



# Positioning measurement using a new artificial vision algorithm in LabVIEW based on the analysis of images on an LCD screen

Óscar de Francisco Ortiz<sup>1</sup> · Manuel Estrems Amestoy<sup>2</sup> · Julio Carrero-Blanco<sup>2</sup>

Received: 28 August 2019 / Accepted: 18 May 2020

© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

Research in vision systems applied to manufacturing processes has increased during the last years. Nevertheless, accurate positioning systems frequently require costly investments. This article presents a new algorithm developed in LabVIEW for controlling a novel positioning system that processes images to obtain the position, which could be implemented in micromachining. With this aim, the method uses the analysis of the LEDs shown in an image projected on an LCD screen to perform an accurate positioning. The ultimate goal of this method is to get the coordinates of the images shown on the screen in order to know the movement made by the system and, in this way, be able to compensate the error. The experimental results and related analysis developed proved the accuracy and consistency in dissimilar situations. In addition, once implemented the algorithm proposed in a closed loop program, a positioning system is achieved where the error is always convergent.

**Keywords** Image processing · Positioning control · Vision algorithm · Precision engineering · Micromachining · LabVIEW

## 1 Introduction

Positioning vision systems are widespread and are increasingly important in domestic and industrial processes. The vision provides the positioning systems with a new range of possibilities that are difficult to achieve with traditional methods, particularly reducing the investment required by accuracy achieved. However, these systems require algorithms developed with powerful and efficient software.

One of the first methods of positioning through vision was developed by Montes et al. [1], in which the position was

calculated based on series of patterns represented on an LCD screen. Subsequently, based on the same principle, but with a different methodology, de Francisco et al. [2] developed a system which included very low-cost elements, achieving positioning in the order of a few microns. Since the distance between the LCD screen and the vision camera is crucial for the calibration of the system, the problem of this measurement has been studied and analyzed from different perspectives by authors such as Cano-García et al. [3], where the analyzed information was extracted from the gray levels of the pixels so that the distance between the camera and the infrared diode (IRED) was estimated. Other authors used different methods with successful results as Wong et al. [4] and Franzi et al. [5]. Likewise, it is essential to control the processing of the image, especially to perform a correct measurement and analysis of the relative position of one image with respect to another [6, 7]. The compensation of sources of error in the vision system, such as lens distortion, lens blurring, and the non-linear pattern of strips, is a complex and laborious procedure that has been studied deeply [8], using sometimes very precise mathematical simplifications [9]. Similarly, the methods used for the calibration of the machine-system where these vision systems are implemented are equally critical for obtaining results that minimize errors [10], thus maximizing the overall positioning of the system. LabVIEW software has been used by different authors for digital image processing [11, 12] and

✉ Óscar de Francisco Ortiz  
oscar.defrancisco@tud.upct.es

Manuel Estrems Amestoy  
manuel.estrems@upct.es

Julio Carrero-Blanco  
julio.carrero@upct.es

<sup>1</sup> Engineering and Applied Techniques Department, University Center of Defense, San Javier Air Force Base, MDE-UPCT, C/Coronel López Peña, s/n, Santiago de la Ribera, 30720 Murcia, Spain

<sup>2</sup> Department of Materials and Manufacturing Engineering, Polytechnical University of Cartagena, 30202 Cartagena, Spain

recognition [13]. In addition, it has been used, among others, to develop robust algorithms to move motors [14], manipulate micro-objects [15], or study multi-phase fluid flow in microfluidic devices [16] with very satisfactory results.

Recently, the use of vision system is rapidly increasing in different fields as inspection [17], welding [18], wood engineering [19, 20], or robot's [21, 22] position control. Vision positioning systems are increasingly common in process automation [23–26], autonomous driving [27–31], or augmented reality assistants [32–34]. Indeed, this is one of the most promising elements in the Industry 4.0 revolution. In addition, pose estimation and marker detection using artificial vision algorithms are widely used tasks for many other technological applications such as autonomous robots [35–37], unmanned vehicles [38–44], and virtual assistants [45–48], among others.

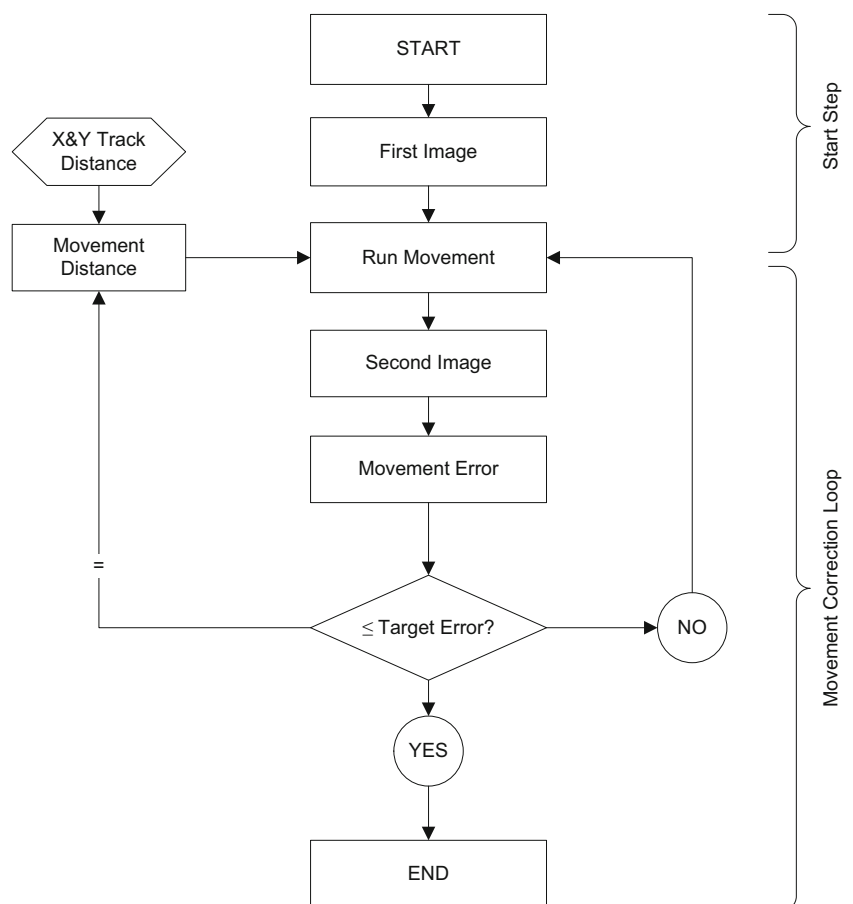
In this paper, we describe the new algorithm developed based on LabVIEW to determine the movement and error compensation of a micro-positioning system. LabVIEW was selected due the advantages of this graphical programming where the code is flexible, reusable, and self-documenting [49].

## 2 Methodology

The developed code focuses on the processing of the image that is taken by the vision camera. Each illuminated LED pattern represents a set of possible target positions for an operation (for example micromachining) as studied by Leviton [50] and Chu at al. [51]. This code has been designed taking into account the experimental equipment that will be used for its validation. The image processing has been developed as a stable and global treatment that allows analyzing all the image captures whatever the conditions in which they have been taken. The basic diagram of the software design is shown in Fig. 1.

Two images are made in each iteration and between these two images; the movement will be performed as shown in Fig. 2. The analysis of these images is calculated immediately after each capture. When the second image is taken, the position error is calculated considering the first and the second image. This error is compared with the inserted objective error determined in the system. As long as the position error remains above the objective error, the system will continue with more

**Fig. 1** Software diagram of the positioning system



iterations to minimize it. All these processes and the calculations required to execute them are explained in detail in this work.

## 2.1 Bitmap file

In the LCD screen used to reference the positioning of the camera-screen system, it is necessary to light a pattern or series of LEDs. A 24-bit red-green-blue (RGB) bitmap file is required to create the image that is used on the LCD screen. There are 8 bits assigned to each component: red, green, and blue. In each component, the value of 0 indicates that component makes no contribution of that color and 255 refers to the fully saturated contribution of that color. As each component has 256 different states, there is a total of 16,777,216 possible colors as shown in Fig. 3. Image segmentation is very essential and critical to image processing and pattern recognition as described by Cheng et al. [52].

The bitmap file was created over a Visual Basic (VB) code within an Excel file. The Excel format has been used because it is easy to see the color and intensity within the mesh of pixels in Excel sheets. Four sheets are opened in the Excel BMP file. The first one contains the necessary input data to create the file and, in the rest, the information about red, green, and blue color is introduced. In this work, only the “red color” sheets are used since the image patterns that are generated only using red color have been checked to optimize the subsequent detection with the vision algorithm [53].

To be able to represent the bitmaps created, a mobile phone LCD screen is used. The pixels distance will limit the theoretical maximum accuracy of the system. A comparison of the resolution and pixels per inch of the LCD screens used in some of the most common current mobile devices is shown

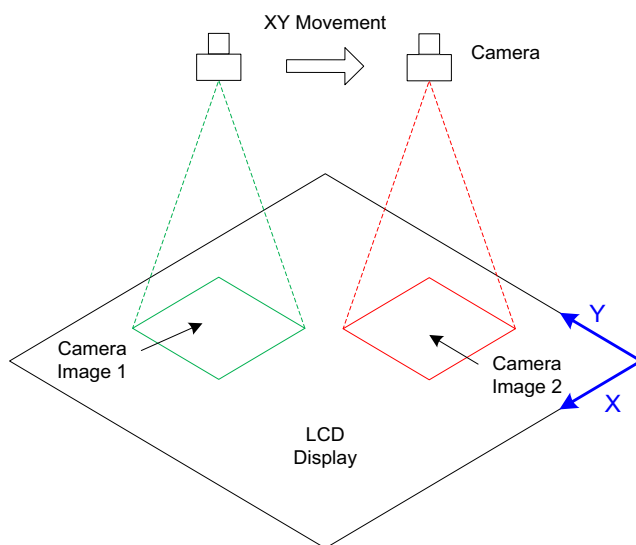


Fig. 2 Software diagram of the positioning system

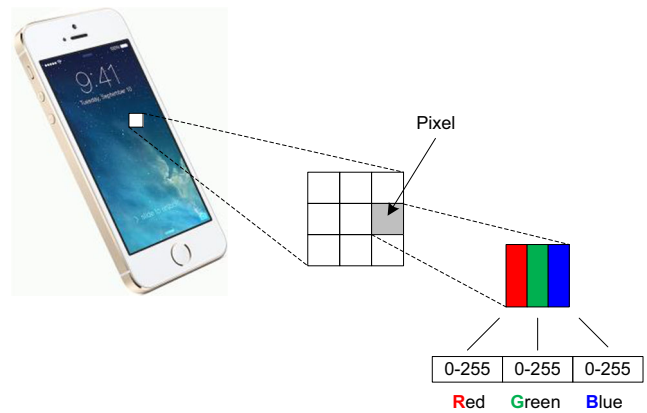


Fig. 3 Composition of the values of a pixel

in Table 1. The newest terminals provide LCD screens with greater resolution and size, and, therefore with increased density of pixels (pixels per inch or ppi).

The lighting on the screen is one of the most important aspects. Subsequently, it has been focused to obtain the best lighting conditions for the LCD screen. During the generation of the images to be displayed on the LCD, the pixel illumination and the mesh pattern can be changed. Therefore, different configurations were tried to find the best image captures that can be obtained with the system. Figure 4 shows fragments of the complete bitmap image (total mesh size of  $640 \times 1136$  pixels) created for the LCD of a mobile phone (iPhone 5s model) in which the pixel intensity values and the illuminated pixel pattern are indicated (1 pixel illuminated every 10, 3, and 2 pixels off respectively).

## 2.2 Image processing

To apply the vision system, it is necessary to transform the physical coordinate of object into the image information acquired by CCD camera [54]. First, a treatment must be performed for all the images taken by the camera. These images pass through filters that are necessary to perform, later, their mathematical analysis (Fig. 5).

As original source, a 24-bit bitmap image (BMP format,  $1280 \times 1024$  resolution) is used to start the analysis of a photo. Using the removing color plane module, one of the three color planes of any image is extracted. The red level will be obtained because only the red LEDs on the LCD screen are lighted. This filter avoids the interferences found in poor-quality photos, such as those that can be found in the limits of illumination over another unlit LED or in errors in the capture of the camera. Figure 6(a) shows the extraction of the red plane from the original image. This module is used to extract a black and white capture from the red plane providing a new image, which will be clearer than the original image. A mask of a region of interest (ROI) needs to be created to focus the processing on a particular region of an image [55–57].

**Table 1** Comparison of LCD screens in different mobile phones

Model	ppi	LCD size (in)	Resolution	Pixel distance (mm)
Ipad pro 10.5"	226	10.5	1668 × 2224	0.112
iPhone 5s <sup>1</sup>	326	4	640 × 1136	0.078
iPhone X	458	5.8	1125 × 2436	0.055
Google Pixel 3XL	523	6.3	1440 × 2960	0.049
Sony Xperia XZ3	396	6	1024 × 2880	0.064
Hauwei P20 Pro	408	6.1	2240 × 1080	0.062
LG G8 ThinQ	564	6.1	1440 × 3120	0.045
Xiaomi Mi 9	403	6.39	1080 × 2340	0.063
Samsung Galaxy S10+	522	6.4	1440 × 3040	0.049

<sup>1</sup> Mobile phone selected for the experimental tests

An image mask is an image that contains values of 1 and 0. The image mask module processes those pixels only in the image source that has a corresponding mask image value of 1. The color exclusion area is shown in Fig. 7 in blue and the valid area inside the green square. This step trims the processed image, which allows eliminating, or at least reducing, two errors: edge of the image (blurring effect [58, 59]) and computational cost [60].

Threshold is used to isolate color characteristics for further analysis and processing or to eliminate unnecessary features based on their color. The result is a binary image (Fig. 8(a)).

For the processing of the image, automatic threshold entropy is used. Image thresholding is the process by an optimal threshold which is sought to distinguish in an image the background objects of the foreground objects [61, 62]. Entropy-based methods result in algorithms that use, among others, the entropy of objects, the regions of the foreground and background, or the entropy crossed between the original image and the binarized image. This class of algorithms takes advantage of the entropy of the distribution of gray levels in a scene. The maximization of the entropy of the image is interpreted as indicative of the maximum information transfer [63, 64]. Some algorithms attempt to minimize the cross entropy between the input gray level image and the output binary image as an indicator of the preservation of information, or a measure of diffuse entropy [65].

The thresholding algorithm used in this research is to obtain the threshold value  $k$  by applying the theory of the information to the histogram data, so that all the level values of gray less than or equal to  $k$  belongs to a *class 0* and the greater values belong to *class 1* [66]. The entropy of the histogram means the amount of information associated with the histogram. Assuming that the probability  $p$  of occurrence of the gray level  $i$  is given by (1), the entropy of a histogram of an image with gray levels in the range  $[0, N-1]$  is given by the expression (2) where  $i$  represents the gray level value,  $k$  represents the value of the gray level chosen as the threshold,  $h(i)$  represents the number of pixels in the image in each gray level value,  $N$  represents the total number of gray levels in the image (256 for an 8-bit image),  $p_b(i)$  is the probability of the background, and  $n$  represents the total number of pixels in the image.

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)} \tag{1}$$

$$H = \sum_{i=0}^{N-1} p(i) \log_2 p_b(i) \tag{2}$$

If  $k$  is the threshold value, then the two entropies can be calculated by means of Eqs. (3) and (4):

$$H_b = \sum_{i=0}^{N-1} p(i) \log_2 p_b(i) \tag{3}$$

**Fig. 4** Examples of different images illuminating different patterns: (a) mesh pattern 10 × 10 and intensity value 80 (0–255); (b) mesh pattern 3 × 3 and intensity value 120 (0–255); (c) mesh pattern 2 × 2 and intensity value 120 (0–255)

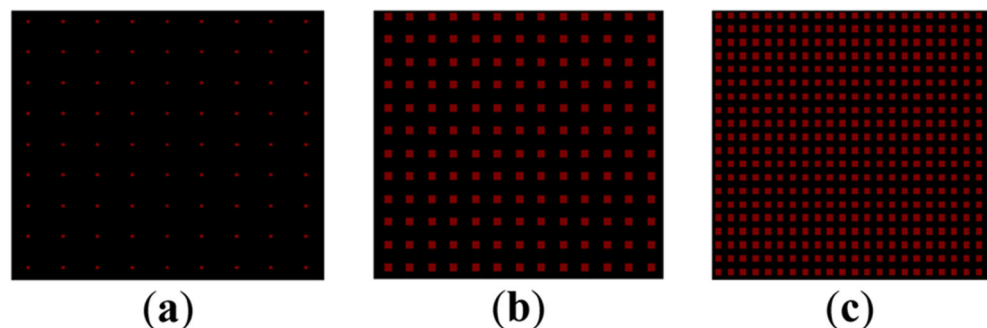




Fig. 5 Diagram-sequence of image processing used by the algorithm in LabVIEW

$$H_w = \sum_{i=k+1}^{N-1} p(i) \log_2 p_w(i) \tag{4}$$

which represent the measures of entropy (information) associated with the black ( $H_b$ ) and white ( $H_w$ ) pixels in the image after the application of the threshold, and where  $p_w(i)$  is the probability of the object.

The optimal threshold value is the gray level value that maximizes the entropy in the threshold image given by Eq. (5):

$$H_b + H_w \tag{5}$$

The threshold value is the value of pixel  $k$  in which the expression (6) is maximized:

$$Max \left[ -\frac{1}{\sum_{i=0}^k h(i)} \sum_{i=0}^k p(i) \log_2(h(i) + 1)h(i) - \frac{1}{\sum_{i=k+1}^{N-1} h(i)} \sum_{i=k+1}^{N-1} \log_2(h(i) + 1)h(i) + \log_2 \left( \sum_{i=0}^k h(i) \sum_{i=k+1}^{N-1} h(i) \right) \right] \tag{6}$$

The next step in the algorithm contains the use of advance morphology modules to perform high-level operations on particles in binary images. The first removes small particles from the image. A small object is defined by the amount of erosions (specified in Iterations) necessary to eliminate the object. The connectivity specifies how the algorithm determines whether an adjacent pixel is the same particle or a different one and square/hexagon specifies whether the pixel frame is treated as a square or hexagon during the transformation. The second module removes objects from the edge of the created mask. The result of the application of the morphology modules to an image is displayed in Fig. 8(b).

The last step consists in the analysis of particles where the vision module assigns numeric labels to all the pixels in the particle analysis and calculates the measurements that will be used. In this study, 25 particles are taken for each image, which are labeled in Fig. 8(c). Once the image is processed, the center of mass  $x_g$  and  $y_g$  of the pixels are calculated in pixel

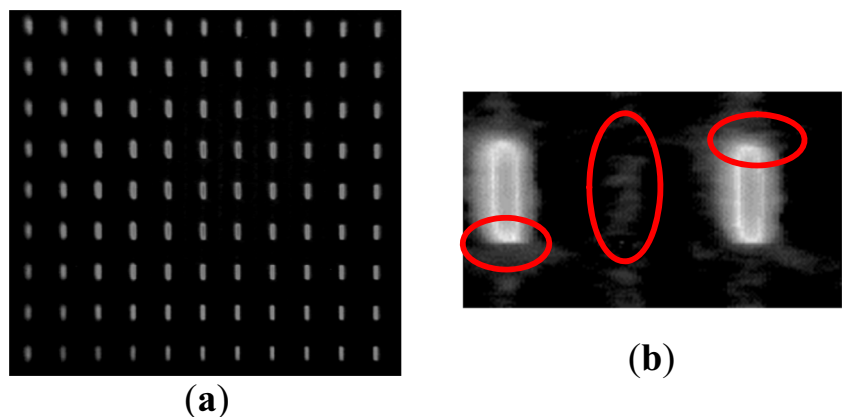
coordinates being the global position in the original image. Equations (7) and (8) are used being  $i \in [1, n]$  where  $n$  is the number of rows in the image,  $j \in [1, m]$  where  $m$  is the number of columns in the image,  $x_{i,j}$  is the  $x$ -coordinate of the element (pixel) in position  $(i,j)$ ,  $y_{i,j}$  is the  $y$ -coordinate of the element (pixel) in position  $(i,j)$ , and  $A(i,j)$  is the element (pixel) in position  $(i,j)$ .

$$x_g = \frac{\sum_{i=1}^n \sum_{j=1}^m x_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m A(i,j)} \tag{7}$$

$$y_g = \frac{\sum_{i=1}^n \sum_{j=1}^m y_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m A(i,j)} \tag{8}$$

In these equations, the intensity of the pixels is not taken into account since once the image has been post-processed by means of the modules previously described, all the pixels have the same intensity level.

Fig. 6 Removing the color plane: (a) filter to extract red color in bitmap image; (b) example of noise in an image



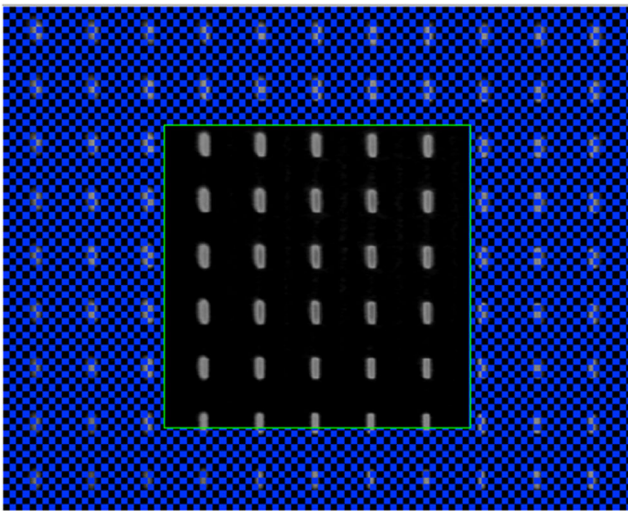


Fig. 7 Image mask applied to the original image

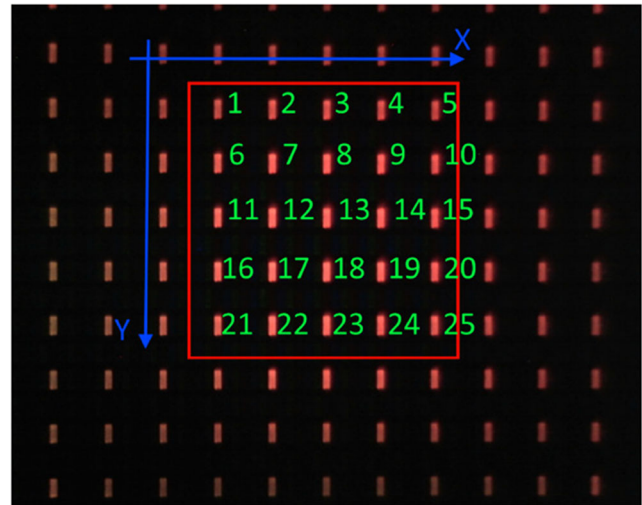


Fig. 9 Example of an ordered array matrix

### 2.3 Artificial vision assembly

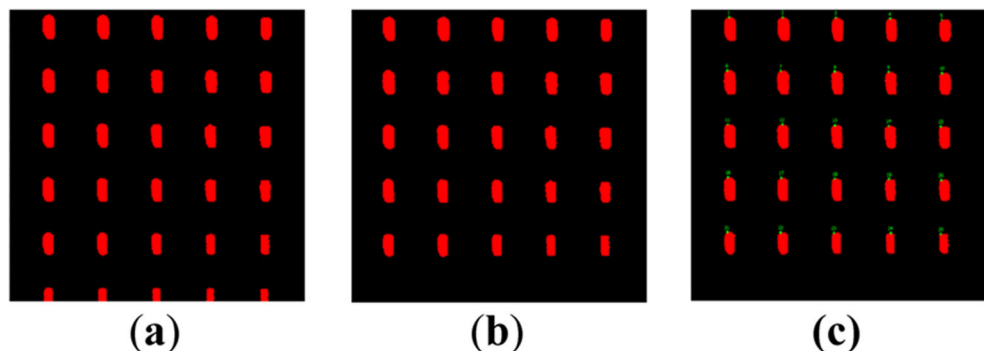
The artificial vision process programed uses the LabVIEW NI Vision Development module and for this research, it consists in five virtual instruments (VI).

- *Input matrix.* The algorithm reworks the input data in three steps: *measurements of particles* to create a matrix with all the saved positions  $x_g$  and  $y_g$  of the centers of gravity of the mesh of pixels calculated at the end of the image processing process; *divide and index the coordinates x and y*, block which separates both axes and places the coordinates of the y axis in the first position; and *order 1D matrix*, which sorts the matrix from low to high values of the y axis and reconstructs the values in a new matrix (ordered array) with the positions of the x axis are linked to their y axis values.
- *Reordered matrix.* The matrix obtained is not ordered with respect to the values of the elements. Consequently, it is required to order the values of the x axis. Therefore, the code creates groups of five elements (25 analyzed particles) and classifies each group from least to greatest along

the x axis. The y axis values are linked to their x axis values, although these values have been ordered. Afterwards, the original set of position in the matrix is rebuilt with the new values ordered along both axes providing an output matrix (Fig. 9).

- *Distance between pixels.* Once the matrix is ordered, the distance between pixels is calculated. The main operations performed for the calculation consists in four steps: (1) calculate the *matrix difference* since the artificial vision algorithm (AVA) needs to know the value of the distance between the pixels that are inside the mesh. To calculate this distance, the matrix  $n-1$  is subtracted from the original matrix. (2) *Eliminate incorrect rows.* There are four steps to eliminate the rows that subtract the first pixel from row  $n$  with the last pixel in row  $n-1$ . (3) *Alignment correction.* To correct the alignment of the pixels, the real distance between the pixels of two consecutive images is calculated. Consequently, for each pixel, the real distance  $d_r(i, j)$  between two images for the element  $(i, j)$  is calculated

Fig. 8 Image processing: (a) image after application of threshold; (b) post-processed image with advance morphology: *small objects elimination* and *edge objects removal*; (c) analysis of particles: *labeling of pixels*



(9) where  $x(i, j)$  and  $y(i, j)$  are the  $x$  and  $y$  coordinates of the element  $i, j$  in the distance matrix obtained first step.  
 (4) *Calculation of the average distance* between the pixels of an image (10), where  $D$  is the average distance between the pixels of an image and  $n$  is the number of elements used to calculate the distance in the previous iteration.

$$\sqrt{x(i, j)^2 + y(i, j)^2} = d_r(i, j) \tag{9}$$

$$\sum_{i,j=1}^n \frac{d_r(i, j)}{n} = D \tag{10}$$

The aspect ratio  $\bar{G}$  between the LCD screen and the images taken by the camera is required by the artificial vision algorithm to transfer the coordinates from pixel image units to metric units. It can be estimated using the expression (11), where *mesh* is the number of pixels uses in the image processing method (i.e., *mesh* = 2 pixels for a  $2 \times 2$  matrix) and  $d_{gap}$  is the gap distance between LEDs in the mesh selected in LCD pixels by mm.

$$\bar{G} = \frac{D}{Mesh} \times d_{gap} \tag{11}$$

- *Angle of alignment*, which allows the compensation of the misalignment between the axes movements and the LCD screen providing better system accuracy. The calculation requires the *subtraction of matrixes* where the difference between pixels on both axes is calculated using the expressions (12) and (13) for the  $x$  axis and (14) and (15) for  $y$  axis.  $\Delta X_m$  and  $\Delta Y_m$  are, respectively, the difference of  $x$  and  $y$  coordinates between elements  $m$  and 1.

$$\Delta X_5 = X_5 - X_1 \tag{12}$$

$$\Delta Y_5 = Y_5 - Y_1 \tag{13}$$

$$\Delta X_{21} = X_{21} - X_1 \tag{14}$$

$$\Delta Y_{21} = Y_{21} - Y_1 \tag{15}$$

After the removal of the rows that are not used, the slope of the line is calculated with the Eq. (16) and the average of the resulting angle is made in both axes.

$$m_m = \frac{\Delta Y_m}{\Delta X_m} \tag{16}$$

- *Mask correction*. The last step in the artificial vision assembly consists in a mask correction block to obtain the theoretical position of the following mask. According to the designed system, it is necessary to focus on a mesh of 25 pixels in both captures. The theoretical position is calculated and the coordinates of the new mask center are saved. The LabVIEW program (VI) developed is divided into three steps.

Step 1. The input data are used to calculate the target position (Fig. 10). The inputs are the  $X$  and  $Y$  axes travel;  $X$  and  $Y$  coordinates from the central mask; the  $X$  and  $Y$  coordinates of the mask and the image pixels/mm. The center of the image is calculated from the size of the original image. The coordinates are referred to the center of the original image and the local coordinates are used to calculate the center of the position of the mask when de movement has been completed

Step 2. Contains two loops for both axes with a true/false block that changes according to the direction of the axes. The goal is to focus the mask within the image taken (Fig. 11).

It includes one *loop centered of the mask* for each axis, to find solutions between the upper and lower limits that have been calculated with the center of the theoretical image and the limit of the region where the center of the mask will be established. Also, it includes a calculation of the *start position*, which gets the values that have the following position coordinates. These outputs are obtained from the calculated position and the alignment angle.

Step 3. In the third step, the distance that the mask has moved is calculated (Fig. 12). This adjustment is necessary to correct the real distance moved on both axes. It uses as inputs the number of loops that are needed to find the focus position and the distance between the illuminated pixels; the direction of the axis; and executes a reverse rotation to transform local coordinates to global coordinates of the system.

## 2.4 Closed loop configuration

The closed loop is achieved with the calculation of the new inputs in the next iteration of the position correction movement. The first step is to set the next mask position to define the new focus area for the second image capture.

The input variables are the coordinates of the “next global position” that were calculated after the mask correction and the mask limit being the dimensions of the next mask. The ROI limits are the coordinates of the start and end points that draws a rectangle, that is to say, the ROI where the second pixel mesh will be focused. The output variables are used as

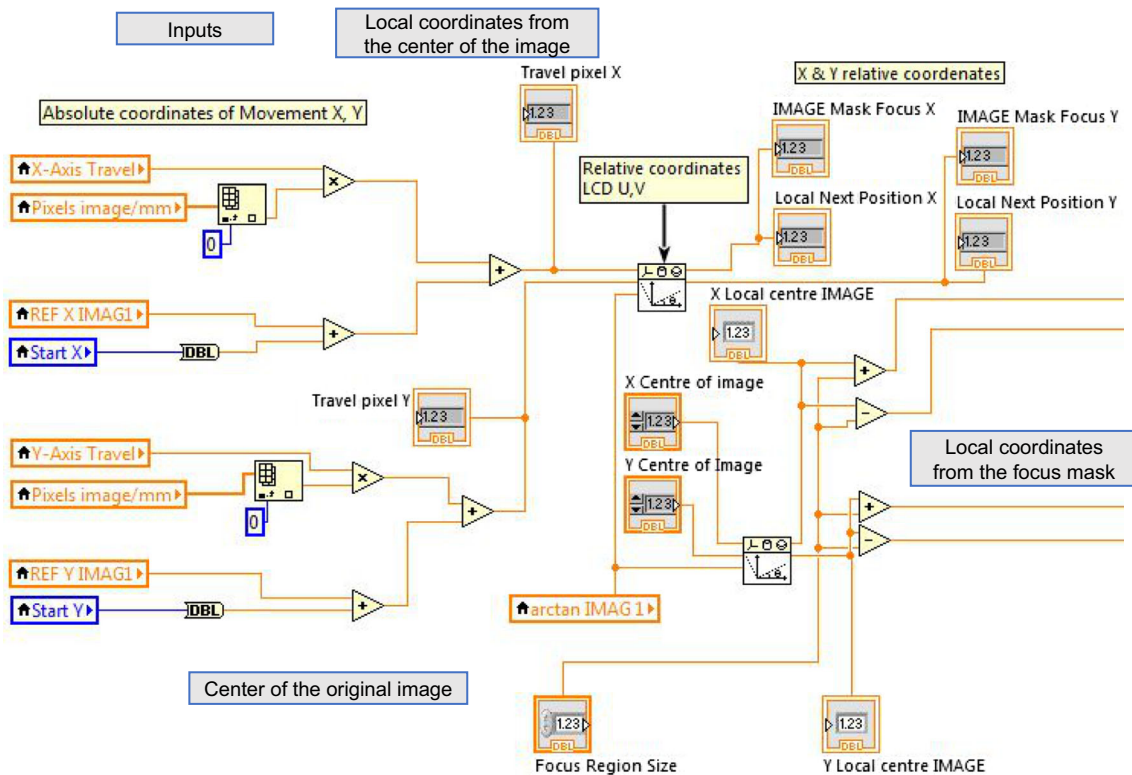


Fig. 10 Mask correction to calculate de target position (step 1)

inputs for the next step while the module reads the ROI coordinates and creates the new mask.

After the analysis of the second image, the average values of both images and the distance between the first and the last image are calculated. The distance between pixels in consecutive images represents the average of all distances between pixels illuminated in two image captures. The values of pixels image/mm, angle of alignment, and distance of the mask as distance between the first and the second position of the ROI mask are considered. Once obtained the average values, the method calculates the real distance, which is the length that the system has physically moved along both axes.

All parameters calculated are focused to determine if the loop continues or stops, which occurs when the objective error condition is achieved. That is, when the error made in the iteration is below the maximum error allowed in the movement introduced as one of the main parameters prior to the start of the movement.

### 2.5 Front panel

The front panel uses as graphical user interface (GUI) includes all the functionalities that the user needs to be able to work with the algorithm developed. Many authors have studied the GUIs proposing basic design principles of the same. Consequently, Hollifield in his book “The high performance HMI handbook” [67] provides useful guidelines to optimize

their efforts and results. Albornoz [68] highlights the importance of having a good GUI design in any application explaining principles and basic rules of design. When building, programming an application should not only be developed thinking of complying with all the functional requirements that the operator requests, but also should be scheduled making it evident at first sight that the application complies with the requested.

It has been determined what information is the most important and most applicable for the developed system to be able to establish the guidelines and design decisions of the graphic interface. The three areas of design principles of the user interface are as highlighted by Pressman [69]: (1) getting users to have control of the interface, (2) reduce the memory load for the user, and (3) make the user interface consistent (stable). Applying these principles, it has been focused to simplify the knowledge necessary to use this software. All unnecessary information and known values have been removed to facilitate control. The values shown are necessary to understand and control the movements of the system in a simple and efficient way. Figure 13 shows the full interface window indicating its two main parts.

The front panel has two different main parts: (1) *work modes*, where the user can select different work modes and other configurations using the available tabs allowing the interaction with the software through actions or data input; (2) *graphic information and control values*, which



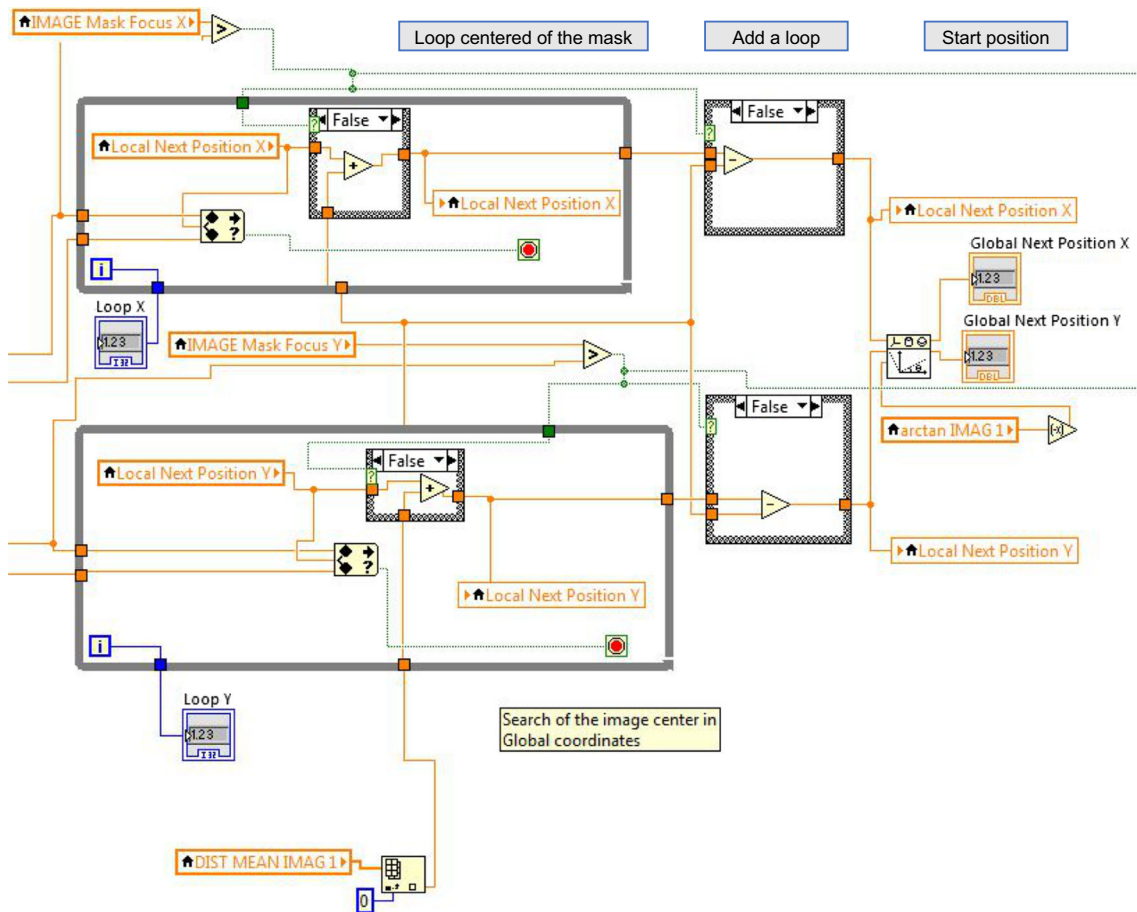
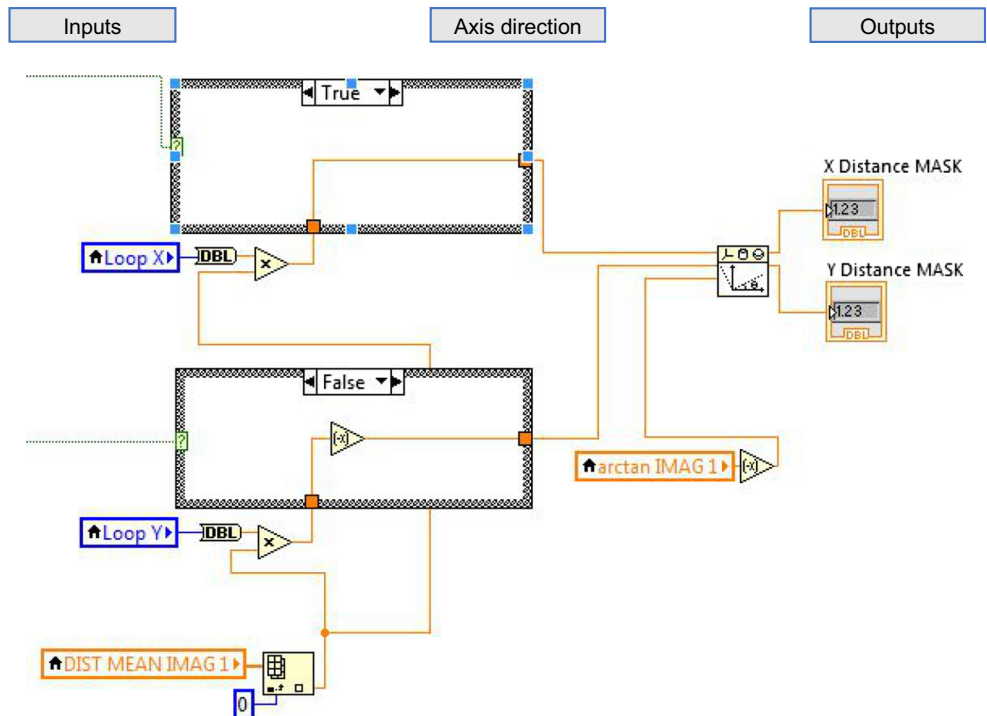
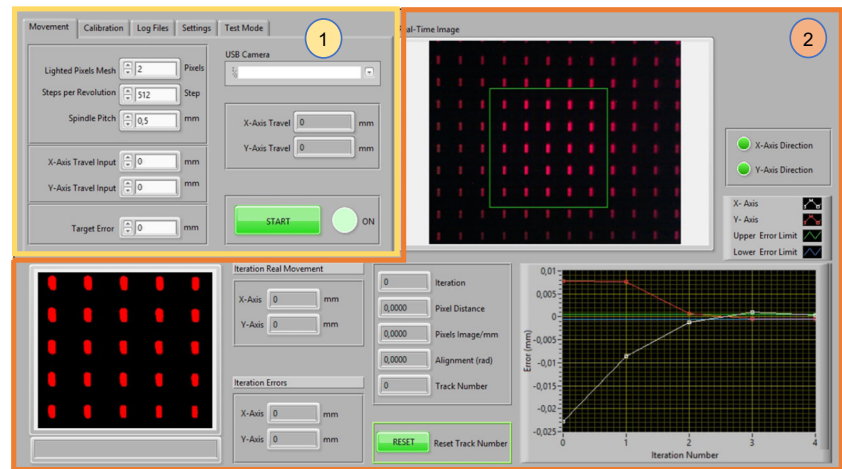


Fig. 11 Mask correction to focus the mask (step 2)

Fig. 12 Mask correction due to mask distance movement (step 3)



**Fig. 13** Front panel. (1) Work modes. (2) Graphic information and control values



displays the information necessary to control the correct functioning of the movement including a real-time position error graph window.

In addition, the software generates an activity log. This registry provides valuable information for the end user and allows verifying the correct work of the demonstrator. There are three types of registration: (1) *Log.txt* where all the positions and corrections of movement are registered; (2) *Image cache*, which captures all the images taken by the demonstrator to obtain the position calculations; and (3) *Graphics cache*, with information about the charts that the software draws for all iterations.

### 3 Experimental results

The described methodology and algorithm has been tested in a two-dimensional control system consisting in an  $X$ - $Y$  movement platform moved using two stepper motors (ST28, 12 V, 280 mA) controlled by a computer vision system and a NI-6001 USB as data acquisition card (Fig. 14).

The demonstrator includes a webcam (Trust © 1280 × 1024 pixel) as camera and an iPhone 5S used to represent the patterns in the LCD screen. In the test was used a 2 × 2 matrix in the ROI. Each stepper motor moves two precision linear guides (IKO © BSR2080 50 mm stroke) each connected to a spindle (M3). The LCD screen of the phone has a definition of 1136 × 640 pixels. The output signals of the NI-USB-6001 are processed by an amplification circuit consisting of two L293 H bridges. For the calculation of the aspect ratio  $\overline{G}$  (11), it was considered a minimum LCD image distance of  $77.93 \times 10^{-3} \mu\text{m}$  and a gap distance of 12.825 LCD pixels/mm providing a value of  $\overline{G} = 729.407 \text{ pixels/mm}$ .

For the validation and analysis of the algorithm described in this article, the results of 2 different tests performed with the described hardware are presented.

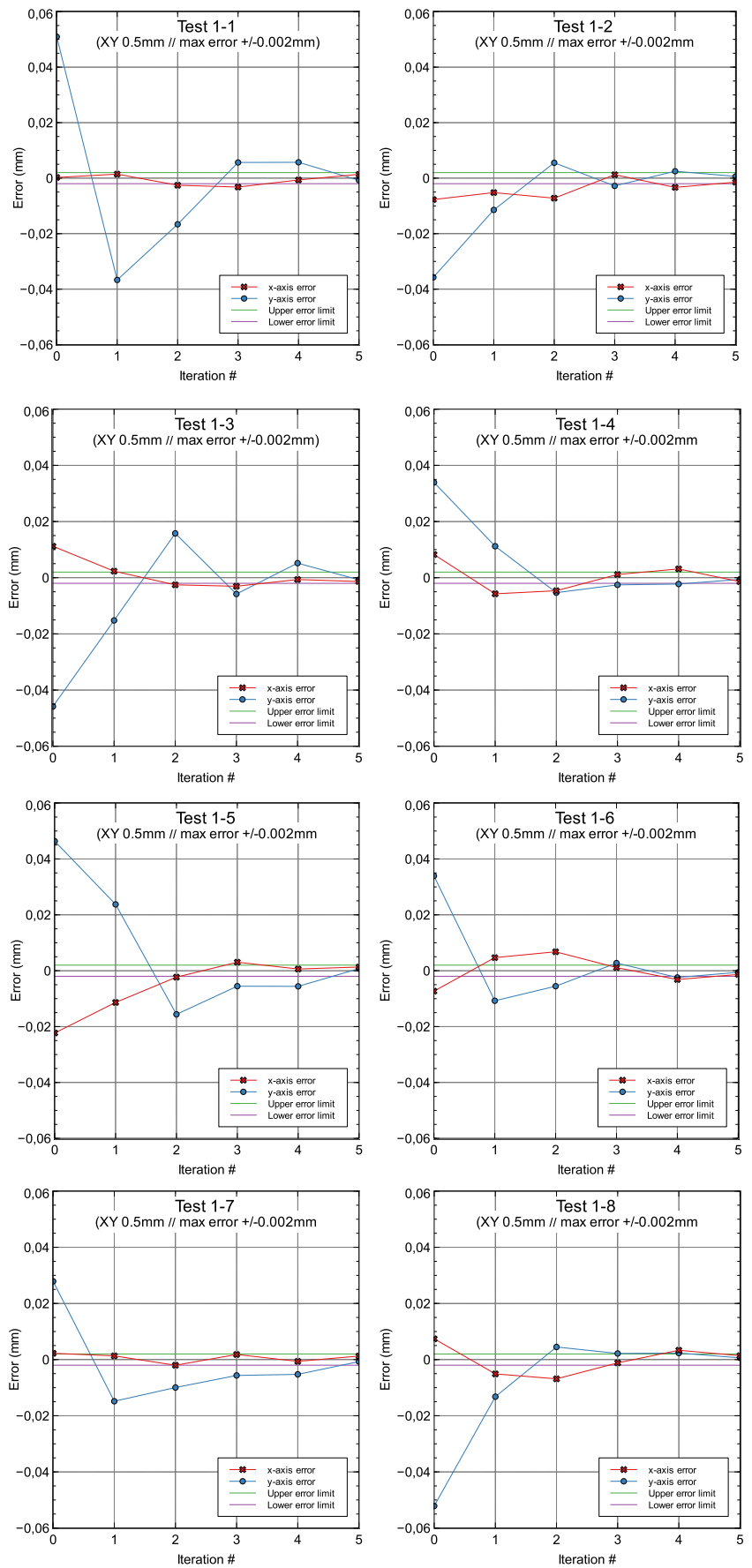
In the first test, a movement of 0.5 mm in both  $x$  and  $y$  axes was requested to the system. Once the system calculates the steps necessary in the stepper motor to move 0.5 mm it activates the movement of the motors.

The position of the  $x$  and  $y$  axes were measured analyzing the pictures taken in every iteration of the close loop. The max error allowed (target error) is set  $\pm 2 \mu\text{m}$ . With these settings, the first movement provides, due to the mechanical design (mechanical adjustments, motor resolution, and electronics), a movement which requires compensation in both axis since the error is greater than the max error allowed. The artificial



**Fig. 14** Micro-machining tool prototype. Two-dimensional control system

Fig. 15 Results of Test 1-1 to 1-8



**Table 2** Test 1-1 to Test 1-4 error data results

Iteration	Error X1-1 ( $\mu\text{m}$ )	Error Y1-1 ( $\mu\text{m}$ )	Error X1-2 ( $\mu\text{m}$ )	Error Y1-2 ( $\mu\text{m}$ )	Error X1-3 ( $\mu\text{m}$ )	Error Y1-3 ( $\mu\text{m}$ )	Error X1-4 ( $\mu\text{m}$ )	Error Y1-4 ( $\mu\text{m}$ )	Max. error ( $\mu\text{m}$ )
0	0.2	50.9	-7.7	-35.7	0.2	-45.9	7.2	33.9	$\pm 2.0$
1	1.5	-36.7	-5.2	-11.4	1.3	-35.2	-4.7	11.2	$\pm 2.0$
2	-2.6	-16.6	-7.2	5.5	-2.5	15.8	-6.6	-5.3	$\pm 2.0$
3	-3.2	5.7	1.2	-2.8	-3.0	-5.2	1.1	-2.5	$\pm 2.0$
4	-0.7	5.7	-3.3	2.5	-0.6	5.5	3.1	-2.2	$\pm 2.0$
5	1.4	-0.8	-1.5	0.6	-1.4	-0.7	-1.4	-0.6	$\pm 2.0$

vision algorithm created in LabVIEW continues analyzing the global error from the previous movement and estimates the next step movement required to achieve the final position until the tolerance (max. error) requested is achieved. The results for 8 trials with similar test configuration, including the sequence of the movements and the algorithm adjustment to reach the target error, are presented in Fig. 15.

The errors made at each iteration are calculated with the above described algorithm and, while they were greater than the maximum error allowed, the next movement will be calculated and consequently the motors move the related axis distance to compensate it. The errors made at each iteration are presented in Tables 2 and 3.

A second test configuration was performed in order to study the stability of the algorithm. The procedure was to make a movement by setting the maximum error at a value of 0.000 mm and detect how the system responds to the long term. The first objective was to verify if the algorithm was able to make a stable and consistent compensation. In addition, it should be convergent or asymptotic with a value very close to zero.

It has been verified after many simulations that the algorithm is valid and stable.

The results of four (Test 2-1 to Test 2-4) of the simulations are presented, running the algorithm 25 iterations. From iterations 4–5, the system was stabilized in a limited error.

Due to the mechanical limitations of the demonstrator used for this validation (mechanical clearances, backlash, precision

of mechanical elements, linear guides, skids), the error cannot continue decreasing. The graphical results are in Fig. 16 and the numeric recap in Table 4.

## 4 Conclusions

In this paper, an alternative algorithm created in LabVIEW to be used as a method of high positioning accuracy has been proposed, described, and tested. It has been developed by treating the images showed in an LCD screen using a webcam fixed to the movement system. The positioning system algorithm has been implemented with a closed loop to be able to perform precise position control. This method could change the paradigm of the actual accurate positioning systems since it presents a new and different approach that requires low-cost hardware compare to the classical positioning systems based in rotary or linear encoders and resolvers.

The algorithm tested in a 2-dimension mobile platform reached the target error ( $\pm 2 \mu\text{m}$ ) after an average of 5 iterations of motion. All trials achieved the requested accuracy.

A second test setup (stress test) provided quick convergence, merging all tests to an asymptotic value of  $\pm 2 \mu\text{m}$  after an average of 5 iterations. The algorithm remains below such error during all remaining iterations proving the consistency and stability.

**Table 3** Test 1-5 to Test 1-8 error data results

Iteration	Error X1-5 ( $\mu\text{m}$ )	Error Y1-5 ( $\mu\text{m}$ )	Error X1-6 ( $\mu\text{m}$ )	Error Y1-6 ( $\mu\text{m}$ )	Error X1-7 ( $\mu\text{m}$ )	Error Y1-7 ( $\mu\text{m}$ )	Error X1-8 ( $\mu\text{m}$ )	Error Y1-8 ( $\mu\text{m}$ )	Max. error ( $\mu\text{m}$ )
0	-22.3	46.4	-7.3	33.9	2.2	27.9	7.4	-52.2	$\pm 2.0$
1	-11.4	23.7	4.7	-10.8	1.3	-14.8	-5.1	-13.2	$\pm 2.0$
2	-2.3	-15.6	6.8	-5.5	-2.0	-10.0	-6.9	4.5	$\pm 2.0$
3	3.0	-5.5	1.1	2.7	1.8	-5.7	-1.2	2.2	$\pm 2.0$
4	0.6	-5.6	-3.2	-2.4	-0.6	-5.3	3.3	2.3	$\pm 2.0$
5	1.3	0.8	-1.3	-0.6	1.3	-0.7	1.4	0.6	$\pm 2.0$

Fig. 16 Results of Test 2-1 to 2-4

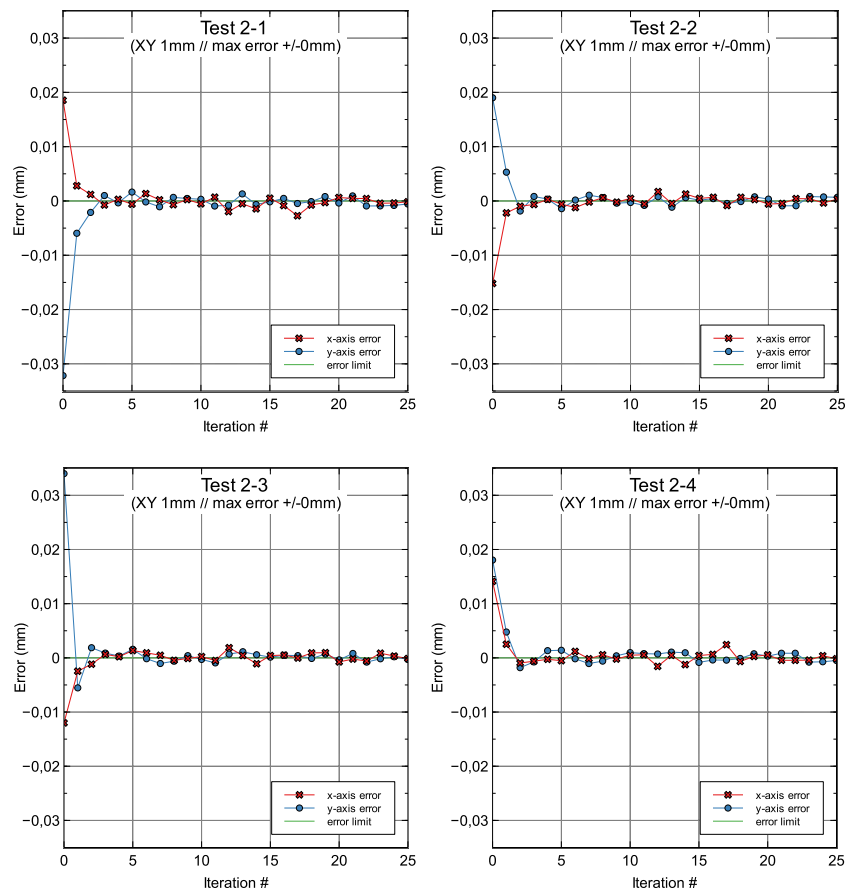


Table 4 Test 2-1 to Test 2-4 error data results

Iteration	Error X2-1 ( $\mu\text{m}$ )	Error Y2-1 ( $\mu\text{m}$ )	Error X2-2 ( $\mu\text{m}$ )	Error Y2-2 ( $\mu\text{m}$ )	Error X2-3 ( $\mu\text{m}$ )	Error Y2-3 ( $\mu\text{m}$ )	Error X2-4 ( $\mu\text{m}$ )	Error Y2-4 ( $\mu\text{m}$ )
0	18.6	-32.2	-15.2	19.0	-12.0	34.0	14.1	18.1
1	2.8	-5.9	-2.2	5.3	-2.5	-5.5	2.5	4.8
2	1.2	-2.1	-1.0	-1.8	-1.2	1.9	-1.0	-1.8
3	-0.7	1.0	-0.6	0.8	0.6	0.9	-0.6	-0.8
4	0.3	-0.3	0.3	0.3	0.2	0.3	-0.3	1.3
5	-0.6	1.6	-0.6	-1.4	1.4	1.6	-0.5	1.4
6	1.4	-0.2	-1.2	0.2	0.9	-0.2	1.2	-0.2
7	0.2	-1.1	-0.2	1.1	0.5	-1.0	-0.2	-1.1
8	-0.7	0.7	0.6	0.6	-0.5	-0.6	0.6	-0.6
9	0.2	0.5	-0.2	-0.4	-0.1	0.4	-0.2	0.4
10	-0.5	0.3	0.5	-0.3	0.2	-0.3	0.5	1.0
11	0.7	-0.9	-0.6	-0.8	-0.5	-0.9	0.6	0.8
12	-2.0	-0.8	1.7	0.8	1.9	0.7	-1.6	0.7
13	-0.5	1.3	-0.5	-1.2	0.4	1.1	0.5	1.1
14	-1.4	-0.6	1.3	0.6	-1.1	0.6	-1.3	0.9
15	0.5	-0.2	0.5	0.1	0.4	0.1	0.4	-0.8
16	-0.8	0.5	0.7	0.4	0.5	0.5	0.6	-0.4
17	-2.7	-0.5	-0.8	-0.5	0.0	0.4	2.4	-0.4
18	-0.7	-0.1	0.7	-0.1	0.9	-0.1	-0.7	-0.1
19	-0.3	0.8	0.3	0.8	1.0	0.7	0.3	0.7
20	0.6	-0.4	-0.6	0.3	-0.7	-0.4	0.6	0.3
21	0.5	0.9	-0.5	-0.9	-0.2	0.8	-0.4	0.9
22	0.4	-0.9	0.4	-0.9	-0.6	-0.8	-0.4	0.9
23	-0.4	-0.9	0.4	0.8	0.9	-0.2	-0.4	-0.8
24	-0.4	-0.8	-0.4	0.7	0.3	0.2	0.4	-0.7
25	-0.2	-0.6	0.3	0.7	-0.1	-0.3	-0.2	-0.5

The limitation of mechanics and electronics in the demonstrator are limited not by the proposed algorithm but by the theoretical accuracy that the system can achieve [70].

The proposed algorithm could be implemented in manufacturing processes providing the required mechanical systems with high accuracy vs investment required.

This algorithm will be enhanced in the future with the implementation of an absolute reference system providing better movement consistency and initial referencing. The compensation of the effect of ambient light in different situations during the process should also be studied and included.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

- Montes CA, Ziegert JC, Wong C, Mears L, Tucker T (2010) 2-D absolute positioning system for real time control applications. In: Proceedings A (ed) Proceedings of the Twenty-fourth Annual Meeting of the American Society for Precision Engineering, Atlanta, GA, USA
- De Francisco-Ortiz O, Sánchez-Reinoso HT, Estrems-Amestoy M (2015) Development of a robust and accurate positioning system in micromachining based on CAMERA and LCD screen. In: Procedia Engineering. Vol 132, pp 895-902
- Cano-García AE, Lázaro JL, Infante A, Fernández P, Pompa-Chacón Y, Espinoza F (2012) Using the standard deviation of a region of interest in an image to estimate camera to emitter distance. *Sensors* 12(5):5687
- Wong CY, Montes C, Mears L, Ziegert J, Ieee (2010) Model-based control to enhance a novel two dimensional positioning system
- Franzi E, Masa P, Heim P (2008) Method of two-dimensional measurement of the position of an object. In: Office EP (ed). Vol EP 1 972 901 A1, EU
- Zhang Z, Han D, Dezert J, Yang Y (2019) A New image registration algorithm based on evidential reasoning. *Sensors* 19(5):1091
- Yusifli E, Yahiaoui R, Qaisar SM et al (2017) FPGA-based HD Camera system for the micropositioning of biomedical micro-objects using a contactless micro-conveyor. *Micromachines* 8(3): 19. <https://doi.org/10.3390/mi8030074>
- Quan CG, Tay CJ, Sun W, He XY (2008) Determination of three-dimensional displacement using two-dimensional digital image correlation. *Appl Opt* 47(4):583-593. <https://doi.org/10.1364/ao.47.000583>
- Xiao S, Tao W, Zhao H (2016) A flexible fringe projection vision system with extended mathematical model for accurate three-dimensional measurement. *Sensors* 16(5):612
- Chen JX, Lin SW, Zhou XL, Tu YL (2017) An on-machine error calibration method for a laser micromachining tool. *Precis Eng* 47: 239-248. <https://doi.org/10.1016/j.precisioneng.2016.08.009>
- Posada-Gómez R, Sandoval-González OO, Sibaja AM, Portillo-Rodríguez O, Alor-Hernández G (2011) Digital image processing using LabVIEW. In: Practical Applications and Solutions Using LabVIEW™ Software. IntechOpen
- Sanguino TDM, Webber PAC (2018) Making image and vision effortless: learning methodology through the quick and easy design of short case studies. *Comput Appl Eng Educ* 26(6):2102-2115. <https://doi.org/10.1002/cae.22003>
- Lamer J, Cymbalak D, Jakab F (2013) Computer vision based object recognition principles in education. 2013 11th Ieee International Conference on Emerging Elearning Technologies and Applications (Iceta 2013): 253-257
- Hornig J-H (2008) Hybrid MATLAB and LabVIEW with neural network to implement a SCADA system of AC servo motor. *Adv Eng Softw* 39(3):149-155
- Dsouza RD, Navin KP, Theodoridis T, Sharma P (2018) Design, fabrication and testing of a 2 DOF compliant flexural microgripper. *Microsyst Technol Micro Nanosyst -Inf Storage Process Syst* 24(9): 3867-3883. <https://doi.org/10.1007/s00542-018-3861-y>
- Mahmoodi M, James LA, Johansen T (2018) Automated advanced image processing for micromodel flow experiments; an application using labVIEW. *J Pet Sci Eng* 167:829-843. <https://doi.org/10.1016/j.petrol.2018.02.031>
- Ijiri T, Mitsushio T, Suzuki Y, Yoshiura T Apparatus for assisting development of program executed in image processing inspection apparatus, displays the screens for setting shape and position about range made into image processing object with respect to processing modules. In: Omron Kk (Omro-C)
- Rout A, Deepak B, Biswal BB, Mahanta GB, Gunji BM (2018) An optimal image processing method for simultaneous detection of weld seam position and weld gap in robotic arc welding. *Int J Manuf Mater Mech Eng* 8(1):37-53. <https://doi.org/10.4018/ijmmme.2018010103>
- Bardak T, Bardak S (2017) Prediction of wood density by using red-green-blue (RGB) color and fuzzy logic techniques. *Politeknik Dergisi* 20(4):979-984. <https://doi.org/10.2339/politeknik.369132>
- Bardak T, Bardak S, Sözen E (2017) Determination of strain distributions of solid wood and plywood in bending test by digital image correlation. *J For Fac* 17(1):354-361
- Yamamoto K, Ishimaru H, Hayashi E (2012) Development of an autonomous-drive personal robot "An environment recognition and the position detecting system that used image processing and an LRS"
- Tayama T, Kurose Y, Nitta T, et al. (2017) Image processing for autonomous positioning of eye surgery robot in micro-cannulation. 3rd Cirp Conference on Biomanufacturing 65: 105-109. <https://doi.org/10.1016/j.procir.2017.04.036>
- Zhou G, Shen B, Yan J (2018) Research on the algorithm for solving the indoor vision positioning model of mobile robot. In: 2018 International Conference on Mathematics, Modelling, Simulation and Algorithms (MMSA 2018). Atlantis Press
- Zhou K, Wang XJ, Wang Z, Wei H, Yin L (2018) Complete initial solutions for iterative pose estimation from planar objects. *Ieee Access* 6:22257-22266. <https://doi.org/10.1109/access.2018.2827565>
- Lyu D, Xia H, Wang C (2018) Research on the effect of image size on real-time performance of robot vision positioning. *EURASIP J Image Video Process* 2018(1):112
- Montijano E, Cristofalo E, Zhou D, Schwager M, Sagüés C (2016) Vision-based distributed formation control without an external positioning system. *IEEE Trans Robot* 32(2):339-351
- Yang S, Jiang R, Wang H, Ge SS (2017) Road constrained monocular visual localization using Gaussian-Gaussian cloud model. *IEEE Trans Intell Transp Syst* 18(12):3449-3456
- Guo D, Wang H, Leang KK (2018) Nonlinear vision-based observer for visual servo control of an aerial robot in global positioning system denied environments. *J Mech Robot* 10(6):061018

29. Vivacqua RPD, Bertozzi M, Cerri P, Martins FN, Vassallo RF (2018) Self-localization based on visual lane marking maps: an accurate low-cost approach for autonomous driving. *IEEE Trans Intell Transp Syst* 19(2):582–597
30. Fang J, Wang Z, Zhang H, Zong W (2018) Self-localization of intelligent vehicles based on environmental contours. In: 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), pp 624–629
31. Islam KT, Wijewickrema S, Pervez M, O'Leary S (2018) Road trail classification using color images for autonomous vehicle navigation. In: 2018 Digital Image Computing: Techniques and Applications (DICTA), pp 1–5
32. Samarasekera S, Oskiper T, Kumar R, Sizintsev M, Branzoi V (2016) Augmented reality vision system for tracking and geolocating objects of interest. In: Google Patents
33. Suenaga H, Tran HH, Liao H et al (2015) Vision-based markerless registration using stereo vision and an augmented reality surgical navigation system: a pilot study. *BMC Med Imaging* 15(1):51
34. Rajeev S, Wan Q, Yau K, Panetta K, Agaian SS (2019) Augmented reality-based vision-aid indoor navigation system in GPS denied environment. In: *Proc SPIE 10993, Mobile Multimedia/Image Processing, Security, and Applications*
35. Sim R, Little JJ (2009) Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters. *Image Vis Comput* 27(1-2):167–177
36. Valencia-García R, Martínez-Béjar R, Gasparetto A (2005) An intelligent framework for simulating robot-assisted surgical operations. *Expert Syst Appl* 28(3):425–433
37. Pichler A, Akkaladevi SC, Ikeda M et al (2017) Towards shared autonomy for robotic tasks in manufacturing. *Proc Manuf* 11:72–82
38. Patterson T, McClean S, Morrow P, Parr G, Luo C (2014) Timely autonomous identification of UAV safe landing zones. *Image Vis Comput* 32(9):568–578
39. González D, Pérez J, Milánés V (2017) Parametric-based path generation for automated vehicles at roundabouts. *Expert Syst Appl* 71: 332–341
40. Sanchez-Lopez JL, Pestana J, De La Puente P, Campoy P (2015) A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-UAV systems: Simulation and experimentation. *J Intell Robot Syst*: 1-19
41. Olivares-Mendez MA, Kannan S, Voos H (2015) Vision based fuzzy control autonomous landing with UAVs: from V-REP to real experiments. In, pp 14-21
42. Romero-Ramírez FJ, Muñoz-Salinas R, Medina-Carnicer R (2018) Speeded up detection of squared fiducial markers. *Image Vis Comput* 76:38–47
43. Germanese D, Leone GR, Moroni D, Pascali MA, Tampucci M (2018) Long-term monitoring of crack patterns in historic structures using UAVs and planar markers: a preliminary study. *J Imag* 4(8)
44. An GH, Lee S, Seo M-W, Yun K, Cheong W-S, Kang S-J (2018) Charuco board-based omnidirectional camera calibration method. *Electronics* 7(12)
45. Pflugi S, Vasireddy R, Lerch T, et al. (2017) Augmented marker tracking for peri-acetabular osteotomy surgery. In, pp 937-941
46. Lima JP, Roberto R, Simões F et al (2017) Markerless tracking system for augmented reality in the automotive industry. *Expert Syst Appl* 82:100–114
47. Chen P, Peng Z, Li D, Yang L (2016) An improved augmented reality system based on AndAR. *J Vis Commun Image Represent* 37:63–69
48. Khattak S, Cowan B, Chepurna I, Hogue A (2014) A real-time reconstructed 3D environment augmented with virtual objects rendered with correct occlusion. *Proc GEM, Toronto, ON, Canada*: 1-8
49. Kalkman CJ (1995) LABVIEW - a software system for data-acquisition, data-analysis, and instrument control. *J Clin Monit* 11(1):51–58. <https://doi.org/10.1007/bf01627421>
50. Leviton DB (2004) Method and apparatus for two-dimensional absolute optical encoding. In: U.S. Patent 6,765,195 B1
51. Chu D, Trujillo H, Whitney E, Young J (2006) Rapid high resolution absolute 2-D encoding by low resolution digital imaging of a mathematically generated 3-Tone target. In: ASPE, Santa Clara, CA, USA
52. Cheng HD, Jiang XH, Sun Y, Wang JL (2001) Color image segmentation: advances and prospects. *Pattern Recogn* 34(12):2259–2281. [https://doi.org/10.1016/s0031-3203\(00\)00149-7](https://doi.org/10.1016/s0031-3203(00)00149-7)
53. de Francisco Ortiz Ó (2020) Análisis e implementación de un método de visión artificial para microposicionado basado en un sistema cámara-pantalla. In: Department of Materials and Manufacturing Engineering. Technical University of Cartagena, PQDT - Global, p 274
54. Shik JW, Lee S-M, Kim K-S (2002) Development of real-time image processing algorithm on the positions of multi-object in an image plane. *J Korean Soc Nondestruct Test* 22(5):523–532
55. Grois D, Hadar O (2013) Advances in region-of-interest video and image processing. In: *Multimedia Networking and Coding*. Igi Global, Hersey, pp 76-123
56. Shah B, Panchal G, Ieee (2017) Comparative analysis on different region of interest (RoI) extraction mechanisms for fingerprint. *Proceedings of the International Conference on Intelligent Sustainable Systems (Iciss 2017)*: 690-694
57. Zhou HB, Zhang YD, Yu ZH (2015) Image classification based on region of interest detection. In: 9th International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR) - Pattern Recognition and Computer Vision. Vol 9813. Spie-Int Soc Optical Engineering, Enshi, PEOPLES R CHINA
58. Salah MM, Napoleon SA, El-Rabaie EM, Abd El-Samie FE, Abd Elnaby MM (2019) Sensitivity analysis of a class of iris localization algorithms to blurring effect. *Wirel Pers Commun* 104(1):269–286. <https://doi.org/10.1007/s11277-018-6019-4>
59. Tashlinskii AG, Smimov PV (2015) Compensation algorithm for the blurring effect of a moving object using a sequence of frames. *Pattern Recognit Image Anal* 25(4):722–722. <https://doi.org/10.1134/s1054661815040240>
60. Cordella LP, Duff MJB, Levialdi S (1978) Analysis of computational cost in image-processing - case-study. *IEEE Trans Comput* 27(10):904–910
61. Cattaneo CA, Larcher LI, Ruggeri AI, Herrera A, Biasoni E (2011) Métodos de umbralización de imágenes digitales basados en entropía de Shannon y otros. *Mach Learn* 20(3):2785–2805
62. Li CH, Lee CK (1993) Minimum cross entropy thresholding. *Pattern Recogn* 26(4):617–625. [https://doi.org/10.1016/0031-3203\(93\)90115-d](https://doi.org/10.1016/0031-3203(93)90115-d)
63. Pun T (1981) Entropic thresholding, a new approach. *Comput Graphics Image Process* 16(3):210–239. [https://doi.org/10.1016/0146-664x\(81\)90038-1](https://doi.org/10.1016/0146-664x(81)90038-1)
64. Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using the entropy of the histogram. *Comput Vis Graph Image Process* 29(3):273–285. [https://doi.org/10.1016/0734-189x\(85\)90125-2](https://doi.org/10.1016/0734-189x(85)90125-2)
65. Kittler J, Illingworth J (1986) Minimum error thresholding. *Pattern Recogn* 19(1):41–47. [https://doi.org/10.1016/0031-3203\(86\)90030-0](https://doi.org/10.1016/0031-3203(86)90030-0)
66. Instruments N (2016) NI vision 2016 for LabVIEW Manual. In
67. Hollifield B, Habibi E, Nimmo I, Oliver D (2008) The high performance HMI handbook: a comprehensive guide to designing,

- implementing and maintaining effective HMIs for industrial plant operations. *Plant Automation Services*
68. Albornoz MC (2014) Diseño de interfaz gráfica de usuario. In: XVI Workshop de Investigadores en Ciencias de la Computación, pp 540-544
  69. Pressman RS (2005) *Software engineering: a practitioner's approach*. Edition 8 (16-03-2014) edn. Palgrave Macmillan
  70. de Francisco Ortiz Ó, Sánchez Reinoso HT, Estrems Amestoy M (2016) Accuracy and repeatability estimation in micromachining using an enhanced artificial vision algorithm for controlling the screen LED positioning on a LCD screen. In: Euspen's 16th International Conference & Exhibition Proceedings, Nottingham, UK, pp 236-264

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.