applied optics

Method for targetless tracking subpixel in-plane movements

JULIAN ESPINOSA,[1,*] JORGE PEREZ,[1] BELEN FERRER,[2] AND DAVID MAS[1]

[1]Institute of Physics Applied to the Sciences and Technologies, Universidad de Alicante, Carretera San Vicente del Raspeig, s/n-03690 San Vicente del Raspeig, Spain

[2]Department of Civil Engineering, Universidad de Alicante, Carretera San Vicente del Raspeig, s/n-03690 San Vicente del Raspeig, Spain

*Corresponding author: julian.espinosa@ua.es

Received 28 May 2015; accepted 7 August 2015; posted 12 August 2015 (Doc. ID 241988); published 31 August 2015

We present a targetless motion tracking method for detecting planar movements with subpixel accuracy. This method is based on the computation and tracking of the intersection of two nonparallel straight-line segments in the image of a moving object in a scene. The method is simple and easy to implement because no complex structures have to be detected. It has been tested and validated using a lab experiment consisting of a vibrating object that was recorded with a high-speed camera working at 1000 fps. We managed to track displacements with an accuracy of hundredths of pixel or even of thousandths of pixel in the case of tracking harmonic vibrations. The method is widely applicable because it can be used for distance measuring amplitude and frequency of vibrations with a vision system.

OCIS codes: (120.7280) Vibration analysis; (070.4790) Spectrum analysis; (100.6640) Superresolution; (100.4999) Pattern recognition, target tracking.

http://dx.doi.org/10.1364/AO.54.007760

1. INTRODUCTION

Movement detection and frequency measurements are of great importance in many fields and can be monitored through contact sensors, such as accelerometers, or noncontact methods, such as laser vibrometers [1]. The use of accelerometers is sometimes dismissed because of the complexity or accessibility of the specimen to be measured, whereas a laser vibrometer's high cost is still a major drawback in most cases. Thus, computer vision systems [2–8] have been demonstrated to be a reliable alternative to traditional methods in the last few years. Independently, whether or not they use object recognition, vision-based methods assume that object movement or vibration can be perceived through changes in the light reflected or diffused by a moving target. Big movements may be easy to detect, but small amplitude ones produce subtle changes, so they usually require of high-resolution and low-noise cameras in order to reduce false movement detection. Additionally, one must take into account that those movements may not affect the whole image but only very few unconnected pixels; thus, special techniques must be applied in order to recognize and track the object movement.

Subpixel techniques have been shown to produce accurate results because they are able to detect object shifts even in those cases where movement is not apparent. Their effect is to increase the theoretical resolution of the system by several orders of magnitude [2]. Basically, these techniques consist of capturing and digitally postprocessing scenes containing targets of known shape. These targets are recognized and extracted from the scene. Then, through smart interpolation of their different features, it is possible to decrease the uncertainty in the object position, even below the basic pixel unit. Therefore, a fully automated tracking method with subpixel accuracy includes target recognition and location through features extraction [9,10]. The most complicated task is target recognition because it requires a proper analysis and segmentation of the image, although it can be simplified using targets easy to recognize [11].

After presenting the mathematical basis of the technique [12] and discussing the maximum theoretical subpixel resolution achievable, in [5], the authors demonstrated the possibility of measuring the vibration frequency of objects in a high-speed video sequence simply by counting luminance changes in local regions. The proposal consisted of selecting a local neighborhood or region of interest and applying a multilevel binary threshold to the image within this region. Then, different binary versions of the region were obtained and analyzed. The object did not have to fulfill any specific requirement, and no special feature was detected or matched. Unfortunately, this technique did not provide information about the amplitude, so the movement description was incomplete.

In this work, we go beyond [5] and propose a technique to track the subpixel movement of an object besides using its frequency of vibration and without using target or template

matching, but with the specific need of the object image to have two nonparallel straight segments and limited to in-plane translation. The manuscript is structured as follows. In the next section, we describe and explain the requirements that the object and the movement must meet. Then, we present the experimental setup and the algorithm used to obtain the results stated in the fourth section. Finally, we explain our conclusions.

2. LIMITS AND CONDITIONS

The movements of an object can be displacements and/or rotations parallel and/or perpendicular to the image plane. In principle, in this work, we only consider those movements parallel to the image plane.

Let us suppose an object that horizontally moves within the image plane and assume that it keeps its shape during the movement. Generally, one can find straight edges that move with the object in the scene and reduce the problem of object tracking to detect and follow the movement of those straight edges or segments.

A digital arc S is the digitization of a straight-line segment if and only if it has the "chord property": the line segment joining any two points of S resides everywhere within a distance 1 of S [13]. As an example, in Fig. 1(a), we represent an oblique segment over an array of sensors. The pixels of the sensor are represented as dotted–lined squares, and the background grid divides each pixel into a 4×4 matrix. The segment is plotted as a white line, and we assume that it activates all the pixels where it falls (gray color pixels). If Fig. 1(a) shows a frame with the digitization of a segment of the moving object, in Figs. 1(b)–1(d), we illustrate the digitization of the segment as the object displaces. Each frame corresponds to a horizontal subpixel displacement of 0.25 px to the right, with respect to the previous one, so the object has moved a distance equivalent to one pixel over the array of sensors.

Cartesian coordinates of the activated pixels from the digitization of a segment can be fitted to a straight-line equation in order to obtain parameters (slope and y intersect) that characterize that segment. Obviously, the shape of the segment that the sensor "sees," i.e., the set of activated pixels, is different to the shape of the segment itself due to the digitization process. However, we know that it corresponds to the digitization of a straight segment, so we force the fitting to a straight line. Moreover, note that the set of activated pixels can change at a subpixel displacement, although the original segment maintains its shape, as we illustrate in Figs. 1(a)–1(d). There, each 0.25 px horizontal displacement completely changes the activated pixels, so the Cartesian coordinates fitting to a straight line will provide different parameters at each subpixel displacement.

Tracking the straight-line parameters allow tracking the movement of the segment. Thus, if the segment rotates, both slope and y intersect will change, whereas, if it just displaces, the slope should remain constant and only vary the y intersect. However, this does not frequently happen due to the discretization process of the image over the sensor array; thus, it is difficult to affirm whether or not the segment has slightly rotated. A clear example is shown in Fig. 1, where the slope resulting from the fitting changes frame to frame, although the segment just displaces. The best-fitting straight lines are plotted in red in Fig. 2 for each one of the frames from Fig. 1. Therefore, from these parameters, we are unable to distinguish small rotations from displacements.

Let us further limit the movements parallel to only translations within the image plane and therefore exclude rotations. Thus, we can take advantage in the fact that we know that the segment does not rotate and impose the slope of the straight segment by fixing this parameter in the least-squares fitting. Turning back to the above example, in Fig. 2 we have represented in blue the best-fitting slope-imposed straight line for each frame. We imposed the slope resulting in the average slope of the previously obtained best-fitting lines. There, the reader can appreciate how the y intersect of the blue line changes frame to frame and its slope remains constant.

This way, it seems that we will be able to track subpixel displacements just by obtaining and tracking the y intersect of a straight line—but not yet. We still cannot distinguish the direction and sense of the displacement. To get a better understanding, in Fig. 3(a), we represent in blue a straight line together with the lines after independent shifts of Δx, Δy, and Δd in the horizontal, vertical, and normal to the segment axis directions (red, green, and blue dashed lines, respectively). Black dots represent x and y intersects. One can see that the three dashed lines intersect at the same points, although the line has displaced different distances in different directions. Therefore, the y intersect does not allow distinguishing between these three different situations.

The problem can be overcome if we take two secant straight lines and track the intersection of both lines. In Fig. 3(b), we represent two blue lines that intersect (black ring) together with

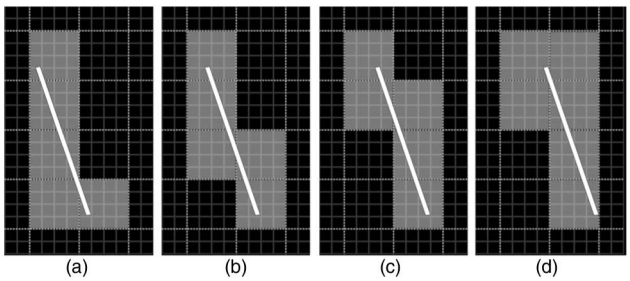

Fig. 1. (a) Oblique segment projected on an array of pixels (dotted–lined squares). Gray-color pixels are those activated. Each pixel is divided into 4×4 areas. The array of activated pixels change if the segment horizontally displaces (b) 0.25 px, (c) 0.50 px, and (d) 0.75 px.

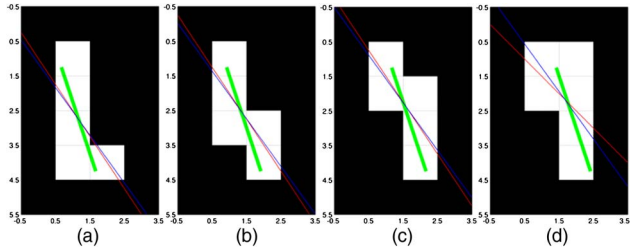

Fig. 2. Activated pixels of the sequence in Fig. 1 together with best-fitting straight line (red lines) and slope-forced best-fitting straight line (blue lines) if the segment horizontally displaces (a) 0 px, (b) 0.25 px, (c) 0.50 px, and (d) 0.75 px.

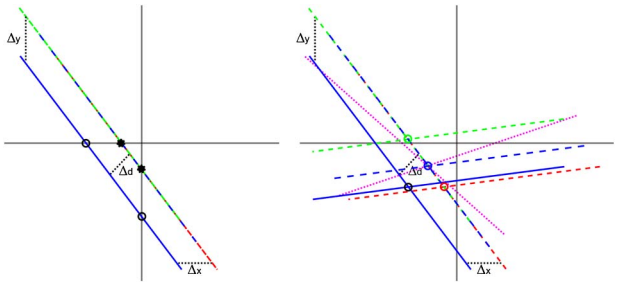

Fig. 3. (a) Straight line (blue solid line) that displaces horizontally, vertically, and arbitrarily (red, green, and blue dashed lines, respectively). (b) Intersections of two straight lines (blue solid lines) that displace horizontally, vertically, and arbitrarily (red, green, and blue dashed lines, respectively) and two straight lines that displace and rotate (pink dotted lines).

the intersections resulting from displacing them Δx horizontally, Δy vertically, and Δd arbitrarily (red, green, and blue rings, respectively). There, we can clearly distinguish each displacement. However, if the object rotates, we cannot distinguish the movement from a displacement simply by tracking the intersection of two secant lines. We illustrate this case in Fig. 3(b), where the intersection of the displaced and rotated object (pink dotted lines) coincides with the point corresponding to one only displaced (blue dashed lines).

In summary, if we limit the movement of an object to only translations within the image plane, we propose to describe that motion simply by tracking the intersection of two straight lines. These lines are the best-fitting straight lines of the digitization of two nonparallel segments of the object image.

3. METHOD

We assume that we are able to extract two nonparallel segments, s_1 and s_2, from the edges of the intensity image of any moving object in a scene. As we previously stated, best-fitting straight lines to each segment intersect in a point (x_c, y_c) that will move like the object will do. If the object simply translates parallel to the image plane, tracking that intersection will allow describing object motion with subpixel accuracy. Next, we present the experiment performed in order to test our proposal, thus measuring displacement and frequency of vibration. Then, we describe the algorithm used to compute the intersection of that segment from a recorded sequence of a moving object in a scene. Briefly, we first obtained the slopes of a segment using the Hough transform from the Canny edges of the image. These slopes were then used to compute the forced best-fitting straight lines that provided the intersection point coordinates at each frame of the sequence $[x_c(t), y_c(t)]$.

A. Experimental Setup

We have implemented our technique to measure the movement of a complex pattern situated on a vibrating platform. We have used a Vibe-Tribe Troll vibrating loudspeaker connected to a computer in order to induce a vertical vibration movement. The speaker was installed upside down on isolating Sorbothane foam. The object was a frame with a QR code, and the speaker vibrated at frequencies of 330 and 445 Hz. Moreover, a ceramic shear ICP accelerometer with a sensitivity of 1000 mV/g and frequency range from 0.5 to 3 kHz was glued at the top of the frame in order to check the vibration frequency and to obtain an estimation of the movement.

The measurement was taken with an AOS X-PRI camera, working at 1000 fps and located at 50 cm from the object. A 50 W standard halogen lamp connected to a stabilized AC/DC converter illuminated the scene. The target side was 40 mm with a frame size of 800×600 px, thus giving a resolution of 5.30 px/mm. Figure 4 shows the first frame of the sequence.

B. Algorithm

The captured sequences were processed off-line. The edges from the intensity image were extracted from each frame of a sequence using the Canny method [14]. This method finds edges by looking for local maxima of the gradient of the image. The gradient is calculated using the derivative of the Gaussian filtered image. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise and better to detect true weak edges. Next, in the first frame of the recorded sequence, we selected and cropped two regions of the processed image containing two nonparallel segments (s_1 and s_2). These regions were later automatically cropped in all frames of the sequence.

The set of Cartesian coordinates $\mathbf{X}_q$ and $\mathbf{Y}_q$ of the activated pixels that form s_1 and s_2 were then fitted to two straight lines:

$$\mathbf{Y}_q = m_k \mathbf{X}_q + n_k; \qquad k = 1, 2; \qquad q = i, j; \tag{1}$$

where $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$, being N_1 and N_2 the number of the activated pixels of each segment. These straight lines intersect at a point (x_c, y_c). Let us remember that the object under study did not rotate or change shape nor did the edges; thus, the slopes of the straight lines must remain constant. Therefore, we looked for the best-fitting slope-imposed straight line for each segment. In order to estimate the slopes (m_k) of the segments, we used the standard Hough transform [12], which is designed to detect lines, using the Hesse normal form [15] of a line:

$$\rho = x \cos(\theta) + y \sin(\theta). \tag{2}$$

The variable ρ is the distance from the origin to the line along a vector perpendicular to the line, and θ is the angle between the x axis and this vector. The Hough function

Fig. 4. First frame of the sequence of the QR code used as an object.

generates a parameter space matrix whose rows and columns correspond to these ρ and θ values, respectively. We computed the Hough transform at each cropped area in the first frame and looked for the peak value in the parameter space, which represents the line in the input image; therefore, we obtain the values for the slopes of segments s_1 and s_2.

The best estimation of y intersections was obtained by solving the linear least-squares problem described by the system of equations that can be deduced from Eq. (1). Our basic equation is $\mathbf{Y} - m\mathbf{X} = n\mathbf{O}$, where n is the y intersect, $\mathbf{O}$ is a vector of ones with the same number of elements than activated pixels, and m is the slope previously estimated using the Hough transform. The problem is the choice of the coefficients n in order to minimize the sum of squared errors. The function to be minimized is [16,17]

$$\phi(n) = n^T n^T + t^T(\mathbf{Y} - m\mathbf{X} - n\mathbf{O}), \tag{3}$$

where t is the vector of Lagrange multipliers, and the super index T means matrix transposition. The minimization yields

$$n = \mathbf{O}^{+}(\mathbf{Y} - m\mathbf{X}), \tag{4}$$

where the super index + stands for matrix pseudoinverse, developed by Moore and later by Penrose [18]. In order to calculate the pseudoinverse, we used a method called singular value decomposition implemented in MATLAB based in LAPACK routines [19]. This method provides a solution to Eq. (4) regardless of being a determined, undetermined, or overdetermined system.

Once we had obtained the straight-line parameters, we computed the intersection coordinates (x_c, y_c):

$$x_c = \frac{\Delta n}{\Delta m}; \qquad y_c = m_k x_c + n_k. \tag{5}$$

Finally, the movement was tracked from the computed intersection coordinates at each frame of the sequence $[x_c(t), y_c(t)]$. We were also interested in evaluating the frequency of vibration of the movement, so we obtained the Fourier transform modulus of the evolution in time of each coordinate. In the Fourier spectrum, we looked for the main frequencies of vibration that appear as peaks. In the case of having a clear unique main peak, we assumed that the signal was a harmonic vibration and reconstructed it by means of the inverse Fourier transform.

4. RESULTS

Following the procedure stated in Section 3, in the initial frame of the recorded sequence, we have first extracted edges using the Canny method and selected two areas where we could find two nonparallel segments (s_1 and s_2). From them, we have computed the intersection and tracked its movement. In Fig. 5, we show the edges computed using the Canny algorithm from a cropped region around a corner of the QR code and the areas that we have selected to obtain the segments.

Note that the movement produced by the vibrating platform in the experimental setup is mainly in the vertical direction; thus, we only have to track the y coordinate of the intersection point to describe the movement and to find the vibration frequencies. First, we would like to highlight that the platform was set to vibrate at 330 Hz during approximately

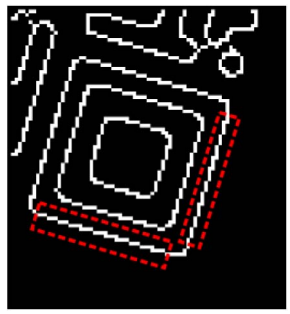

Fig. 5. Image obtained after cropping and Canny edge extraction. Red boxes indicate the areas from where we extracted the nonparallel segments.

the first second and after that, at 445 Hz. Figure 6(a) shows the evolution in time of the computed y coordinate of the intersection point, and Fig. 6(b) represents its Fourier spectrum. The change in frequency of vibration cannot be appreciated in Fig. 6(a), but both frequencies are present in the Fourier spectrum in Fig. 6(b) together with a peak around 5 Hz.

In order to perform a better analysis, we have cut the sequence in two and then looked for a unique vibrating frequency. Thus, we have chosen around a half second from each part of the signal (vibrating at 330 and 445 Hz). The displacement values have been compared with those computed from the acceleration data that were captured by the accelerometer. The double integration of acceleration consisted of several steps. First, the acceleration signal was filtered with a sixth-order high-pass Butterworth filter and a cut-off frequency of 40 Hz. This frequency has been set where the magnitude response of the filter is $\sqrt{1/2}$. Next, in order to obtain the velocity, we computed the cumulative integral of the filtered acceleration with respect to time using trapezoidal integration. Then, the resulting velocity was a sixth-order high-pass Butterworth filtered with a cut-off frequency of 40 Hz. Finally, the filtered signal of velocity was again integrated to obtain the displacement signal.

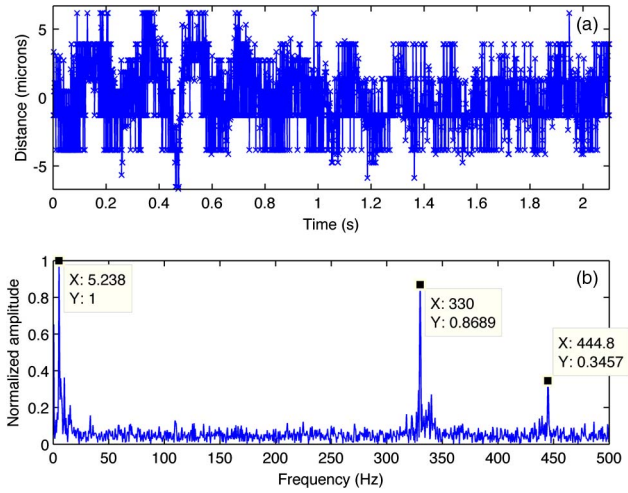

Fig. 6. (a) y coordinate of the intersection. (b) Main frequencies in the Fourier spectrum.

In Fig. 7(a), we represent the computed Fourier spectrum of the y coordinate of the intersection point in a 0.5 s fragment of the sequence vibrating at 330 Hz. There, we have found the expected frequency of 330 Hz together with a low frequency peak of 8 Hz. The main frequency peak was selected to reconstruct the signal through the inverse Fourier transform. In Fig 7(b), we represent with blue dots the y coordinate of the intersection computed using our technique and, with black dots, the main frequency peak reconstructed signal. In this graph, a short time interval of only 0.05 s has been selected for better display. The red line and circles represent the displacement obtained from the accelerometer subsampled at the rate of the camera. In order to check the accuracy of our method, we compare the y coordinate and the reconstructed signals with that from the accelerometer using the root mean square error (RMSE). They, respectively, resulted 3.6 and 0.73 μm, i.e., a subpixel accuracy of around 0.02 and 0.004 px.

Regarding the fragment of the sequence vibrating at 445 Hz, in Fig. 8(a), we represent the computed Fourier spectrum of the y coordinate of the intersection point. There, we have found the expected frequency of 445 Hz (main peak), a low band of frequencies between 5 and 8 Hz together, and a peak at 110 Hz. In Fig 7(b), we represent, again in a time interval of 0.05 s, the y coordinate of the intersection computed using our technique (blue dots), the main frequency peak reconstructed signal (black dots) and the displacement obtained from the accelerometer subsampled at the rate of the camera (red line and dots). The RMSE between the y coordinate and the signal from the accelerometer was 3.3 μm, and the RMSE between the reconstructed signal and that from the accelerometer was 0.47 μm, i.e., a subpixel accuracy of around 0.02 and 0.003 px.

Therefore, the use of the presented method allows tracking in plane translation with a subpixel accuracy of 0.02 px. Moreover, one can deduce from the Fourier analysis that the tracked movement is a periodic vibration and reconstruct the signal using only the main frequency peak. If that is the case, the obtained accuracy reaches one order of magnitude less.

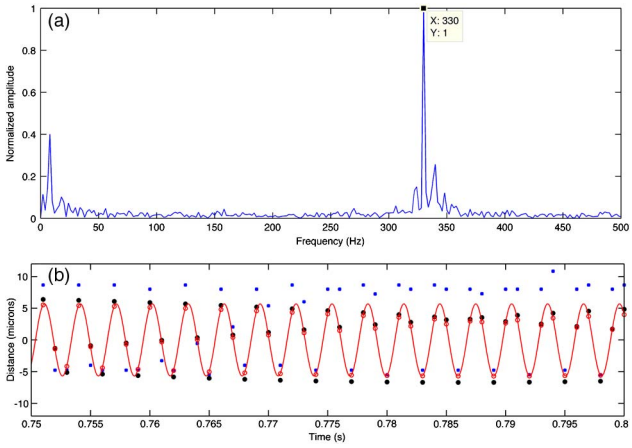

Fig. 7. (a) Fourier spectrum and main frequency peak selected to reconstruct the signal of a fragment of the y-coordinate signal when the object vibrates at 330 Hz. (b) Blue dots represent the y coordinate of the intersection, black dots are the reconstructed signal, and red line and circles represent the displacement from the accelerometer subsampled at the rate of the camera.

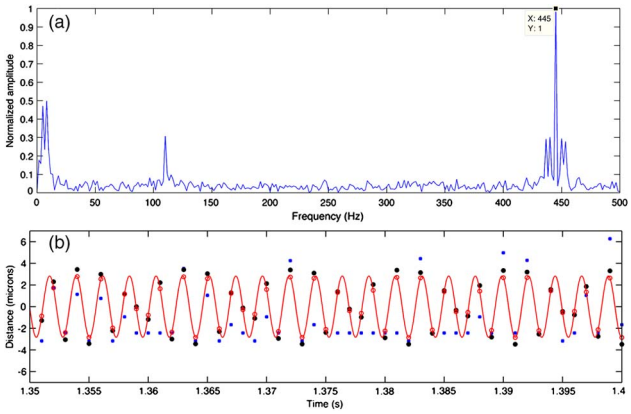

Fig. 8. (a) Fourier spectrum and main frequency peak selected to reconstruct the signal of a fragment of the y coordinate signal when the object vibrates at 445 Hz. (b) Blue dots represent the y coordinate of the intersection, black dots are the reconstructed signal, and red line and circles represent the displacement from the accelerometer subsampled at the rate of the camera.

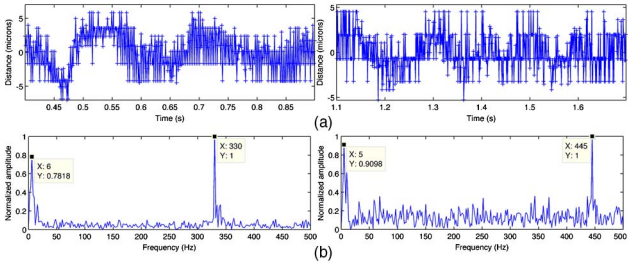

Fig. 9. (a) x coordinate of the intersection computed in the two selected fragments. (b) Main frequencies in the Fourier spectrum.

Regarding the x coordinate, although the object was expected not to vibrate or move in a horizontal direction, we have performed an analysis similar to that done to the y coordinate. We have found in the Fourier spectrum the frequencies of 330 and 445 Hz together with some low frequencies. This may be due to an imperfect alignment between the camera and the vibrating platform. We have separated the same fragments of the sequence (Fig. 9) and have found that the peaks at the frequencies of vibration are not as dominant as previously noted, being the height of low frequency peaks similar to that of those expected.

5. CONCLUSIONS

We have proposed a method to track movements with a subpixel accuracy of 0.02 px simply by tracking the intersection of the best-fitting straight lines of the digitization of two nonparallel segments of the object. Moreover, if the movement is a harmonic vibration, the accuracy can be increased up to 0.003 px. The subpixel accuracy is therefore improved with respect to previous works [6], whose authors claimed they were able to resolve movements up to 0.16 px on the CCD sensor.

The movement must be limited to only translations parallel to the image plane. Rotations cannot be distinguished by

tracking only one moving point. Regarding the movements in the axial direction, they would be appreciated in the image as a change in the size of the object, or they would simply blur the image. In this case, our method will not work correctly.

The method simultaneously provides a 2D description of the displacement, which is a significant advantage in front of uniaxial accelerometers besides being a noncontact measuring technique. It also may be more affordable than standard laser vibrometers.

In our experiment, we have used a high-speed camera, but it is not necessary for this method because it depends on the magnitude of the amplitude and frequency of the motion to be measured.

Funding. Generalitat Valenciana (Regional Government of Valencia) (PROMETEO/2011/021); Ministerio de Economía y Competitividad (Ministry of Economy and Competitiveness) (BIA2011-22704); Universidad de Alicante (GRE13-10).

REFERENCES

1. H. Tabatabai, D. E. Oliver, J. W. Rohrbaugh, and C. Papadopoulos, "Novel applications of laser Doppler vibration measurements to medical imaging," Sens. Imaging **14**, 13–28 (2013).
2. D. Mas, J. Espinosa, A. B. Roig, B. Ferrer, J. Perez, and C. Illueca, "Measurement of wide frequency range structural microvibrations with a pocket digital camera and sub-pixel techniques," Appl. Opt. **51**, 2664–2671 (2012).
3. Y. Fukuda, M. Q. Feng, Y. Narita, S. Kaneko, and T. Tanaka, "Vision-based displacement sensor for monitoring dynamic response using robust object search algorithm," IEEE Sens. J. **13**, 4725–4732 (2013).
4. G. Busca, A. Cigada, P. Mazzoleni, and E. Zappa, "Vibration monitoring of multiple bridge points by means of a unique vision-based measuring system," Exp. Mech. **54**, 255–271 (2014).
5. B. Ferrer, J. Espinosa, A. B. Roig, J. Perez, and D. Mas, "Vibration frequency measurement using a local multithreshold technique," Opt. Express **21**, 26198–26208 (2013).
6. Z. Wang, H. Nguyen, and J. Quisberth, "Audio extraction from silent high-speed video using an optical technique," Opt. Eng. **53**, 110502 (2014).
7. A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: passive recovery of sound from video," ACM Trans Graph. **33**, 1–10 (2014).
8. J. G. Chen, N. Wadhwa, Y.-J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk, "Modal identification of simple structures with high-speed video using motion magnification," J. Sound Vib. **345**, 58–71 (2015).
9. M. R. Shortis, T. A. Clarke, and T. Short, "Comparison of some techniques for the subpixel location of discrete target images," Proc. SPIE **2350**, 239–250 (1994).
10. M. R. Shortis, T. A. Clarke, and S. Robson, "Practical testing of the precision and accuracy of target image centering algorithms," Proc. SPIE **2598**, 65–76 (1995).
11. A. M. Bruckstein, L. O'Gorman, and A. Orlitsky, "Design of shapes for precise image registration," IEEE Trans. Inf. Theory **44**, 3156–3162 (1998).
12. D. Mas, B. Ferrer, J. T. Sheridan, and J. Espinosa, "Resolution limits to object tracking with subpixel accuracy," Opt. Lett. **37**, 4877–4879 (2012).
13. A. Rosenfeld, "Digital straight line segments," IEEE Trans. Comput. **C-23**, 1264–1269 (1974).
14. J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**, 679–698 (1986).
15. R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Commun. ACM **15**, 11–15 (1972).
16. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University, 1990).
17. T. J. Lemaire and A. Bassrei, "Three-dimensional reconstruction of dielectric objects by the coupled-dipole method," Appl. Opt. **39**, 1272–1278 (2000).
18. R. Penrose, "A generalized inverse for matrices," Math. Proc. Cambridge Philos. Soc. **51**, 406–413 (1955).
19. E. Anderson, Z. Bai, C. Bischof, L. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Software, Environments and Tools* (Society for Industrial and Applied Mathematics, 1999).