




Article

Enhanced Positioning Algorithm Using a Single Image in an LCD-Camera System by Mesh Elements' Recalculation and Angle Error Orientation

Óscar de Francisco Ortiz ^{1,*}, Manuel Estrems Amestoy ², Horacio T. Sánchez Reinoso ²
and Julio Carrero-Blanco Martínez-Hombre ²

¹ Department of Engineering and Applied Technologies, University Center of Defense, San Javier Air Force Base, MDE-UPCT, 30720 Santiago de la Ribera, Spain

² Mechanics, Materials and iManufacturing Engineering department, Technical University of Cartagena, 30202 Cartagena, Spain; manuel.estrems@upct.es (M.E.A.); horacio.sanchez@upct.es (H.T.S.R.); julio.carrero@upct.es (J.C.-B.M.-H.)

* Correspondence: oscar.defrancisco@ cud.upct.es; Tel.: +34-968-189918

Received: 31 October 2019; Accepted: 11 December 2019; Published: 16 December 2019



Abstract: In this article, we present a method to position the tool in a micromachine system based on a camera-LCD screen positioning system that also provides information about angular deviations of the tool axis during its running. Both position and angular deviations are obtained by reducing a matrix of LEDs in the image to a single rectangle in the conical perspective that is treated by a photogrammetry method. This method computes the coordinates and orientation of the camera with respect to the fixed screen coordinate system. The used image consists of 5×5 lit LEDs, which are analyzed by the algorithm to determine a rectangle with known dimensions. The coordinates of the vertices of the rectangle in space are obtained by an inverse perspective computation from the image. The method presents a good approximation of the central point of the rectangle and provides the inclination of the workpiece with respect to the LCD screen reference system of coordinates. A test of the method is designed with the assistance of a Coordinate Measurement Machine (CMM) to check the accuracy of the positioning method. The performed test delivers a good accuracy in the position measurement of the designed method. A high dispersion in the angular deviation is detected, although the orientation of the inclination is appropriate in almost every case. This is due to the small values of the angles that makes the trigonometric function approximations very erratic. This method is a good starting point for the compensation of angular deviation in vision based micromachine tools, which is the principal source of errors in these operations and represents the main volume in the cost of machine elements' parts.

Keywords: image processing; position control; accuracy; micromachines; position compensation; inverse conical perspective; micromanufacturing; manufacturing systems; mechatronics

1. Introduction

Positioning systems are increasingly present in all industrial processes. Furthermore, technology requires progressively more precise systems capable of positioning rapidly and robustly. The cost of those is one of the key factors to integrate high precision systems.

Thanks to the advances in screen and camera technology, positioning algorithms that analyze a pattern shown in a photographic image have been developed [1,2]. More recently, camera-screen positioning systems with dedicated artificial vision algorithms [3–5] have provided high precision at a very interesting cost compared to other positioning technologies such as encoders or resolvers.

Vision positioning systems are increasingly common in process automation [6–10], autonomous driving [11–15], or augmented reality assistants [16–20]. Indeed, this is one of the most promising elements in the Industry 4.0 revolution. However, the current positioning systems used in the machine tool industry based on high precision encoders and sensors are limited by their cost. Therefore, machine tools used for micro-manufacturing have very high prices and require large floor space. Due to this, in micro-manufacturing, the methods that use vision can be competitive by including high performance commercial elements and reducing space such as cameras and mobile phones' LCD screens. In addition, such devices are increasing in definition and resolution, providing vision with much better accuracy.

The methodology used in this article to calculate the position and orientation of the camera in relation to the screen is based on pose determination [21–24], which is used to estimate the position and orientation of one calibrated camera. Several similar methods for calculating the position and orientation of a camera in space using a single image have been described and presented [22,25,26]. Nevertheless, pose estimation and marker detection are widely used tasks for many other technological applications such as autonomous robots [27–29], unmanned vehicles [30–37], and virtual assistants [38–41], among others.

Consequently, this article presents an enhanced method of recalculating the center of the image used by the positioning algorithm in an LCD-camera system, similar to that developed by de Francisco [4] and improved in subsequent studies [42], but being completely different from such previous studies regarding the procedure to calculate the positioning of the part with respect to the reference system of the screen. In previous works, the positioning was obtained through the global center of gravity of the 25 selected LEDs in the image. In this work, the position of the piece is calculated by previously determining an equivalent square obtained by means of regressions of the different lines that form the grid of the 25 LEDs.

In addition, this manuscript also presents the calculation and correction of the orientation angle, which, although very small, always influences the precision positioning due to the large distance between the location of the cutter and the screen. The new method is based on the calculation of the equivalent quadrangle that allows not only the positioning of the center of the image, but also the inclination. The method uses the treatment of an image to obtain the pixel coordinates of a 5×5 dot matrix that serves to locate the focus and orientation of the camera, where the error is due to the distance between the focus and the screen and can be assumed as sine error.

2. Materials and Methods

2.1. Experimental Setup and Measurements

The experimental study was applied to a two-dimensional control system (X and Y). Figure 1 shows the model of the Micromachine Tool (MMT) demonstrator developed for this research. Two stepper motors (ST28, 12, 280 mA) controlled and moved two precision guides (IKO BSR2080 50 mm stroke), which were connected to a M3 ball screw/nut. The LCD screen used provided a 1136×640 pixel resolution, 326 ppi, and 0.078 mm dot pitch. The screen size was 88.5×49.9 mm. Both stepper motors were controlled by the digital output signals provided by an NI 6001-USB data acquisition card connected to the USB port of a laptop computer. The output signal of the acquisition card was treated by a pre-amplification power station composed of two L293 H-bridges. The control was programmed in LabVIEW. It received the image captured by the camera and processed it according to an image enhancing process. It consisted of an image mask application with color plane extraction, fuzzy pixel removal, small object removal, and particle analysis of the mass center of each evaluated pixel. Once it was processed using the developed artificial vision algorithm, it provided the positioning feedback signals needed to move the X and Y axes.

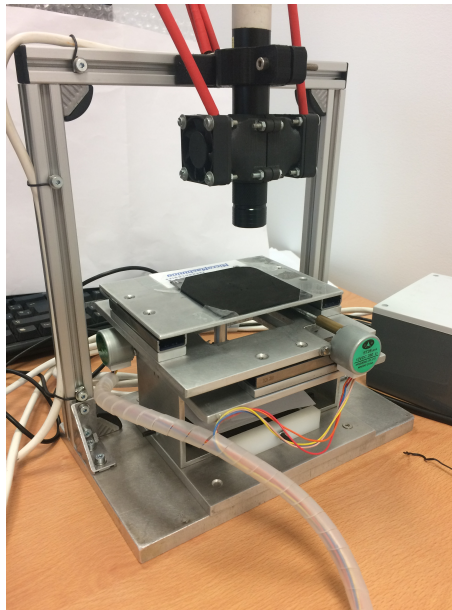


Figure 1. Model of the micromachine tool demonstrator used during the experimental test.

The images were taken by the camera included in the MMT, a Model MITSUBISHI 1.3M digital camera with a resolution of 1280×1024 pixels (1.3 MPixels). To analyze the position, a Coordinate Measuring Machine (CMM) Pioneer DEA 03.10.06 with measuring strokes $600 \times 1000 \times 600$ mm was used (Figure 2). The maximum permissible error of the DEA in the measurements was $2.8 + 4.0 L/1000 \mu\text{m}$. The software used for the measurements was PC-DMIS.

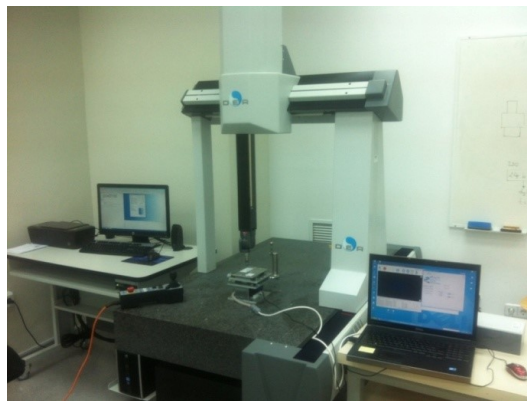


Figure 2. Setup used during the experimental test for the measurement with the Micromachine Tool (MMT) and the Coordinate Measuring Machine (CMM).

Several tests were performed over a 2×2 gap pattern using the camera-LCD algorithm. The simulation consisted of testing a 5 mm X axis movement using 10 steps of 0.5 mm. Each travel was repeated 3 times in both the forward and backward direction, according to the VDI/DGQ 3441 standard: Statistical Testing of the Operational and Positional Accuracy of Machine Tools - Basis.

2.2. Image Acquisition

Image acquisition was done using a procedure developed by the authors in VBA similar to that performed by software such as ImageJ© in its tool “Analyze Particles ...”.

The image may not be focused, although many webcams have autofocus mechanisms that make the focal length variable. In our case, it was unimportant because what matters was the bulk and its center of gravity. It should also be noted that if the extraction was from the complete image, the image usually contained the spherical errors of the lenses that focused the image onto the sensor.

In our case, to speed up the process and calculations, only the central area from the BMP image file that included all 25 LEDs was extracted. Only the red layer was analyzed because it was proven to be the most efficient and the only one used to generate the image. Given the size of the LEDs, an image size of 600×600 was sufficient to ensure the presence of at least 25 LEDs in the image.

3. Obtaining the Equivalent Quadrilateral

Once the 25 coordinates of the centers of the LEDs were obtained, as seen in Figure 3, these data had to be statistically treated to obtain four vertices of a quadrilateral that collected information about the coordinates of the 25 points. With this quadrilateral and knowing the real side dimensions given by the size of the pixels, the position and orientation of the camera with respect to this square were obtained.

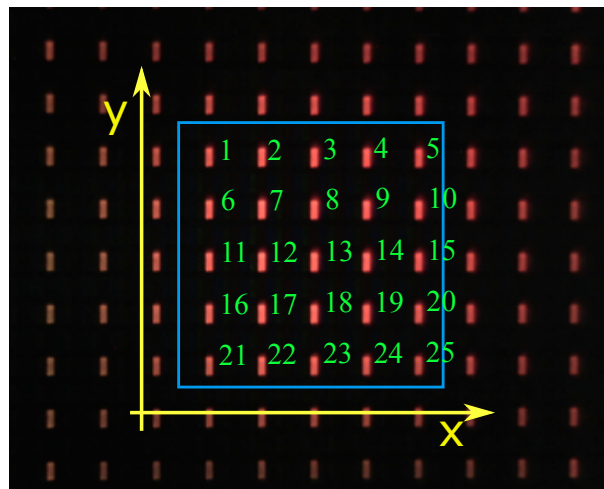


Figure 3. The 5×5 mesh captured by the camera with numbered elements.

3.1. Regression of Lines

From the analysis of the 5×5 grid, different horizontal lines could be segregated, rearranging the table of coordinates by values in y , obtaining 5 groups of 5 values corresponding to the horizontal lines. Reordering by the values in x , the vertical lines were obtained in the same way. The 5 horizontal lines must be translated into 2 lines, the same with the vertical lines, so that the intersection of the four lines gave rise to the 4 vertices of the quadrilateral that represented a square in conical perspective. The two vanishing points were obtained by the intersection of opposite sides.

Figure 4 shows the regression lines, vertical and horizontal, that represented the different groups of points. The slope and interception terms of the lines followed a tendency that could be anyway also found as shown in Figure 5. These tendencies allowed the calculation of the different slopes in the extreme lines of the rectangle that represented adequately the 25 points, as the border of a chessboard included the dimension and position of the interior squares. The correlated lines and the rectangle used to determine the position and inclination of the axis of the camera are represented in Figure 6.

Since the angles of the slopes had very small variation, the line equations had the form $y = m_i x + n_i$. The intersection of a horizontal line with another vertical line is given by Equation (1):

$$x = \frac{n_j + m_j n_i}{1 - m_i m_j} \quad (1)$$

where subindex i corresponds to horizontal lines, while subindex j corresponds to vertical lines.

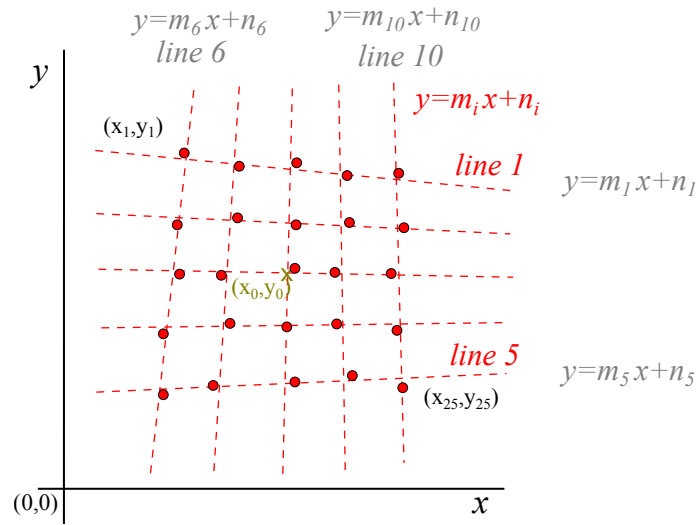


Figure 4. Regression lines in the 5×5 elements used in the image analysis.

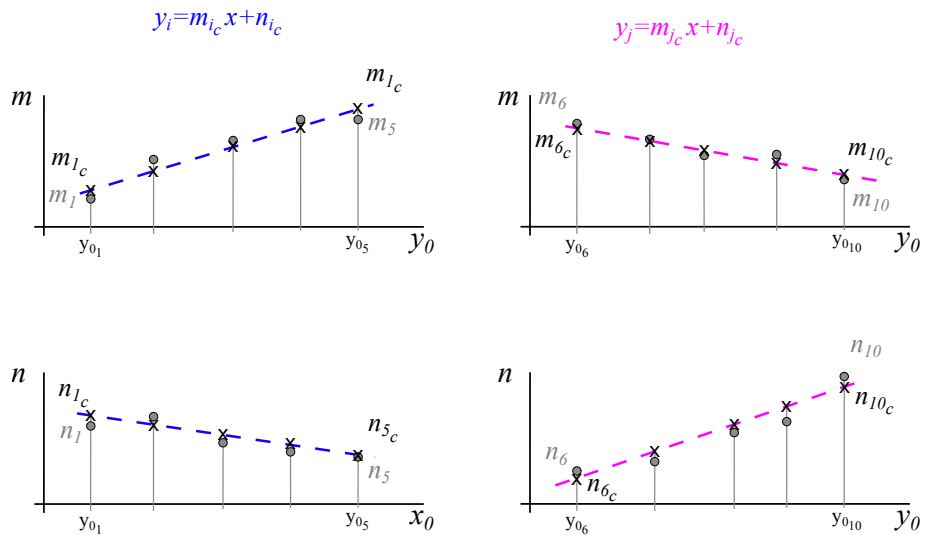


Figure 5. Regression lines (compensation) to optimize the position of the lines.

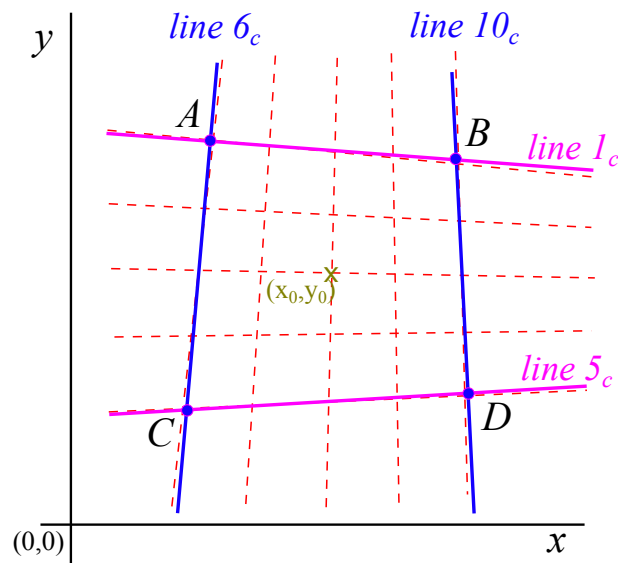


Figure 6. Final distribution of the rectangle used for the position and angle correction.

The steps to obtain the two horizontal lines were the following:

1. Sort by coordinate the data of the grid table obtained.
2. Separate this into groups of 5 points as they belong to the same line by similarity in coordinate y .
3. Perform regression of the five groups obtaining the equations of the five lines $y = m_i x + n_i$, $i = 1 \dots 5$.
4. In a similar way, the data were sorted by x coordinates, then separated into 5 groups of 5 points, and the regression was performed obtaining the equations $y = m_j x + n_j$, $j = 6 \dots 10$.
5. Obtain intersection points (x_i, y_i) from the central vertical line $y = m_8 y + n_8$ with each of the horizontal lines.
6. Accomplish the regression of the slopes of the horizontal lines m_i based on the vertical intersection coordinates y_i ; thereby, the slope was obtained based on the vertical intersection.
7. In such a manner, we proceeded to select the slopes and the points through which the two horizontal lines indicated would be selected. The points of the extreme horizontal lines 1 and 5 were chosen. The two slopes of these two lines were calculated by means of the regression of the 5 slopes. The line was forced to pass through the intersection points of these extreme points, with the following remaining equations of the lines (Equations (2) and (3)):

$$y = m(y_1)x + (y_1 - m(y_1)x_1) \quad (2)$$

$$y = m(y_5)x + (y_5 - m(y_5)x_5) \quad (3)$$

A similar method was used for the calculation of the two vertical lines.

The intersection of the two almost parallel lines provided the 4 extreme vertices A , B , C , and D (Figure 6) that were introduced to the program of the inverse conical perspective to obtain the position and orientation of the camera in relation to the fixed coordinates located and oriented with the square that represented the grid of departure. The position and orientation of the contact point with respect to the screen reference system were obtained using an improvement of the method developed by Haralik for the rectangle reconstruction [22].

3.2. Example of the Calculation of Vertices

To obtain the straight lines, the slopes of the linear regression lines through the data point in the x and y arrays were calculated. In addition, the point at which a line intersected the y axis by using existing x and y values was also calculated for each vertex. The interception point was based on a best fit regression line drawn through the known x and y values, using an internal algorithm for the least squares regression procedure.

The starting point was the table of the centers of each of the zones sorted by the number of pixels comprising the area called mass (Table 1) as the number of pixels that included each zone.

Next, they were sorted by the y coordinates and were classified into groups that corresponded to the horizontal coordinates (Table 2).

As a result, the 5 horizontal lines were obtained $y = m_i x + n_i$, $j = 1 \dots 5$ (Table 3).

In the same way, we proceeded to obtain the vertical lines $y = m_j + n_j$, $j = 6 \dots 10$ (Table 4).

It is noted in Tables 3 and 4 that all coefficients m and n had a tendency that could be also the object of a regression. This indicated that the lines were parallel as they had a vanishing point, and the plane that contained all the lines was not perpendicular to the focal line of the camera.

To find the two horizontal lines that represented the 5 lines, we proceeded to find the intersection points of the vertical center line with the 5 horizontal lines, obtaining the intersection points (x_i, y_i) . The slope m_i was correlated with the vertical coordinates y_i , obtaining the two slopes of the representative lines as lines that had a slope $m(y_1)$ and $m(y_5)$ and passing them through Points 1 and 5, respectively (Table 5).

Table 1. Table of the center of gravity example for Image 1.

Element #	x	y	Mass
1	112.591	195.242	1024
2	113.366	82.265	1015
3	226.495	196.041	990
4	112.105	309.020	982
5	227.189	83.003	968
6	226.096	309.625	961
7	340.272	197.033	947
8	340.892	83.472	943
9	112.080	423.237	938
10	567.863	198.706	925
11	453.996	197.528	917
12	454.825	84.746	897
13	339.734	310.306	890
14	568.349	86.025	873
15	225.773	423.599	872
16	453.313	311.012	857
17	567.139	311.749	854
18	111.869	536.665	833
19	339.310	423.985	813
20	566.355	425.027	786
21	225.561	537.419	776
22	452.760	424.705	757
23	339.075	537.371	717
24	452.296	537.806	682
25	565.838	538.361	673

Table 2. Groups of points corresponding to horizontal lines.

x	y
565.838	538.361
452.296	537.806
225.561	537.419
339.075	537.371
111.869	536.665
566.355	425.027
452.760	424.705
339.310	423.985
225.773	423.599
112.080	423.237
567.139	311.749
453.313	311.012
339.734	310.306
226.096	309.625
112.105	309.020
567.863	198.706
453.996	197.528
340.272	197.033
226.495	196.041
112.591	195.242
568.349	86.025
454.825	84.746
340.892	83.472
227.189	83.003
113.366	82.265

Once having performed the regression of m with respect to y_i , a function of the slope that varied regularly across the different heights ($m = -1.136 \times 10_i^{-3} + 9.331 \times 10^{-3}$) was obtained. As a result, the equations of the horizontal lines that passed through Points 1 and 5 could be calculated.

$$y = -6108 \times 10^{-3}x + 540.807 \quad (4)$$

$$y = -9533 \times 10^{-4}x + 83.981 \quad (5)$$

Table 3. Coefficients of horizontal lines.

<i>m</i>	<i>n</i>
3.331×10^{-3}	536.395
4.127×10^{-3}	422.710
6.018×10^{-3}	308.298
7.393×10^{-3}	194.394
8.142×10^{-3}	81.126

Table 4. Coefficients vertical lines.

<i>m</i>	<i>n</i>
-5.774×10^{-3}	568.910
-5.553×10^{-3}	455.166
-4.050×10^{-3}	341.114
-3.500×10^{-3}	227.307
-3.080×10^{-3}	113.355

Table 5. Intersection points of the central vertical line with horizontal lines.

Point	<i>x_i</i>	<i>y_i</i>
1	338.937	537.525
2	339.396	424.111
3	339.857	310.344
4	340.316	196.911
5	340.774	83.901

In the same way, we proceeded to obtain the 2 representative vertical lines:

$$x = -6444 \times 10^{-3}y + 570.765 \quad (6)$$

$$x = -1277 \times 10^{-3}y + 112.547 \quad (7)$$

The intersection of the opposite lines of this square provided the coordinates of the four vertices that represented the 25 LEDs (Table 6).

Table 6. Quadrilateral points to deal with the reverse perspective program.

Point	<i>x</i>	<i>y</i>
1	111.857	540.124
2	567.302	537.343
3	570.227	83.437
4	112.439	83.874

The vertices represented in Table 6 were treated using the photogrammetry method of the reconstruction of a rectangle described in Estrems [24], and the coordinates of the camera with respect to the square coordinate system were obtained, as well as the cosine direction of the focal line in this system.

$$a = \sqrt{d_f^2 - (d_f \cdot \cos b)^2} = d_f \sqrt{1 - \cos^2 b} \quad (8)$$

$$a_2 - a_1 = d_f \left(\sqrt{1 - \cos^2 b_2} - \sqrt{1 - \cos^2 b_1} \right) \quad (9)$$

In Figure 7, the Abbe error *a* is represented and calculated by the focal distance *d_f* and the cosine in the *z* direction *cos b*. The Abbe error is calculated in Equation (8), and the step error due to the variation of angle *b* during the movement is compensated at each point according to Equation (9).

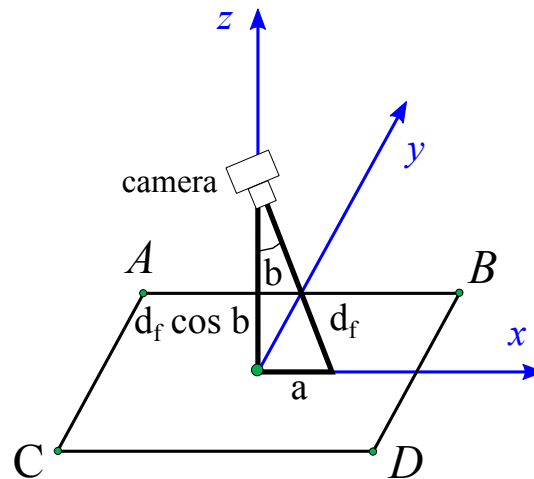


Figure 7. Position error in the vision system due to camera inclination for axis direction movement.

4. Experimental Results and Discussion

The data obtained in the experimental test are described in Tables 7–12, where CMM is the distance measure by the Coordinate Measuring Machine in the movements done by the MMT in every step during the test; Image is the distance moved in every step analyzed by the vision system; Error Image is the error provided by the vision system in every step after a comparison with the distance moved provided by the CMM; Compensation is the distance compensated due to angle error calculated in every step; Image compensated is the distance measured by the CMM after the application of the compensation calculated; Error compensation is the error after the compensation is applied; and Coincidence provides information about the coincidence in the orientation calculated for the angle compensation (YES means orientation coincidence, and NO means the angle orientation calculated is opposite the compensation required to minimize the error).

Table 7. Data for Run #1 forward with a mesh of 5 × 5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
501	−501.897	−0.897	0.603	−501.294	−0.294	YES
1000	−995.031	4969	−4.432	−999.463	0.537	YES
1487	−1483.389	3611	−0.714	−1484.102	2.898	YES
1998	−1984.546	13.454	−2.164	−1986.710	11.290	YES
2489	−2489.081	−0.081	0.471	−2488.610	0.390	YES
2988	−2986.138	1.862	1.062	−2985.076	2.924	NO
3490	−3487.323	2.677	0.802	−3486.521	3.479	NO

Table 8. Data for Run #1 backward with a mesh of 5 × 5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
496	−496.099	−0.099	2.394	−493.705	2.295	YES
992	−992.400	−0.400	−1.744	−994.144	−2.144	NO
1482	−1493.937	−11.937	0.505	−1493.432	−11.432	YES
1991	−1993.130	−2.130	0.632	−1992.498	−1.498	YES
2494	−2495.707	−1.707	0.912	−2494.795	−0.795	YES
2993	−2994.327	−1.327	2.214	−2992.113	0.887	YES
3494	−3493.969	0.031	−1.950	−3495.918	−1.918	YES

Table 9. Data for Run #2 forward with a mesh of 5×5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
490	−498.217	−8.217	1.671	−496.545	−6.545	YES
985	−989.868	−4.868	2.260	−987.608	−2.608	YES
1481	−1489.347	−8.347	−0.803	−1490.150	−9.150	NO
1976	−1993.442	−17.442	1.153	−1992.290	−16.290	YES
2493	−2499.212	−6.212	0.604	−2498.608	−5.608	YES
2991	−2996.982	−5.982	2.728	−2994.254	−3.254	YES
3485	−3494.890	−9.890	1.978	−3492.912	−7.912	YES

Table 10. Data for Run #2 backward with a mesh of 5×5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
495	−498.695	−3.695	1.631	−497.065	−2.065	YES
991	−993.169	−2.169	2.490	−990.679	0.321	YES
1486	−1497.023	−11.023	4.948	−1492.075	−6.075	YES
2003	−1994.767	8.233	−0.105	−1994.872	8.128	YES
2501	−2493.756	7.244	−0.142	−2493.898	7.102	YES
2995	−2995.898	−0.898	1.446	−2994.452	0.548	YES
3494	−3494.915	−0.915	2.048	−3492.867	1.133	YES

Table 11. Data for Run #3 forward with a mesh of 5×5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
496	−500.617	−4.617	1.720	−498.897	−2.897	YES
994	−998.341	−4.341	1.814	−996.527	−2.527	YES
1497	−1496.865	0.135	−0.990	−1497.855	−0.855	YES
1997	−1997.961	−0.961	−0.007	−1997.968	−0.968	NO
2500	−2502.260	−2.260	0.168	−2502.092	−2.092	YES
2996	−3007.135	−11.135	2.567	−3004.567	−8.567	YES
3498	−3503.156	−5.156	0.660	−3502.496	−4.496	YES

Table 12. Data for Run #3 backward with a mesh of 5×5 LEDs (values in μm).

CMM	Image	Error Image	Compensation	Image Compensated	Error Compensation	Coincidence
498	−497.074	0.926	−0.532	−497.606	0.394	YES
999	−991.402	7.598	−4.628	−996.030	2.970	YES
1492	−1493.747	−1.747	2.333	−1491.414	0.586	YES
1992	−1992.997	−0.997	1.684	−1991.313	0.687	YES
2487	−2491.310	−4.310	1.007	−2490.303	−3.303	YES
2985	−2993.930	−8.930	0.791	−2993.139	−8.139	YES
3486	−3493.567	−7.567	−3.367	−3496.934	−10.934	NO

Table 13 summarizes the mean errors (\bar{e}) and the standard deviation errors (σ) calculated in each run of the experimental tests. The global mean (4.786 μm) and standard deviation (5.698 μm) were also calculated.

Table 13. Summary of the errors, in absolute value, provided by the proposed vision positioning algorithm.

	#1 Forward	#1 Backward	#2 Forward	#2 Backward	#3 Forward	#3 Backward	Global
Mean (μm)	3.936	2.519	8.708	4.883	4.086	4.582	4.786
Mean (%)	0.79	0.50	1.74	0.98	0.8	0.92	0.96
σ (μm)	4.771	4.238	4.212	6.586	3.699	5.567	5.698
σ (%)	0.95	0.85	0.84	1.32	0.74	1.11	1.19

As is seen in the graphs of Figure 8, the precision of positioning depended strongly on the initial error function, so the variation of the error was less than $\pm 2\mu\text{m}$, except several discrete points that were measured in a transition between columns of LEDs that were not so homogeneous in the LCD.

One remarkable result was that the orientation of the compensation error coincided with the sign of the error checked at almost all points. This indicated that there was variation of the orientation of the focal line with respect to the screen coordinate system, although this was not applied efficiently to improve the precision of measurement. This was probably due to the problems evaluating the trigonometric functions in angles with values less than 10^{-3} radians.

Therefore, due to the distance from the vanishing points of the lines, the estimation of the angular error did not turn out to be very precise in its quantification, although it provided qualitative descriptors on the direction and magnitude of the variation in the inclination of the camera.

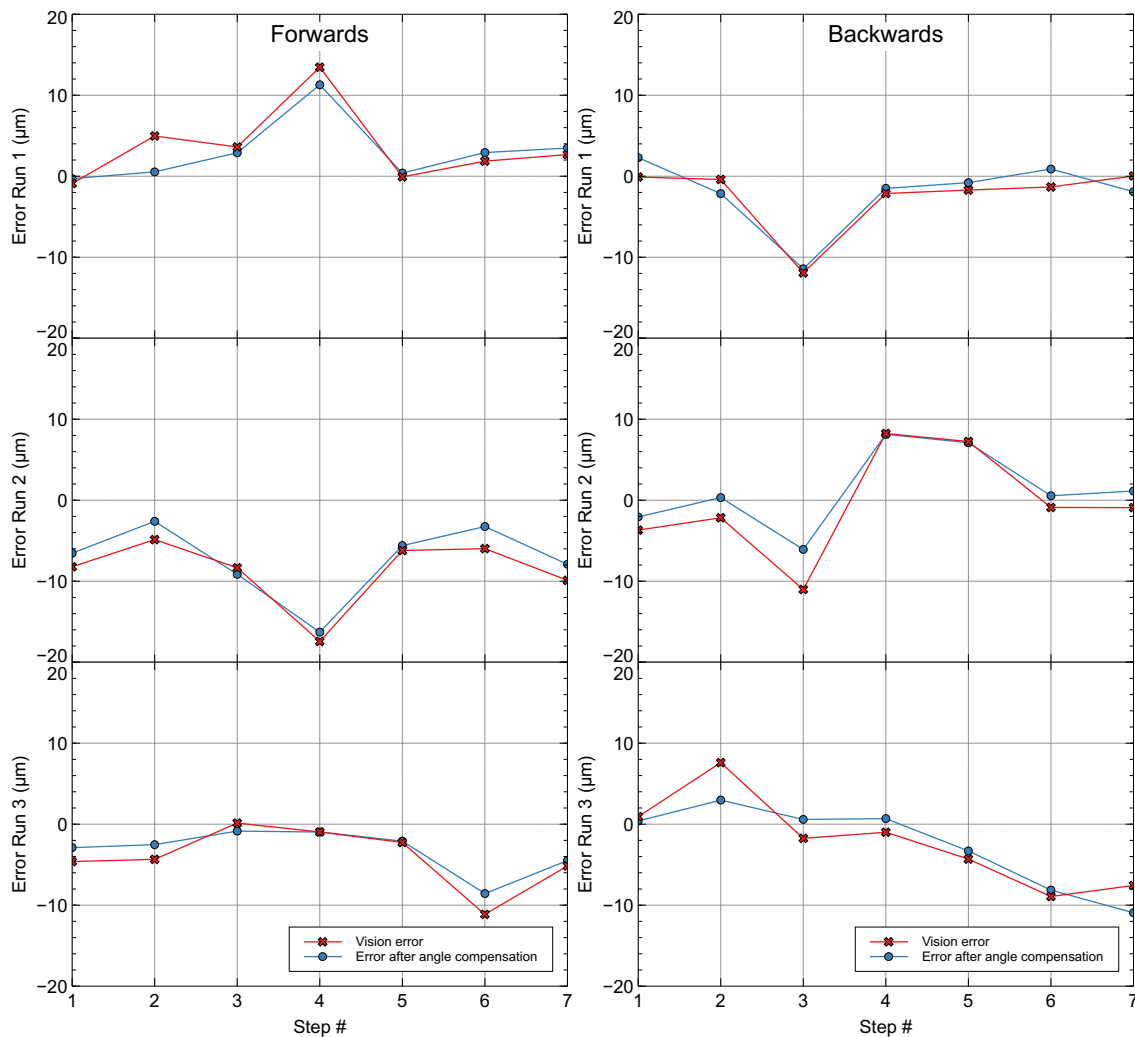


Figure 8. Error due to the vision system before and after angle compensation was applied.

5. Conclusions

A new method was developed to position the tool in a micromachine system based on a camera-LCD screen positioning system that also provided information on the angular deviations of the tool axis during operation.

The method gave a good approximation of the center point of the rectangle with a mean error of 0.96%, considering not only the vision algorithm, but also the mechanical test device, and provided the inclination of the workpiece with respect to the LCD-screen reference coordinate system.

The equivalent square was calculated as regression of the lines that could be drawn through the centers of gravity of each of the LEDs. The lack of parallelism between the sides of the square indicated an inclination of the camera axis relative to perpendicular to the screen. The variation of this

inclination introduced errors in the displacements that were added to the simple displacement of the center of gravity and whose compensation was also calculated in this article.

A test of the method was designed with the assistance of a Coordinate Measurement Machine (CMM) to verify the accuracy of the positioning method. The test performed provided good accuracy in measuring the position of the designed method, but a high dispersion in the angular deviation was detected, although the orientation of the inclination was appropriate in almost all cases (85.7%). This was due to the small value of the angles that made the approximations of the trigonometric functions very erratic. With accurate formulas to approximate trigonometric functions for small angles, the method could help in obtaining more accurate measurements.

Author Contributions: Conceptualization, Ó.d.F.O. and M.E.A.; methodology, Ó.d.F.O.; software, Ó.d.F.O. and M.E.A.; validation, Ó.d.F.O., H.T.S.R., and J.C.-B.M.-H.; formal analysis, Ó.d.F.O., H.T.S.R., and J.C.-B.M.-H.; writing, original draft preparation, Ó.d.F.O.; writing, review and editing, M.E.A., H.T.S.R., and J.C.-B.M.-H.; visualization, Ó.d.F.O., M.E.A., H.T.S.R., and J.C.-B.M.-H.

Funding: This research was funded by Ingeniería Murciana SL by a private contract.

Acknowledgments: The authors want to thank the University Center of Defense at the Spanish Air Force Academy, MDE-UPCT, for financial support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LCD	Liquid Crystal Display
MMT	Micromachine Tool
ppi	points per inch
CMM	Coordinates Measuring Machine

References

1. Leviton, D.B.; Kirk, J.; Lobsinger, L. Ultra-high resolution Cartesian absolute optical encoder. In *Recent Developments in Traceable Dimensional Measurements II, Proceedings of the SPIE's 48th Annual Meeting, San Diego, CA, USA, 2–4 August 2003*; SPIE: Bellingham, WA, USA, 2003; pp. 111–121, doi:10.1117/12.518376. [[CrossRef](#)]
2. Leviton, D.B. Method and Apparatus for Two-Dimensional Absolute Optical Encoding. U.S. Patent 6765195 B1, 20 July 2004.
3. Montes, C.A.; Ziegert, J.C.; Wong, C.; Mears, L.; Tucker, T. 2-D absolute positioning system for real time control applications. In *Proceedings of the Twenty-Fourth Annual Meeting of the American Society for Precision Engineering*, Rosemont, IL, USA, 13 September 2010.
4. De Francisco-Ortiz, O.; Sánchez-Reinoso, H.; Estrems-Amestoy, M. Development of a Robust and Accurate Positioning System in Micromachining Based on CAMERA and LCD Screen. *Procedia Eng.* **2015**, *132*, 8, doi:10.1016/j.proeng.2015.12.575. [[CrossRef](#)]
5. De Francisco Ortiz, O.; Sánchez Reinoso, H.; Estrems Amestoy, M.; Carrero-Blanco Martínez-Hombre, J. Position precision improvement throughout controlled led paths by artificial vision in micromachining processes. *Procedia Manuf.* **2017**, *13*, 197–204, doi:10.1016/j.promfg.2017.09.044. [[CrossRef](#)]
6. Byun, S.; Kim, M. Real-Time Positioning and Orienting of Pallets Based on Monocular Vision. In *Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence*, Dayton, OH, USA, 3–5 November 2008; Volume 2, pp. 505–508, doi:10.1109/ICTAI.2008.124. [[CrossRef](#)]
7. Zhang, B.; Wang, J.; Rossano, G.; Martinez, C.; Kock, S. Vision-guided robot alignment for scalable, flexible assembly automation. In *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, Karon Beach, Phuket, Thailand, 7–11 December 2011; pp. 944–951, doi:10.1109/ROBIO.2011.6181409. [[CrossRef](#)]
8. Zhou, K.; Wang, X.J.; Wang, Z.; Wei, H.; Yin, L. Complete Initial Solutions for Iterative Pose Estimation from Planar Objects. *IEEE Access* **2018**, *6*, 22257–22266, doi:10.1109/access.2018.2827565. [[CrossRef](#)]

9. Lyu, D.; Xia, H.; Wang, C. Research on the effect of image size on real-time performance of robot vision positioning. *EURASIP J. Image Video Process.* **2018**, *2018*, 112, doi:10.1186/s13640-018-0328-0. [[CrossRef](#)]
10. Montijano, E.; Cristofalo, E.; Zhou, D.; Schwager, M.; Sagüés, C. Vision-Based Distributed Formation Control Without an External Positioning System. *IEEE Trans. Robot.* **2016**, *32*, 339–351, doi:10.1109/TRO.2016.2523542. [[CrossRef](#)]
11. Yang, S.; Jiang, R.; Wang, H.; Ge, S.S. Road Constrained Monocular Visual Localization Using Gaussian-Gaussian Cloud Model. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3449–3456, doi:10.1109/TITS.2017.2685436. [[CrossRef](#)]
12. Guo, D.; Wang, H.; Leang, K.K. Nonlinear vision-based observer for visual servo control of an aerial robot in global positioning system denied environments. *J. Mech. Robot.* **2018**, *10*, 061018. [[CrossRef](#)]
13. Vivacqua, R.P.D.; Bertozzi, M.; Cerri, P.; Martins, F.N.; Vassallo, R.F. Self-Localization Based on Visual Lane Marking Maps: An Accurate Low-Cost Approach for Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 582–597. doi:10.1109/TITS.2017.2752461. [[CrossRef](#)]
14. Fang, J.; Wang, Z.; Zhang, H.; Zong, W. Self-localization of Intelligent Vehicles Based on Environmental Contours. In Proceedings of the 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), Singapore, 18–20 July 2018; pp. 624–629. doi:10.1109/ICARM.2018.8610687. [[CrossRef](#)]
15. Islam, K.T.; Wijewickrema, S.; Pervez, M.; O’Leary, S. Road Trail Classification using Color Images for Autonomous Vehicle Navigation. In Proceedings of the 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, Australia, 10–13 December 2018; pp. 1–5, doi:10.1109/DICTA.2018.8615834. [[CrossRef](#)]
16. You, S.; Neumann, U.; Azuma, R. Hybrid inertial and vision tracking for augmented reality registration. In Proceedings of the IEEE Virtual Reality (Cat. No. 99CB36316), Houston, TX, USA, 13–17 March 1999; pp. 260–267, doi:10.1109/VR.1999.756960. [[CrossRef](#)]
17. Kim, J.; Jun, H. Vision-based location positioning using augmented reality for indoor navigation. *IEEE Trans. Consum. Electron.* **2008**, *54*, 954–962, doi:10.1109/TCE.2008.4637573. [[CrossRef](#)]
18. Samarasekera, S.; Oskiper, T.; Kumar, R.; Sizintsev, M.; Branzoi, V. Augmented Reality Vision System for Tracking and Geolocating Objects of Interest. U.S. Patent 9,495,783, 15 November 2016.
19. Suenaga, H.; Tran, H.H.; Liao, H.; Masamune, K.; Dohi, T.; Hoshi, K.; Takato, T. Vision-based markerless registration using stereo vision and an augmented reality surgical navigation system: A pilot study. *BMC Med. Imaging* **2015**, *15*, 51. [[CrossRef](#)]
20. Rajeev, S.; Wan, Q.; Yau, K.; Panetta, K.; Agaian, S.S. Augmented reality-based vision-aid indoor navigation system in GPS denied environment. In *Proceedings of the SPIE 10993, Mobile Multimedia/Image Processing, Security, and Applications*; SPIE: Baltimore, MA, USA, 2019, doi:10.1117/12.2519224. [[CrossRef](#)]
21. Wefelscheid, C.; Wekel, T.; Hellwich, O. Monocular Rectangle Reconstruction Based on Direct Linear Transformation. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP 2011), Vilamoura, Portugal, 5–7 March 2011; pp. 271–276.
22. Haralick, R.M. Determining camera parameters from the perspective projection of a rectangle. *Pattern Recognit.* **1989**, *22*, 225–230, doi:10.1016/0031-3203(89)90071-x. [[CrossRef](#)]
23. Quan, L.; Lan, Z. Linear N-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 774–780, doi:10.1109/34.784291. [[CrossRef](#)]
24. Estrems Amestoy, M.; de Francisco Ortiz, O. Global Positioning from a Single Image of a Rectangle in Conical Perspective. *Sensors* **2019**, *19*, 5432, doi:10.3390/s19245432. [[CrossRef](#)] [[PubMed](#)]
25. Abidi, M.A.; Chandra, T. A new efficient and direct solution for pose estimation using quadrangular targets—Algorithm and evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 534–538, doi:10.1109/34.391388. [[CrossRef](#)]
26. Haralick, R.M. Using perspective transformations in scene analysis. *Comput. Graph. Image Process.* **1980**, *13*, 191–221, doi:10.1016/0146-664x(80)90046-5. [[CrossRef](#)]
27. Sim, R.; Little, J. Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters. *Image Vis. Comput.* **2009**, *27*, 167–177, doi:10.1016/j.imavis.2008.04.003. [[CrossRef](#)]
28. Valencia-Garcia, R.; Martínez-Béjar, R.; Gasparetto, A. An intelligent framework for simulating robot-assisted surgical operations. *Expert Syst. Appl.* **2005**, *28*, 425–433, doi:10.1016/j.eswa.2004.12.003. [[CrossRef](#)]

29. Pichler, A.; Akkaladevi, S.; Ikeda, M.; Hofmann, M.; Plasch, M.; Wögerer, C.; Fritz, G. Towards Shared Autonomy for Robotic Tasks in Manufacturing. *Procedia Manuf.* **2017**, *11*, 72–82, doi:10.1016/j.promfg.2017.07.139. [[CrossRef](#)]
30. Broggi, A.; Dickmanns, E. Applications of computer vision to intelligent vehicles. *Image Vis. Comput.* **2000**, *18*, 365–366. [[CrossRef](#)]
31. Patterson, T.; McClean, S.; Morrow, P.; Parr, G.; Luo, C. Timely autonomous identification of UAV safe landing zones. *Image Vis. Comput.* **2014**, *32*, 568–578, doi:10.1016/j.imavis.2014.06.006. [[CrossRef](#)]
32. González, D.; Pérez, J.; Milanés, V. Parametric-based path generation for automated vehicles at roundabouts. *Expert Syst. Appl.* **2017**, *71*, 332–341, doi:10.1016/j.eswa.2016.11.023. [[CrossRef](#)]
33. Sanchez-Lopez, J.; Pestana, J.; De La Puente, P.; Campoy, P. A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-UAV systems: Simulation and experimentation. *J. Intell. Robot. Syst.* **2015**, *84*, 779–797. [[CrossRef](#)]
34. Olivares-Mendez, M.; Kannan, S.; Voos, H. Vision based fuzzy control autonomous landing with UAVs: From V-REP to real experiments. In Proceedings of the 2015 23rd Mediterranean Conference on Control and Automation (MED), Torremolinos, Spain, 16–19 June 2015; pp. 14–21, doi:10.1109/MED.2015.7158723. [[CrossRef](#)]
35. Romero-Ramirez, F.J.; Muñoz-Salinas, R.; Medina-Carnicer, R. Speeded up detection of squared fiducial markers. *Image Vis. Comput.* **2018**, *76*, 38–47, doi:10.1016/j.imavis.2018.05.004. [[CrossRef](#)]
36. Germanese, D.; Leone, G.R.; Moroni, D.; Pascali, M.A.; Tampucci, M. Long-Term Monitoring of Crack Patterns in Historic Structures Using UAVs and Planar Markers: A Preliminary Study. *J. Imaging* **2018**, *4*, 99, doi:10.3390/jimaging4080099. [[CrossRef](#)]
37. An, G.H.; Lee, S.; Seo, M.W.; Yun, K.; Cheong, W.S.; Kang, S.J. Charuco Board-Based Omnidirectional Camera Calibration Method. *Electronics* **2018**, *7*, 421. doi:10.3390/electronics7120421. [[CrossRef](#)]
38. Pflugi, S.; Vasireddy, R.; Lerch, T.; Ecker, T.; Tannast, M.; Boemke, N.; Siebenrock, K.; Zheng, G. Augmented marker tracking for peri-acetabular osteotomy surgery. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2017**, *2017*, 937–941. doi:10.1109/EMBC.2017.8036979. [[CrossRef](#)]
39. Lima, J.P.; Roberto, R.; Simões, F.; Almeida, M.; Figueiredo, L.; Teixeira, J.M.; Teichrieb, V. Markerless tracking system for augmented reality in the automotive industry. *Expert Syst. Appl.* **2017**, *82*, 100–114. [[CrossRef](#)]
40. Chen, P.; Peng, Z.; Li, D.; Yang, L. An improved augmented reality system based on AndAR. *J. Vis. Commun. Image Represent.* **2016**, *37*, 63–69, doi:10.1016/j.jvcir.2015.06.016. [[CrossRef](#)]
41. Khattak, S.; Cowan, B.; Chepurna, I.; Hogue, A. A real-time reconstructed 3D environment augmented with virtual objects rendered with correct occlusion. In Proceedings of the 2014 IEEE Games Media Entertainment, Toronto, ON, Canada, 22–24 October 2014; pp. 1–8.
42. De Francisco Ortiz, O.; Sánchez Reinoso, H.; Estrems Amestoy, M.; Carrero-Blanco, J. Improved Artificial Vision Algorithm in a 2-DOF Positioning System operated under feedback control in micromachining. In *Euspen's 17th International Conference & Exhibition Proceedings*; Billington, D., Phillips, D., Eds.; Euspen: Northampton, UK, 2017; pp. 460–461.

