respective calculated velocities

	Experimental data					Calculated velocity (cm/s)
	Time intervals obtained at each experiment					
	Δt_0 (s)	Δt_1 (s)	Δt_2 (s)	Δt_3 (s)	Δt_4 (s)	
Experiment 1	0.0000	0.5150	1.0620	1.5540	2.1230	28.4 ± 0.3
Experiment 2	0.0000	0.5200	1.0730	1.5690	2.1470	28.1 ± 0.3
Experiment 3	0.0000	0.5200	1.0780	1.6180	2.2710	26.6 ± 0.6
Experiment 4	0.0000	0.5210	1.0760	1.5780	2.1610	27.9 ± 0.3
$\Delta \bar{t}$ (s)	0.0000	0.5190	1.0722	1.5798	2.1755	27.7 ± 0.4
Δs (cm)	0.0000	15.000	30.000	45.000	60.000	

$\Delta \bar{t}$ —Time interval average values

Δt_i —Value of each time interval i measured in the experiment

Δs —Value of the covered distance by the rigid object in each time interval

$$\Delta a = \frac{\sqrt{n} \cdot \sigma}{\sqrt{n \sum_1^n x_i^2 - (\sum_1^n x_i)^2}} \tag{30}$$

$$\sigma = \sqrt{\frac{\sum_1^n (y_i - ax_i - b)^2}{n - 2}} \tag{31}$$

$$\Delta b = \Delta a \cdot \sqrt{\frac{\sum_1^n x_i^2}{n}} \tag{32}$$

The slope (angular coefficient) is written as $a \pm \Delta a$, and the intercept (linear coefficient) as: $b \pm \Delta b$. The correlation coefficient (r) is a parameter for the study of bi-dimensional distributions, which indicates the level of dependency among associated data with the variables X and Y . The correlation coefficient r is a value obtained by applying the following expression:

$$r = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n (x_i - \bar{x})^2} \sqrt{\sum_1^n (y_i - \bar{y})^2}} \tag{33}$$

with $\bar{x} = \frac{\sum_1^n x_i}{n}$ and $\bar{y} = \frac{\sum_1^n y_i}{n}$.

Knowing that $r \in [-1, +1]$, and is interpreted in the following way:

- $r = +1$, indicates that the linear correlation between values X and Y is perfect and direct;
- $r = -1$, indicates that the linear correlation between values X and Y is perfect and inverse;
- $r = 0$, indicates that there is no correlation, i.e., there is total independence of values X and Y .

4.3.1 Computed results using photoelectric sensors

Table 3 exhibits the velocity of a rigid object computed by the sensors on the air track.

Time data collected in the four experiments were used to calculate the respective $\Delta \bar{t}$ average values. These average

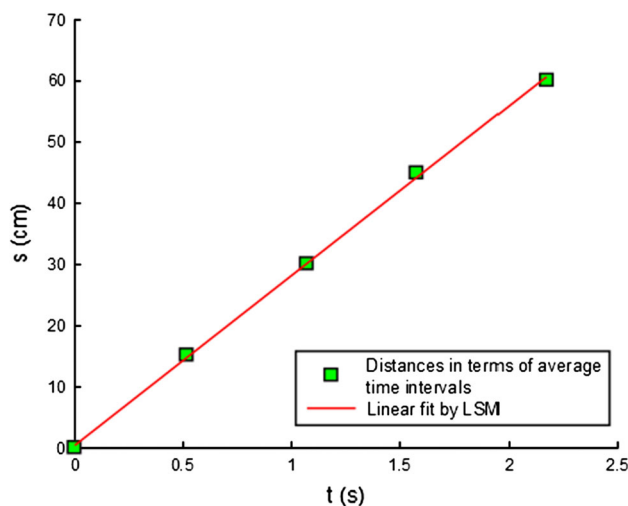


Fig. 9 Linear fit of the distances s using LSM as a function of the average time intervals $\Delta \bar{t}$

values minimize the occurrence of systematic errors occurred during the performance of these experiments. Only these average values were used in the analysis performed in this study.

The linear equation (Fig. 9) calculated using LSM was $s = 0.4 + 27.7t$. The correlation factor (Fig. 9) of the calculated line using LSM was 0.9997. This value demonstrated that the experimental data were strongly correlated with the linear model of uniform linear motion $s = s_0 + vt$, where: s is the covered distance from an initial position s_0 as a function of time t with constant velocity v .

4.3.2 Computed results based on collinearity equations and polynomial transformation method using automatic object segmentation

Table 4 illustrates the data obtained by the software developed, using the segmentation method developed by

Table 4 Automatic data extraction from images sequence acquired in the laboratory experiments

Automatic point extraction									
Frame	Time (s)	Plane image coordinates		Space object coordinates and distance computation					
				Polynomial transformation			Collinearity equations		
		Xi	Yi	Xo	Yo	Distance (mm)	Xo	Yo	Distance (mm)
1	0.000000	367	211	311.122	803.369	0.0000	323.655	424.786	0.0000
2	0.068966	367	213	311.428	791.296	12.0729	323.856	415.896	8.8900
3	0.103448	367	216	311.868	773.609	29.7602	324.154	402.727	22.0600
4	0.137931	367	218	312.150	762.081	41.2881	324.351	394.056	30.7300
5	0.172414	366	218	309.811	762.061	41.3081	322.508	394.074	30.7100
7	0.241379	367	223	312.819	734.107	69.2619	324.833	372.750	52.0400
8	0.275862	367	226	313.198	717.857	85.5121	325.116	360.213	64.5700
9	0.310345	367	228	313.442	707.229	96.1401	325.303	351.956	72.8300
10	0.344828	367	231	313.797	691.577	111.7920	325.579	339.718	85.0700
11	0.379310	367	233	314.026	681.327	122.0420	325.761	331.656	93.1300
12	0.413793	366	235	312.035	671.205	132.1640	324.174	323.687	101.1000
13	0.448276	366	237	312.265	661.230	142.1390	324.361	315.776	109.0100
14	0.482759	367	240	314.786	646.518	156.8510	326.385	304.032	120.7500
15	0.517241	368	242	317.165	636.864	166.5050	328.299	296.289	128.5000
16	0.551724	368	245	317.448	622.590	180.7790	328.546	284.830	139.9600
18	0.620690	369	250	320.027	599.356	204.0130	330.657	266.064	158.7200
19	0.655172	369	253	320.271	585.722	217.6470	330.883	255.017	169.7700
20	0.689655	369	255	320.430	576.755	226.6140	331.031	247.734	177.0500
21	0.724138	369	257	320.586	567.883	235.4860	331.179	240.516	184.2700
22	0.758621	370	260	322.891	554.746	248.6230	333.067	229.795	194.9900
23	0.793103	370	263	323.098	541.803	261.5660	333.272	219.229	205.5600
24	0.827586	370	265	323.234	533.280	270.0900	333.408	212.262	212.5200
25	0.862069	378	267	339.721	524.810	278.5590	346.691	205.242	219.5400
26	0.896552	379	270	341.839	512.283	291.0860	348.437	194.981	229.8100
27	0.931034	378	273	339.891	499.941	303.4280	346.916	184.880	239.9000
28	0.965517	379	276	341.979	487.747	315.6220	348.638	174.880	249.9100
29	1.000000	378	279	340.054	475.726	327.6430	347.135	165.033	259.7500
30	1.034480	376	282	336.172	463.863	339.5060	344.060	155.324	269.4600

The space object coordinates transformations and the distance computation by methods: polynomial transformation and collinearity equations

[15], and the coordinate conversion method from plane image to space object based on the collinearity equations and polynomial transformation method.

The values for the distance and time intervals were extracted from the sequences of image frames, which resulted in a velocity of 26.691 cm/s with a correlation factor equivalent to 0.999 (Fig. 10a) for the use of collinearity equations, and a velocity of 33.484 cm/s with a correlation factor equivalent to 0.999 (Fig. 10c) for the use polynomial transformation method.

Comparing the velocity values obtained by the method based on the collinearity equations (0.691 cm/s) with the value obtained by the photoelectric sensors (27.7 cm/s), the deviation was in the order of 3.64 %, whereas for the

velocity values obtained by the polynomial transformation method (33.484 cm/s) with the value obtained by the photoelectric sensors (27.7 cm/s), the deviation was in the order of 20.88 %.

4.3.3 Computed results based on collinearity equations and polynomial transformation method using manual object segmentation

Table 5 illustrates the data obtained by the manual capture of points in the plane image, using the coordinate conversion method from plane image to space object based on the collinearity equations and polynomial transformation method.

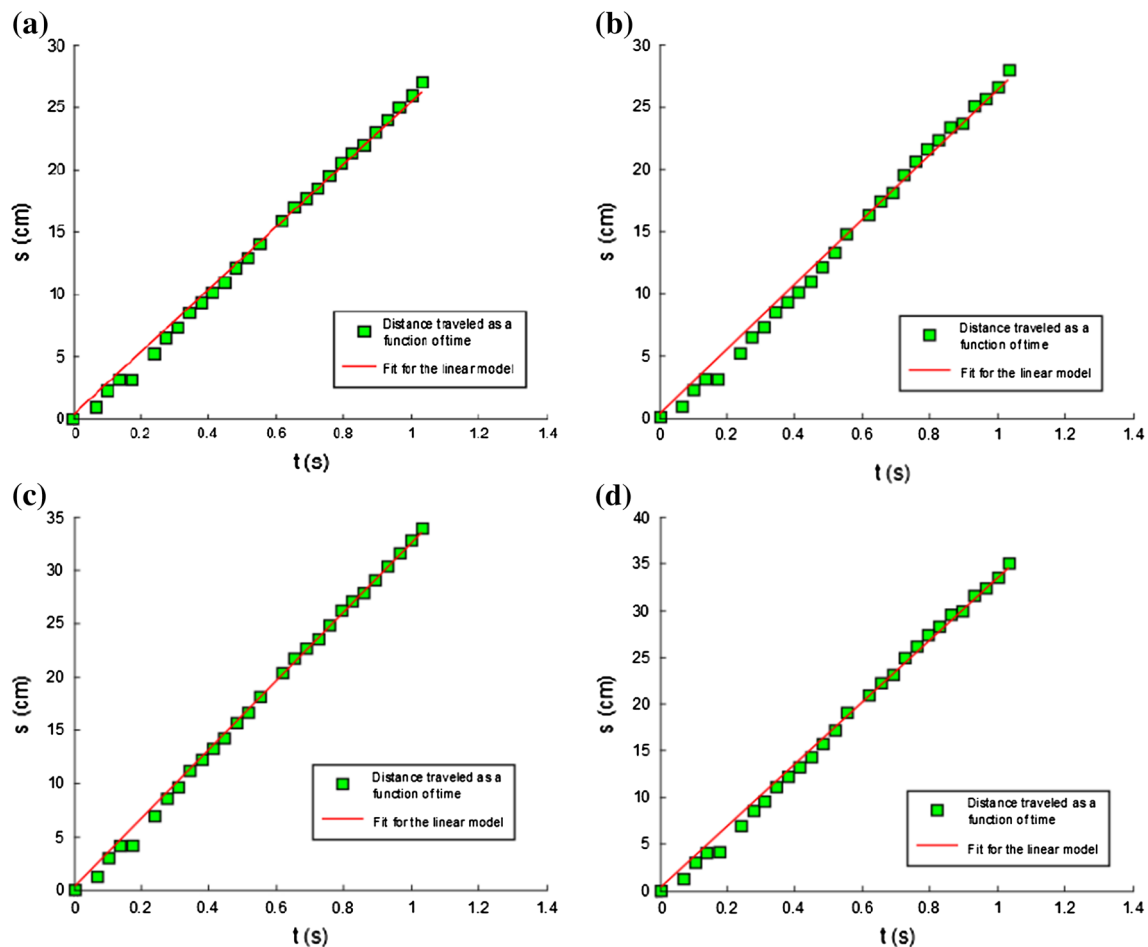


Fig. 10 Linear fit of the time (t) versus distance (s) using LSM. **a** Result Computed by collinearity equations using automatic object segmentation. **b** Result Computed by collinearity equations using manual object segmentation. **c** Result Computed by polynomial

transformation method using automatic object segmentation. **d** Result Computed by polynomial transformation method using manual object segmentation

The values for the distances and time intervals were extracted from the sequence of image frames, which resulted in a velocity of 27.905 cm/s with a correlation factor equivalent to 0.9980 (Fig. 10b) for the use of collinearity equations, and a velocity of 34.99 cm/s with a correlation factor equivalent to 0.9983 (Fig. 10d) for the use polynomial transformation method.

Comparing the velocity values obtained by the transformation method based on the collinearity Eqs. (27.905 cm/s) with the value obtained by the photoelectric sensors (27.7 cm/s), the deviation was in the order of 0.74 %, whereas for the velocity values obtained by the polynomial transformation method (34.99 cm/s) with the value obtained by the photoelectric sensors (27.7 cm/s), the deviation was in the order of 26.31 %.

4.3.4 Analysis of the results

Table 6 illustrates the results obtained with the tests performed, highlighting the velocity measured by the automatic detection of rigid objects using the segmentation method described by [15] and by manual detection. It also shows the error in percentage caused by the automatic identification of rigid objects in the velocity measurement process, and the error in percentage of the coordinate conversion methods from plane image to space object in comparison with the velocity obtained by the photoelectric sensors data.

Based on the results obtained, it is possible to state that the technique used for the automatic identification of moving rigid objects may lead to errors in the process of

Table 5 Manual data extraction from images sequence acquired in the laboratory experiments

Manual point extraction									
Frame	Time (s)	Plane image coordinates		Space object coordinates and distance computation					
				Polynomial transformation			Collinearity equations		
		Xi	Yi	Xo	Yo	Distance (mm)	Xo	Yo	Distance (mm)
1	0.00000	367	211	311.122	803.369	0.0000	323.655	424.786	0.0000
2	0.06897	367	213	311.428	791.296	12.0729	323.856	415.896	8.8900
3	0.10345	367	216	311.868	773.609	29.7602	324.154	402.727	22.0592
4	0.13793	367	218	312.150	762.081	41.2881	324.351	394.056	30.7300
5	0.17241	366	218	309.811	762.061	41.3081	322.508	394.074	30.7121
7	0.24138	367	223	312.819	734.107	69.2619	324.833	372.75	52.0368
8	0.27586	367	226	313.198	717.857	85.5121	325.116	360.213	64.5733
9	0.31035	367	228	313.442	707.229	96.1401	325.303	351.956	72.8304
10	0.34483	367	231	313.797	691.577	111.7920	325.579	339.718	85.0683
11	0.37931	367	233	314.026	681.327	122.0420	325.761	331.656	93.1301
12	0.41379	366	235	312.035	671.205	132.1640	324.174	323.687	101.1000
13	0.44827	366	237	312.265	661.230	142.1390	324.361	315.776	109.0100
14	0.48276	365	240	310.418	646.498	156.8710	322.891	304.063	120.7230
15	0.51724	365	243	310.761	632.052	171.3170	323.176	292.497	132.2890
16	0.55172	366	247	313.344	613.201	190.1680	325.27	277.308	147.4780
18	0.62069	364	251	309.502	594.756	208.6130	322.214	262.438	162.3480
19	0.65517	364	254	309.823	581.200	222.1690	322.497	251.439	173.3470
20	0.68966	365	256	312.126	572.290	231.0790	324.368	244.174	180.6120
21	0.72414	364	260	310.437	554.725	248.6440	323.05	229.879	194.9080
22	0.75862	364	263	310.732	541.787	261.5830	323.322	219.311	205.4750
23	0.79310	364	266	311.018	529.036	274.3340	323.589	208.882	215.9040
24	0.82759	364	268	311.205	520.635	282.7340	323.766	202.004	222.7820
25	0.86207	364	271	311.479	508.180	295.1900	324.027	191.799	232.9880
26	0.89655	364	272	311.569	504.065	299.3040	324.114	188.426	236.3600
27	0.93103	364	276	311.920	487.790	315.5790	324.456	175.078	249.7080
28	0.96552	364	278	312.092	479.757	323.6120	324.625	168.489	256.2970
29	1.00000	364	281	312.344	467.836	335.5330	324.875	158.709	266.0770
30	1.03448	364	285	312.671	452.169	351.2010	325.204	145.858	278.9280

The space object coordinates transformations and the distance computation by methods: polynomial transformation and collinearity equations

Table 6 Results of the tests performed in the velocity measurement

Method used in velocity measurement	Velocity (cm/s) obtained with the identification of objects		Error (%) relative to photoelectric sensors (velocity measured by object identification)	
	Automatic	Manual	Automatic	Manual
Photoelectric sensors	27.700	–	–	–
Collinearity equations	26.691	27.905	3.64	0.74
Polynomial transformation method	33.484	34.990	20.80	26.31

measuring the velocity. This was due to the fact that errors occurred in the identification of the lower left-hand corner (Fig. 11) of the moving rigid object used as a reference point to perform the velocity measurement.

5 Final considerations

This study presented, implemented and experimented a methodology for the velocity measurement of a rigid object in motion, having as the starting point a monocular images sequence acquired by a digital video camera.

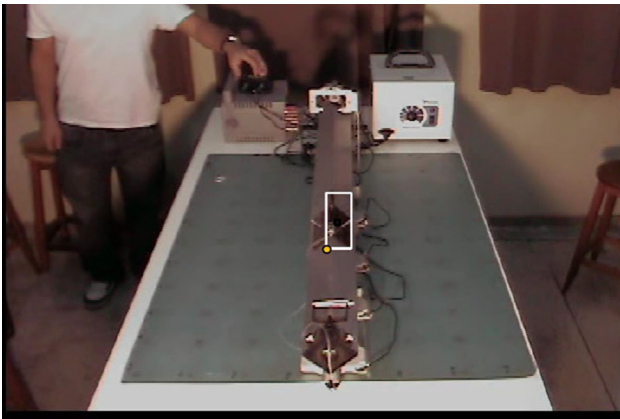


Fig. 11 Capture of the moving rigid object with the reference point identified wrongly

The application of the modified collinearity model presented in the experiments conducted to the best results in comparison to the other models. In the best result, the estimated velocity was 27.90 cm/s, with an error of 0.74 % in relation to the velocity obtained by the photoelectric sensors (27.7 cm/s).

It is necessary to complement this study approaching the influence of scale variation in image acquisition and know if the use of points belonging to the object, with different elevation values on the object of interest have influence on the accuracy of the results achieved by the application the proposed methods.

The implemented and tested prototype in controlled conditions, with the proper considerations, demonstrated that the methodology is valid. However, further experimental tests should be performed, especially with the use of different camera viewing angles, using different distances between the camera and the scene (scale factor) and using different frame rate image acquisition to better understand how these factors may influence the results.

The advantages of using the proposed methods in this paper are:

1. need inexpensive and commercially available technology;
2. not need to use other types of sensors, only the video camera is sufficient;
3. has high computational efficiency because the required algorithms have linear complexity;
4. does not need to make calculations related to reconstruction of 3D images acquired at different viewing angles (stereo pairs).

The disadvantages of the use of the proposed methods in this paper are:

1. these methods are based on the analysis of images, then you must use a robust segmentation method that is

able to overcome the problems encountered by the scene environment;

2. it is necessary that specific points of the scene (space object) with known coordinates are previously established and these points must be identified at any time operating system.

References

1. Aguilar, M.A., Aguilar, F.J., Agüera, F., Sánchez, J.A.: Geometric accuracy assessment of quickbird basic imagery using different operational approaches. *Photogramm. Eng. Remote Sens.* **73**(12), 1321–1332 (2007)
2. Atkociunas, E., Blake, R., Juozapavicius, A., Kazimianec, M.: Image processing in road. *Traffic Anal., Nonlinear Anal. Model. Control* **10**(4), 315–332 (2005)
3. Avidan, S., Shashua, A.: Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(4), 348–357 (2000). doi:[10.1109/34.845377](https://doi.org/10.1109/34.845377)
4. Barranco, F., Tomasi, M., Diaz, J., Vanegas, M., Ros, E.: Parallel architecture for hierarchical optical flow estimation based on FPGA. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* **20**(6), 1058–1067 (2012)
5. Bonneval, H.: Levés topographiques par photogrammétrie aérienne. In *Photogrammétrie Générale: Tome 3, Collection scientifique de l'Institut Géographique National*. Paris, France: Eyrolles Editeur 1972
6. Botella, G., Garcia, A., Rodriguez, M., Ros, E., Baese, U., Molina, M.: Robust bioinspired architecture for optical flow computation. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* **18**(4), 616–629 (2010)
7. Botella, G., Ros, E., Rodriguez, M., Garcia, A., Romero, S.: Pre-processor for bioinspired optical flow models: a customizable hardware implementation. *Proceedings of the IEEE Mediterranean Electrotechnical Conference (MELECON)*, Málaga, Spain, 93–96 (2006). doi:[10.1109/MELCON.2006.1653044](https://doi.org/10.1109/MELCON.2006.1653044)
8. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vision* **61**(3), 211–231 (2005). doi:[10.1023/B:VISI.0000045324.43199.43](https://doi.org/10.1023/B:VISI.0000045324.43199.43)
9. Davis, J., Bobick, A.: The representation and recognition of human movement using temporal templates. *IEEE Computer Soc Conf Computer Vision Pattern Recognit (CVPR'97)*, 928–934 (1997). doi:[10.1109/CVPR.1997.609439](https://doi.org/10.1109/CVPR.1997.609439)
10. Davis, J., Bradiski, G.: Real-time motion template gradients using Intel CVLib. In: *proceedings of the ICCV Workshop on Frame-rate Vision*, 1–20 (1999)
11. Ghilani, C.D.: *General Least Squares Method and Its Application to Curve Fitting and Coordinate Transformations, in Adjustment Computations: Spatial Data Analysis (Fifth Edition)*. Hoboken, NJ, USA: Wiley, Inc. (2010). doi:[10.1002/9780470586266.ch22](https://doi.org/10.1002/9780470586266.ch22)
12. Gupte, S., Masoud, O., Martin, R.F.K., Apanikolopoulos, N.P.: Detection and classification of vehicles. *IEEE Trans. Intell. Transp. Syst.* **3**(1), 37–47 (2002). doi:[10.1109/6979.994794](https://doi.org/10.1109/6979.994794)
13. Habib, Ayman F., Morgan, Michel F.: Automatic calibration of low-cost digital cameras. *Opt. Eng.* **42**(4), 948–955 (2003). doi:[10.1117/1.1555732](https://doi.org/10.1117/1.1555732)
14. Hamid, N.F.A., Ahmad, A.: Calibration of high resolution digital camera based on different photogrammetric methods. *Proceedings of the 8th International Symposium of the Digital Earth*

- (ISDE8), IOP Publishing on IOP Conf. Series. Earth Environ Sci **18**, 1–6 (2014). doi:[10.1088/1755-1315/18/1/012030](https://doi.org/10.1088/1755-1315/18/1/012030)
15. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* **11**(3), 167–256 (2005). doi:[10.1016/j.rti.2004.12.004](https://doi.org/10.1016/j.rti.2004.12.004)
 16. Kraus K.: *Photogrammetry Vol. 1. Fundamentals and Standard Processes* (4th edition). Bonn, Germany: Ferdinand Dummlers (1993)
 17. Li, R., Niu, X., Liu, C., Wu, B., Deshpande, S.: Impact of imaging geometry on 3d geopositioning accuracy of stereo ikonos imagery. *Photogramm. Eng. & Remote Sens.* **75**(9), 1119–1125 (2009)
 18. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, vol.2, pp. 674–679. Vancouver (1981)
 19. Novak, K.: Rectification of digital imagery. *Photogramm. Eng. and Remote Sens.* **58**(3), 339–344 (1992)
 20. Petrie, G., El Niwari, A.E.H.: The applicability of space imagery to the small scale topographic mapping of developing countries: a case study—the Sudan. *ISPRS J. Photogramm. Remote Sens.* **47**(1), 1–42 (1992). doi:[10.1016/0924-2716\(92\)90002-Q](https://doi.org/10.1016/0924-2716(92)90002-Q)
 21. Soh, J., Chun, B.T., Wang, M.: Analysis of road image sequences for vehicle counting. *IEEE Trans. Intell. Transp. Syst.* **1**, 679–683 (1995). doi:[10.1109/ICSMC.1995.537842](https://doi.org/10.1109/ICSMC.1995.537842)
 22. Toutin, T.: Review article: Geometric processing of remote sensing images: models, algorithms and methods. *Int. J. Remote Sens.* **25**(10), 1893–1924 (2004). doi:[10.1080/0143116031000101611](https://doi.org/10.1080/0143116031000101611)
 23. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Autom.* **3**(4), 323–344 (1987). doi:[10.1109/JRA.1987.1087109](https://doi.org/10.1109/JRA.1987.1087109)
 24. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(10), 965–980 (1992). doi:[10.1109/34.159901](https://doi.org/10.1109/34.159901)
 25. Wolberg, J.: *Data analysis using the method of least squares: extracting the most information from experiments*. Berlin and Heidelberg GmbH & Co. KG, Germany: Springer (2005)
 26. Wong, K.W.: Basic mathematics of photogrammetry. In: Slama, C.C. (ed.) *Manual of Photogrammetry* (4th edition), pp. 37–101. ASP Publishers, Falls Church (1980)
 27. Yan, Y., Yancong, S., Zengqiang, M.: Research on vehicle speed measurement by video image based on Tsai's two stage method. In: *proceedings of the 5th International Conference on Computer Science and Education*, 502–506 (2010). doi:[10.1109/ICCSE.2010.5593565](https://doi.org/10.1109/ICCSE.2010.5593565)
 28. Zhiwei, H., Yuanyuan, L., Xueyi, Y.: Models of vehicle speeds measurement with a single camera. *Proceedings of the International Conference on Computational Intelligence and Security Workshops*, 283–286 (2007). doi:[10.1109/CISW.2007.4425492](https://doi.org/10.1109/CISW.2007.4425492)



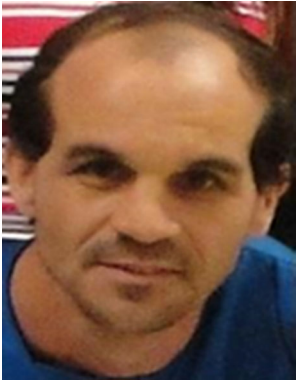
Danilo Filitto master in Computer Science by State University of Maringá (UEM), postgraduate in Computer Networks and Data Communication by State University of Paraná (UEP), bachelor in Computer Science by University of Oeste Paulista (UNOESTE). Acts in academia as a professor since 2006, teaching at the Union of Educational Institutions of São Paulo (UNIESP—Presidente Prudente Campus), and in the National Commercial Training Service (Senac—Presidente Prudente Campus). Area of Research/Expertise: Software Development, Data Structure, Digital Image Processing, Computer Networks.



Júlio Kiyoshi Hasegawa Degree in Cartographic Engineering by State University Paulista Júlio de Mesquita Filho (UNESP—Presidente Prudente Campus), master in Geodetic Sciences by Federal University of Paraná (UFPR) and doctorate in Electrical Engineering by State University of Campinas (UNICAMP). Currently acts as adjunct professor at the State University Paulista Júlio de Mesquita Filho (UNESP—Presidente Prudente Campus), with experience in Geosciences with an emphasis in Photogrammetry, acting on the following topics: phototriangulation, exterior orientation, photogrammetry, digital photogrammetry and GPS.



Airton Marco Polidório degree in Chemical Engineering from the State University of Maringá (UEM), master in Electrical Engineering and Industrial Informatics by Federal Technological University of Paraná (UTFPR), and doctorate in Cartography by State University Paulista Júlio de Mesquita Filho (UNESP—Presidente Prudente Campus). Currently acts as adjunct professor at the State University of Maringá (UEM), with experience in Pattern Recognition, Computer Vision, and Image Processing.



Nardênio Almeida Martins master in Electrical Engineering at Federal University of Santa Catarina (UFSC), and doctorate in Automation and Systems Engineering at Federal University of Santa Catarina (UFSC). Currently acts as adjunct professor at the State University de Maringá (UEM), with research concentrated in the areas of control dynamical systems, robot manipulators, mobile robots and control mechatronic systems.



Franklin César Flores doctorate in Electrical Engineering by State University of Campinas (UNICAMP), master in Computer Science by University of São Paulo (USP), and Bachelor in Computer Science by State University of Maringá (UEM). Currently acts as adjunct professor in the Informatics Department of the State University of Maringa (DIN-UEM), with research interests in Digital Image Processing, Computer Vision and Mathematical

Morphology.