

Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Aeronáutica

**IDENTIFICAÇÃO DA DEFLEXÃO ESTRUTURAL DE UMA ASA
VIA PROCESSAMENTO DE IMAGENS**

Théodore Meynard
Orientador: Prof. Ricardo Afonso Angélico

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos da Universidade de São Paulo para obtenção do título de
Engenheiro Aeronáutico.

São Carlos
Dezembro / 2016

Contents

1	Introduction	7
1.1	Tendency in aeronautics	7
1.2	Exacerbated problems with high aspect flexible wings	8
1.3	Laboratory scale testing	8
1.4	The project	8
2	Experiments	11
2.1	Experiment description	11
2.2	Parameters description	13
3	Image processing	15
3.1	Transforming the raw image	15
3.2	Region of interest (ROI)	16
3.3	Corner identification in the chessboard motif	17
3.4	Filter and order the points	21
3.5	Interpolate the missing points	24
3.6	Smoothing of the points	26
3.7	Identification problem	26
4	Results	29
4.1	Comparison of the tip displacement	29
4.2	Verification of the global displacement	30
4.2.1	Full displacement calibration	32
4.2.2	Edge displacement calibration	32
4.2.3	Without calibration	35
5	Final remarks	37
5.1	Discussion of the results	37
5.2	Future work	37
5.3	Future application	37

Abstract

This graduation project will present an innovative way to measure the displacement field for a wing in a wind tunnel. This method needs only one camera at the root of the wing and a chessboard motif on the wing. All others studies done to measure the displacement with image processing use two cameras or another point of view. The location of the camera at the root of the wing allow an easier installation in a wing tunnel. Moreover this method can be easily adapt to measure the displacement of a wing in a flying airplane from inside the airplane. Finally, this method gives the total displacement field and not just the displacement at some points of the wing like conventional displacement sensors.

After an introduction of the aeronautics evolution and the need to test aeroelasticity models, this report will present the experimental setup used to simulated the displaced wing. Then, the methodology to treat the images to get an estimation will be presented. The software used for the image processing and the inverse problem is Matlab. Next, the results for all the experiment cases with different calibrations will be presented. Finally, the results and future work will be discussed.

This project can be use to test the wing of the airplane SKYFERRY developed in Airplane Project. It could be used to measure the aeroelastic response of aircraft's wing in a wind tunnel or during the flight tests.

Introduction

1.1. Tendency in aeronautics

With the constant progress done in materials the challenges that face the modern airplanes to stay competitive are always more complex. The modern wings are made in composite materials which are much lighter allow bigger aspect ratio. In consequence the wings are much more flexible and deflect much more than the precedents generation. Military research is pushing in this direction to improve performance of the high altitude long endurance aircrafts (HALE). These airplanes need to stay light with very long and thin wing to maximize the efficiency at very high altitude. To improve this airplane there is a need of nonlinear structural model coupled with nonlinear CFD [3]. The last commercial airplane like the A350 from Airbus or the Boeing 787 show also high deflection in the wings.



Figure 1.1: Image of the deflection for the Boeing 787

The tendency also have economical motivations for the commercial airplane: lighter and bigger wing will reduce the flying cost and the flexibility improve the passenger comfort during the flight. In consequence there will be increased research and development in this area and the complexity

they bring with.

1.2. Exacerbated problems with high aspect flexible wings

Although these new wings allow better aerodynamic efficiency, they complicate others issues especially aeroelasticity phenomenons. The divergence, control surface reversal need to be controlled. The wings are also swept to increase the cruise velocity but raised difficulties in calculation for the torsion bending when large deflection are involved. With high aspect flexible wings the displacement is too big to get precise models with linear deformation. The aeroelasticity models used are done with nonlinear model like the Dowell–Hodges beam theory [2] and [9].

It also create instabilities in the control of the airplane, like for example the latero-directional stability or Pilot-induced oscillations. For modern flexible airplane a model of rigid airplane is not accurate anymore. It is needed to represent the elastic deflection due to the effects of aeroelasticity and propulsive forces. These phenomenons need to be predictable to be able to conceive efficient control system for these airplanes [7].

1.3. Laboratory scale testing

This new fields are actively researched and numerical tools are developed to estimate these responses. These numerical tools need validation from experimental results. Some studies compare numerical and experimental result to validate the computational model like [11] and [10]. However the data extracts from this experimental results is limitated. It is complex to install many captor on the wing and it could change the wing response. It usually is just the displacement at the tip of the wing and not the displacement field of the wing. In this more complex models, compare the experimental displacement field with the numerical one would help to validates these codes.

1.4. The project

The goal of this research is to develop and test methodology and tools to measure displacement of a flexible wing using a unique camera at the root. The image processing of the pictures allow to get the total displacement of the wing and not just in some isolated points. Using image processing to calculate the wing displacement have already been done by Albertani(2007) [1]. They used two cameras to measure the flexible wings displacement from micro airplane using visual image correlation (VIC). This need two camera taking pictures at the same time from two different point of view which is not practical in a wind tunnel. This multiple camera system is also tested by Park [8] in a structure.

A unique camera is used in some construction to monitor the displacement of constructions and especially bridge [4] [6]. They use a chessboard drawing on the structure and by taking picture of this drawing from an immobile camera you can track the structure displacement by identifying the edge of the chessboard drawing.

In addition another study measure the displacement of a flexible membrane by tracking points drawn on the surface [5].

The concept which I will develop during this study is to measure the displacement of a flexible wing using a camera at the root of the wing. A chessboard motif will be put on the wing which will allow to identify and track a points field on the wing. The displacement of each point will be calculate and will give an approximation of all the displacement field in the wing.

The camera at the root of the wings allow easier installation in a wind tunnel and also this system could be used later on a plane to take picture from the interior to get real time information of the displacement field of the wing.

Experiments

2.1. Experiment description

An aluminum plate was used to simulate the wing. The chessboard pattern was printed on paper sheets and stuck to the plate. The plate was clamped at one extremity as shown in Figure 2.1. On the free side, two lines have been attached at each free plate edges. With this system,

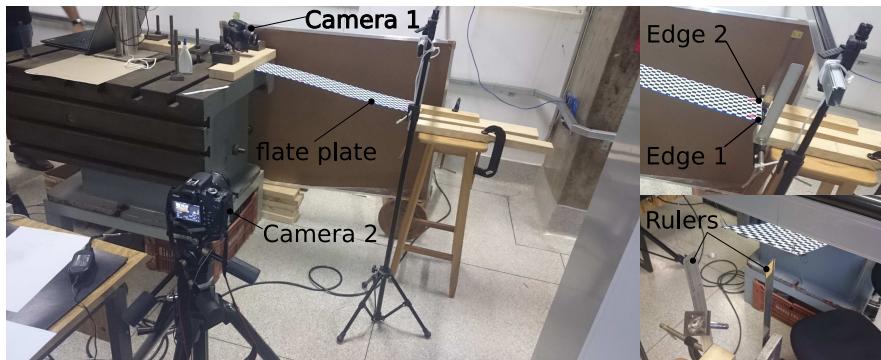


Figure 2.1: Experimental setup using an aluminum flat plate to represent the wing.

different forces on each side can create a flexion and torsion. Two rulers were fixed to measure the displacement at the end of the plate (see Figure 2.1)

To record the plate position, two cameras were used. The first one was located at the root of the flat plate and the second one behind the flat plate. The point of view of the camera 1 allows to see the chessboard motif but not the curvature of the plate. The camera 1 can only use the chessboard pattern to calculate the plate position. Create a program to estimate the displacement field from a picture from this point of view will precisely be investigated on this work (see Figure 2.3).

The camera 2 was set to measure directly the flat plate displacement of the front edge (see Figure 2.3). The displacement field identified via image processing of camera 1 data will be compared with

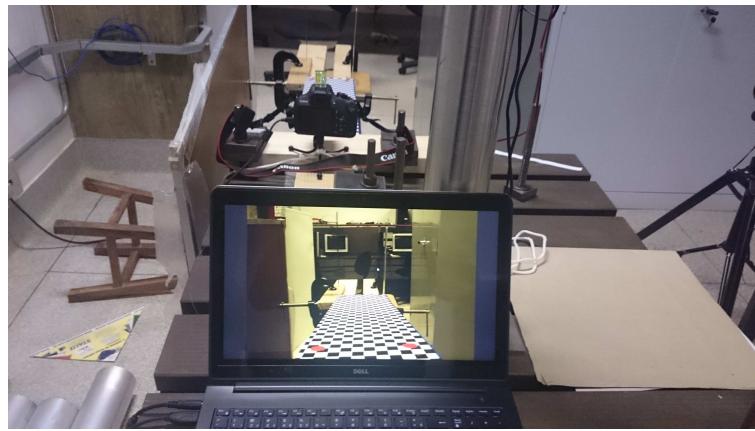


Figure 2.2: Camera 1 point of view.

the measured one by camera 2.



Figure 2.3: Camera 2 point of view.

The cameras specifications are described in the following table 2.1.

Table 2.1: Specification of the cameras for the experiment.

	Camera 1	Camera 2
Model	Canon EOS REBELS T5	Canon EOS REBELS T5
ISO speed	ISO-100	ISO-100
F-stop	f / 11	f / 5
Exposure time [s]	0.6	1.3

2.2. Parameters description

The coordinates (x, y, z) used have their origin at the at the root of the flat plate aligned with the red circle as shown with in the Figure 2.4.

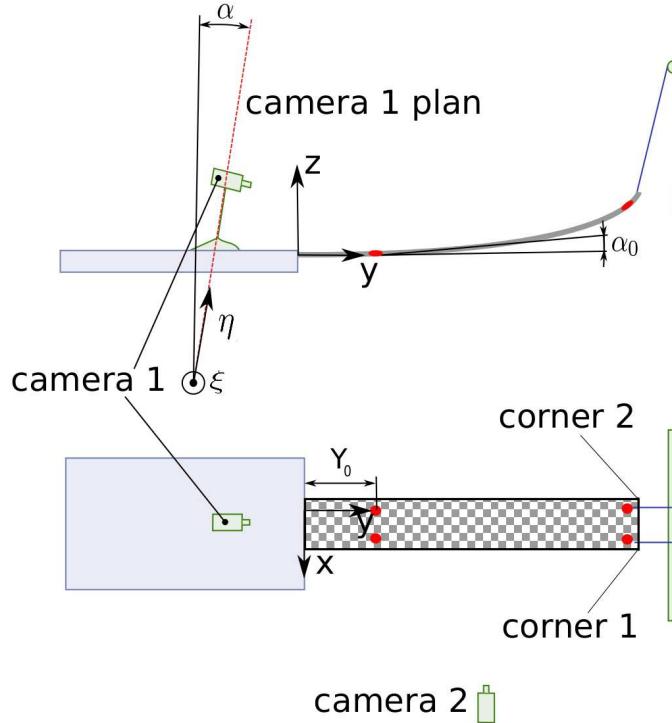


Figure 2.4: Sketch of the frond and lateral view of the set up with the parameters used for the image processing and the inverse problem

The angle α is the angle of the camera plane with the plan (x, z) . On the camera plane, where the image of the flat plat is projected, two coordinates (ξ, η) have been introduced. The angle α_0 is the medium angle between the vector y and the tangent of the plate at the first red disc. The distance Y_0 is the distance between the root of the plate and the first red disc.

Image processing

3.1. Transforming the raw image

From the experiment, it was obtained two images in the format CR2. The software DCRAW allows to treat the raw CR2 image by applying filters (contrast, luminosity, colors, etc...) and then to create a new image in the TIFF format, which is readable in Matlab. In this experiment, the command **dcraw -a -T CR2** was used to convert the image in the format CR2 to the format TIFF using the camera parameters.

To treat the image on a laptop, a new image using 2 by 2 superpixel was used. Each 4 pixels were transformed in 1 superpixel. In consequence, the size of the image was reduced by 4. The new images were around 10 MB which allowed it to load and process it on Matlab. The laptop used Matlab 2013a and is a Samsung R530, which has 32-bit with 4.00 GB RAM and a chip Intel GMA4500 MHD. When the program was functional the high quality images were used to improve the precision of the displacement.

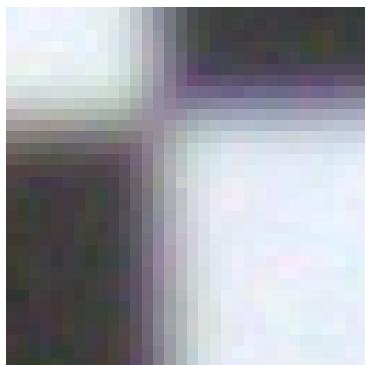


Figure 3.1: sample of 32×32 pixel of an image

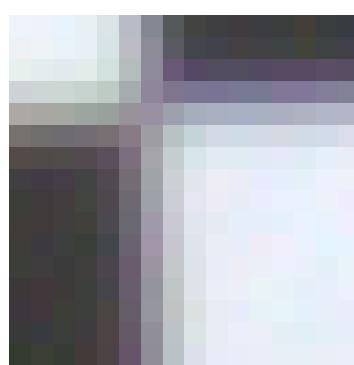


Figure 3.2: Same of 16×16 pixel of the image after applying superpixel

3.2. Region of interest (ROI)

The filtering procedure is illustrated using the same image. Some part of the filtering does not have a considerable influence on one picture but can be much more significant on another one. The total procedure allows to treat the 90% of the picture with the same algorithm. There is no adjustment parameter needed to treat the set of pictures.

From the loaded image, it is possible to identify the red circles on the plate. The raw image is presented on the figure 3.3.



Figure 3.3: Loaded image in Matlab in the TIFF format

By linear combination of the colors, the four circle positions can be easily identified using the *regionprops* function. To identify the red circle, a linear combination of the 3 image color layers enable to obtain the red color.

$$I_{out} = I_{in}^r - 0.5 \cdot I_{in}^g - 0.5 \cdot I_{in}^b \quad (3.1)$$

Since I_{in} is the raw matrix picture matrix represented on figure 3.3. The index r, g and b represent the tree colors layer red, green and blue respectively. I_{out} is the black and white resulting matrix. Then the matrix is filtered keeping only the values which are 10 times the mean of the matrix I_{out} . The resulting binary image is shown in figure 3.4.

To identify the big circles and not the small ones, an average filter is used over 20×20 pixels to delete the noise and only see the 4 big circles from the figure 3.4. With the position of the four red circles a filter which deletes all the information out of the Region Of Interest (ROI) is applied. The picture with the ROI filter is shown in the figure 3.5

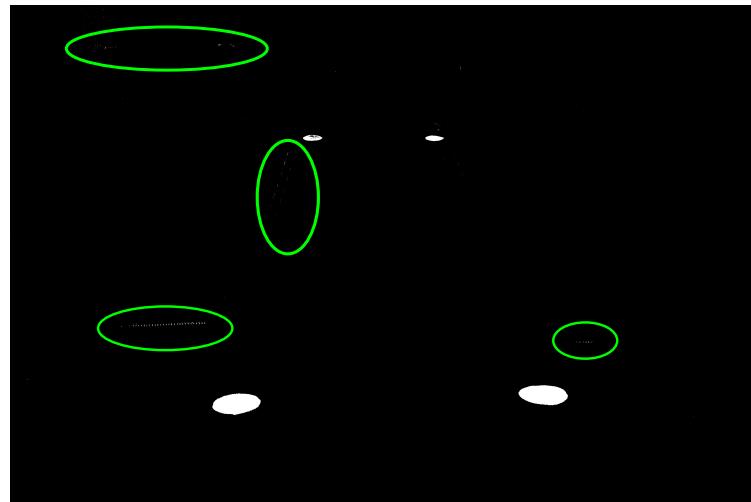


Figure 3.4: Red corner identification image with in the green ellipse noise which will need to be deleted before to identify the circle

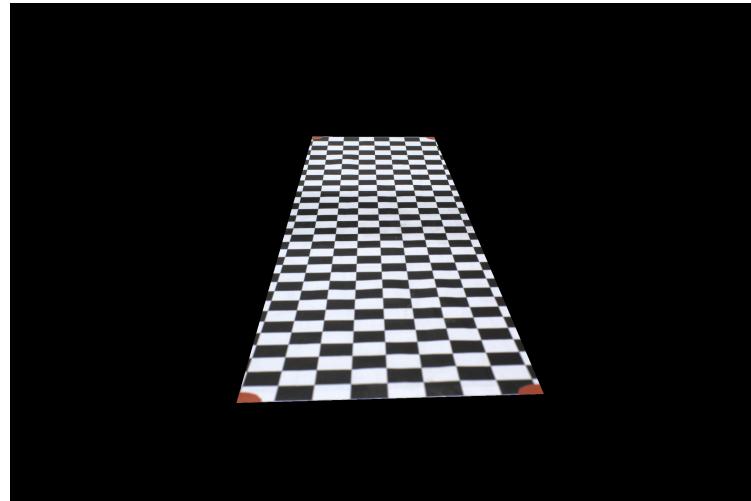


Figure 3.5: ROI

3.3. Corner identification in the chessboard motif

The next step is to identify the corners of the chessboard motif on the plate. First, the previous image (see figure 3.5) is transformed into a binary image (see figure 3.6).

Then an convolution operator is applied to identify the corner. The two following $2n \cdot 2n$ matrix

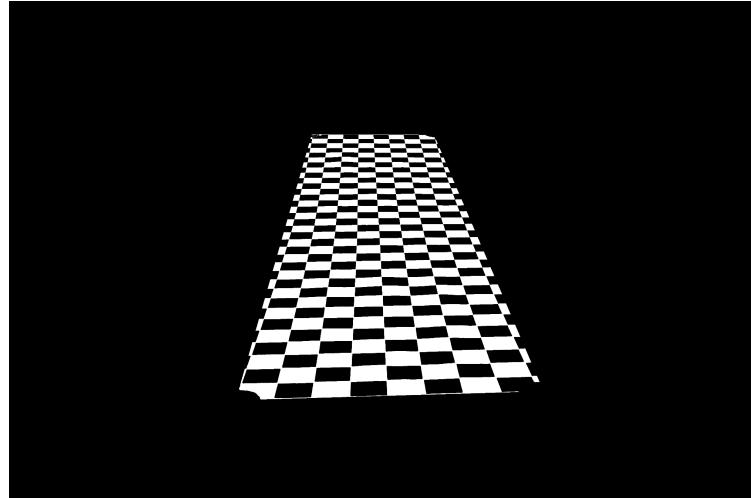


Figure 3.6: Binary image with the mask ready for the corner identification

are defined,

$$A_1 = \begin{pmatrix} -1 & \cdots & -1 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & -1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & -1 & \cdots & -1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & -1 & \cdots & -1 \end{pmatrix} \quad (3.2)$$

$$A_2 = \begin{pmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix} \quad (3.3)$$

For each pixel of the matrix $I_{picture}(\xi, \eta)$ let construct a $2n \times 2n$ matrix which is a piece of the original image and where the pixel of interest is in the middle. Let call this matrix $P(\xi, \eta)$.

Then the following quantity is calculated:

$$I(\xi, \eta) = \mathbf{v}^T [(\mathbf{A}_2 - \mathbf{P}(\xi, \eta)) \circ \mathbf{A}_1] \mathbf{v} \quad (3.4)$$

Where \circ is the Hadamard product operator ¹

If the pixel is in a corner of the type: $\begin{array}{c|c} \text{black} & \text{white} \\ \hline \text{white} & \text{black} \end{array}$ then $I(\xi, \eta)$ will be close to 0. If the pixel is in a corner of the type: $\begin{array}{c|c} \text{white} & \text{black} \\ \hline \text{black} & \text{white} \end{array}$ then $I(\xi, \eta)$ will be close to 1.

Otherwise $I(\xi, \eta)$ will be approximately 0.5.

So $I(\xi, \eta)$ indicates if the image at the coordinate (ξ, η) is on a corner and which kind of corner. The resulting image by doing the operation corner detection 3.4 to the image 3.6 is shown on figure 3.7.

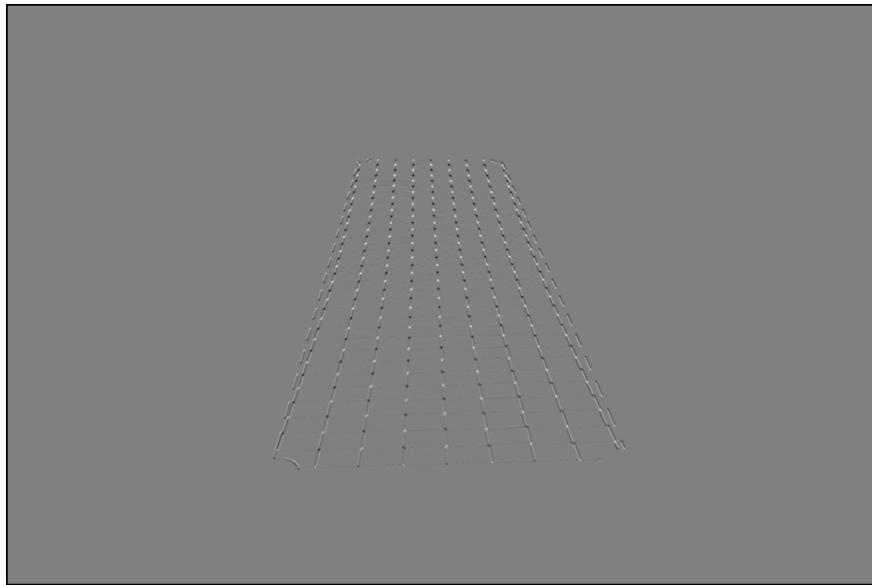


Figure 3.7: Resulting image using the operation corner detection to the black and white image with the ROI mask

From this image 3.7,a band reject filter is applied to the image. Each pixel is updated by the following value:

$$I(\xi, \eta) = \begin{cases} 1, & \text{if } I(\xi, \eta) > 0.6 \text{ or } I(\xi, \eta) < 0.4 \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

From this image ??, it is important to notice that the left and right line of points of the picture have a strong noise due to the filter. This creates a lot of points which are hard to treat afterwards if they are not deleted now. To avoid this problem, the first and last column are deleted. To delete these points, the center points of the red circle are used again. The ξ coordinates or horizontal coordinates from these 4 points are brought closer from the center of the image. A new mask is

¹the Hadamard product operator is defined as $A \circ B \equiv A_{ij} \cdot B_{ij}$

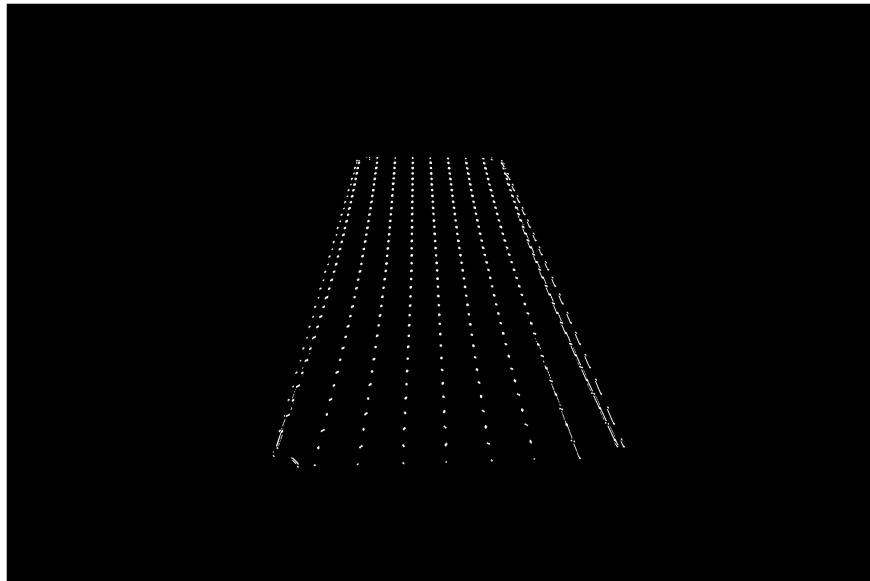


Figure 3.8: Binary filter from the image computed from the corner detection

applied to the image using this new four points which allow to delete most the noisy signal at the border and the first and last column. The resulting image is shown in the figure 3.9.

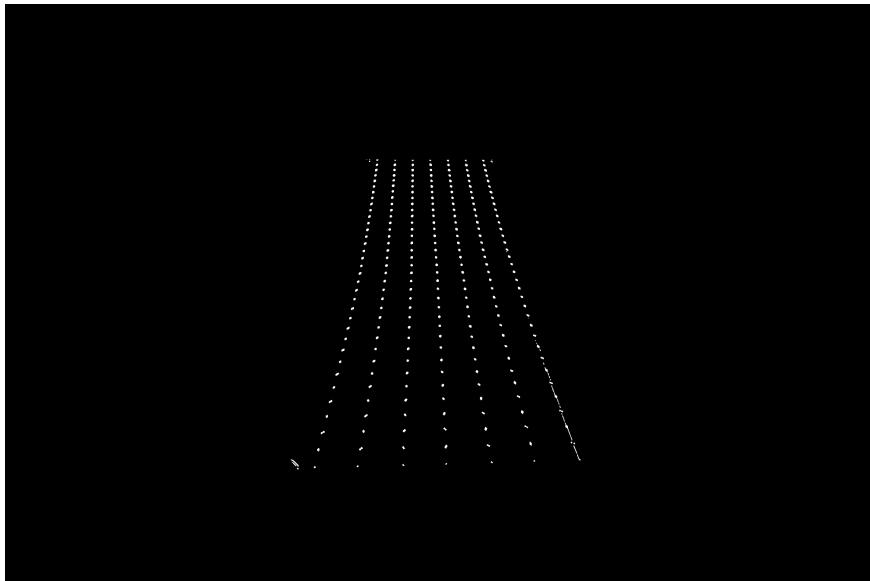


Figure 3.9: Binary picture after the mask to delete the first and last column

Then using the *regionprops* function on the corner recognition image 3.9 for the white dots, the coordinates (ξ, η) of all the corners are obtained.

Some noise is present which creates wrong points as shown in the figure 3.10.

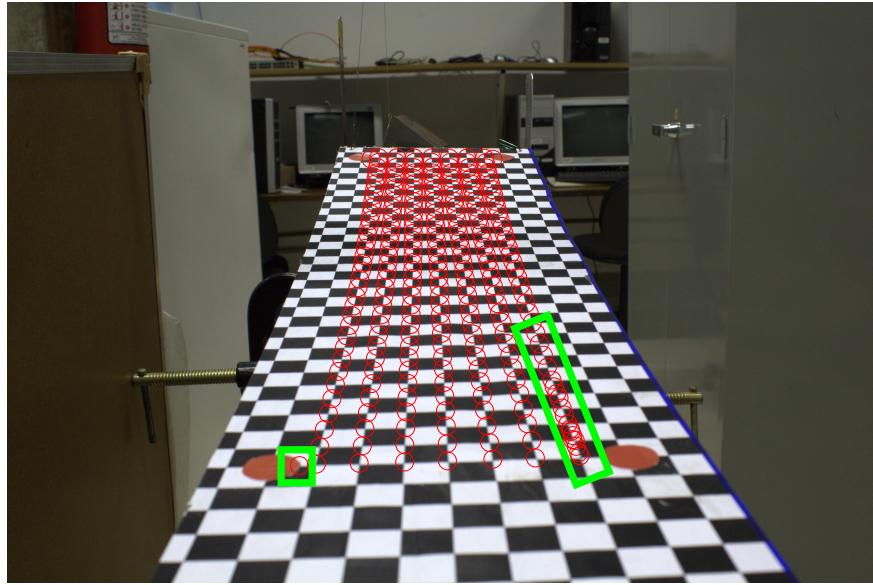


Figure 3.10: Final points from the corner detection with in the green rectangles inaccurate points which will need to be filtered

3.4. Filter and order the points

Once the points are identified, they need to be reordered and filtered. Indeed they are not ordered by line as we want but by the abscissa ξ as shown in figure 3.11.

To order these points, the following algorithm called `sort_point` have been developed. From the origin point, the closest point is calculated and register the angle between the horizontal axis and the vector formed between the first point and the closest point. Then the origin point is replaced with the closest point and delete the closest point from the not sorted points list. The angle corresponding at the point A_n is the angle formed by the vector $\overrightarrow{A_{n-1}A_n}$ and the vertical axis. When the angle is big enough then the algorithm detects that the closest point is not on the same line and the algorithm can go to another line. The result of this algorithm is shown on the figure 3.12.

Then the algorithm `deleteclosepoints` regroups the points which are closer compare to their neighbors. It allows to regroup corners which have been separated in two different circle during the corner detection process. It happen when the chessboard motif is twisted compared to the (ξ, η) axis. However even if it can regroup points together is cannot delete group of inaccurate points which are close one to each other to 'protect themselves' from the algorithm like in the start of the top right column in figure ???. The updated points are shown on the figure 3.13.

Then the new set of points are reordered using the same algorithm `sort_point`. The figure ??

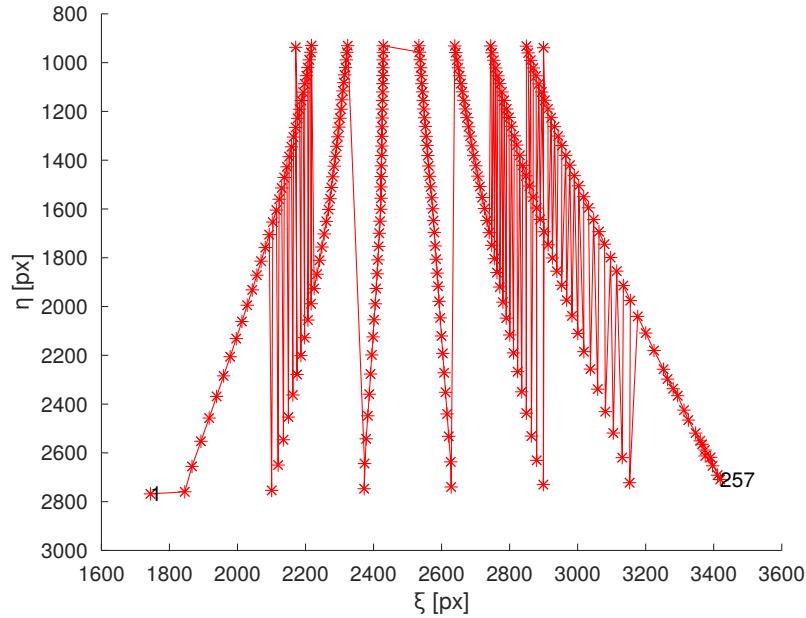


Figure 3.11: Points from the corner detection with the Matlab original ordering

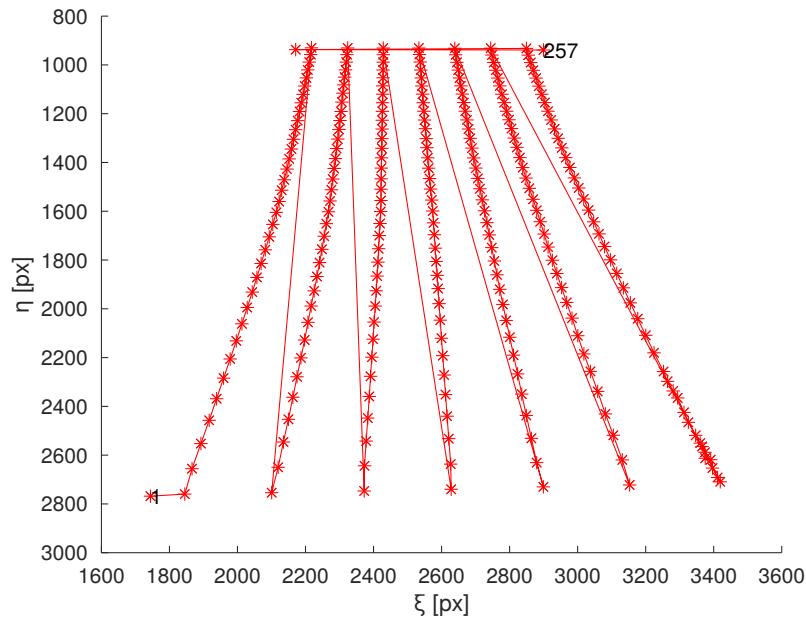


Figure 3.12: Ordered points using the algorithm `sort_point`

show the resulting point ordering.

Using the angle, the algorithm `deletepoints` determines when a column starts and ends. Moreover,

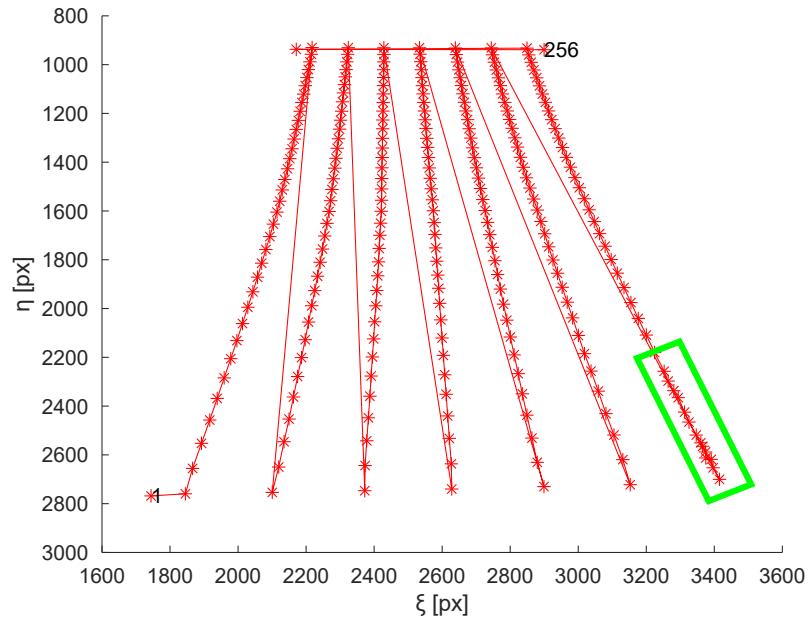


Figure 3.13: Points after the deleting the close points and in the green rectangle inaccurate points which are not deleted from the algorithm *deleteclosepoints*

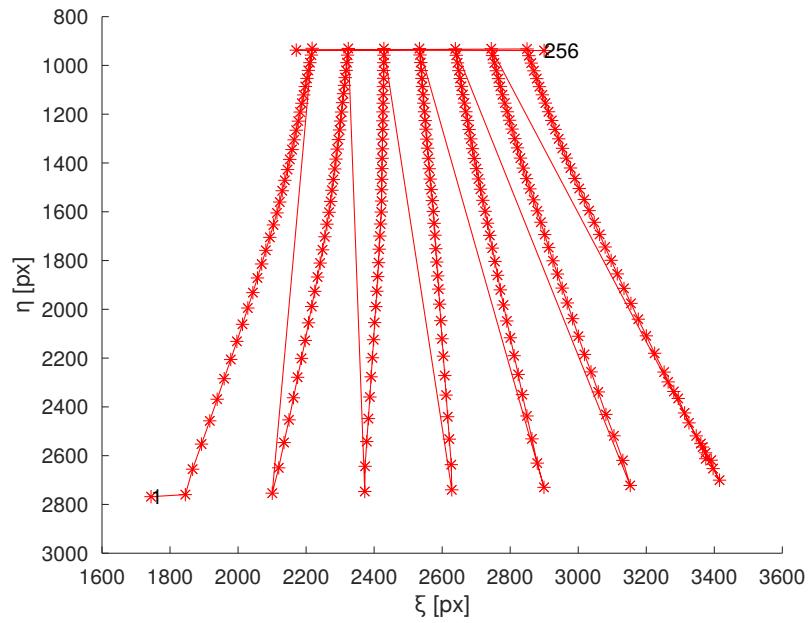


Figure 3.14: Points reordered using for the second time *sort_point*

Figure 3.15: fig:sort3

it helps to identify irregularities. For example, lines of only 5 or less elements should not be relevant and are deleted.

The resulting points are shown in the figure 3.16.

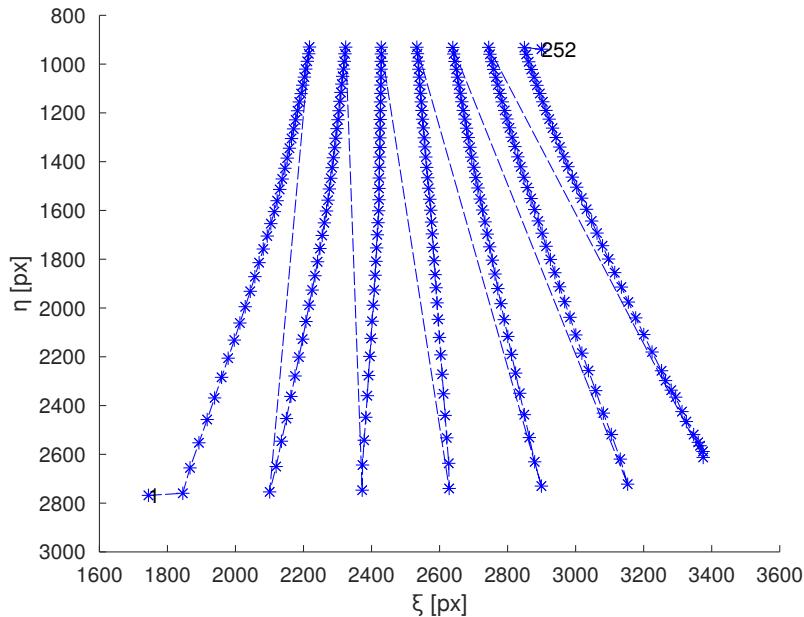


Figure 3.16: Points after the algorithm *deletepoints* which delete points outside the main columns

It permits to break the sorted points in 7 different columns which are clearly identify but could not have the same number of points as can be seen in figure 3.18.

3.5. Interpolate the missing points

Now 7 columns of points need to be filtered. Indeed, each column cannot have the same number of point. Some points can be missing on a column or it can have too many points due to noise.

To solve this problem a function called *fillMissingPoints* is used. For each line a linear regression is done. If some points are far from the interpolate line that mean one point is missing on the line.

In this case, the algorithm take away the point which are the farthest from the linear regression. It does a new regression with just these new points.

Then, the program does also a polynomial regression on the points from the column. It is possible to estimate the missing point looking for the intersection of the two regressions.

The estimated missing point is added to the column.

Also, the algorithm allows to delete the noise points. then the points of the column which have an ordinate η higher than the added points on the column are noise points which can be deleted.

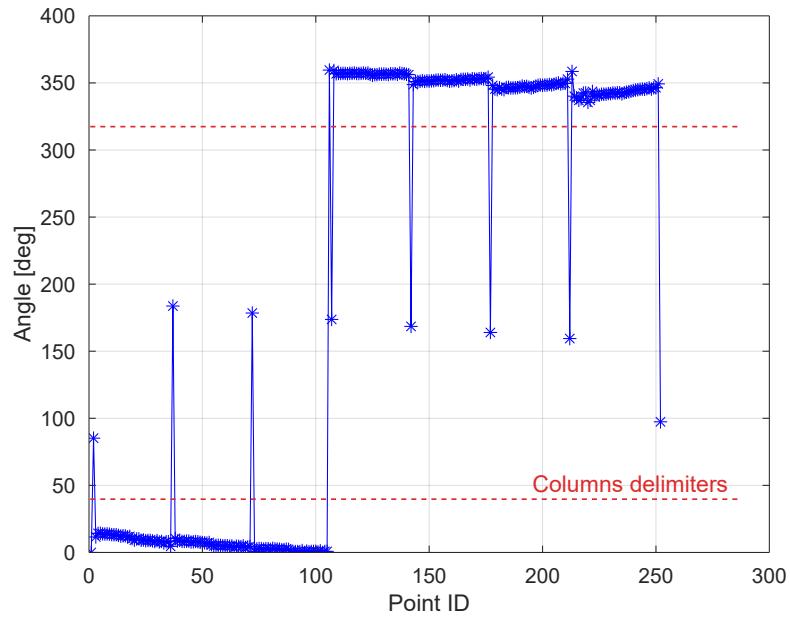


Figure 3.17: Angle for all the points generated from the function `sort_point`

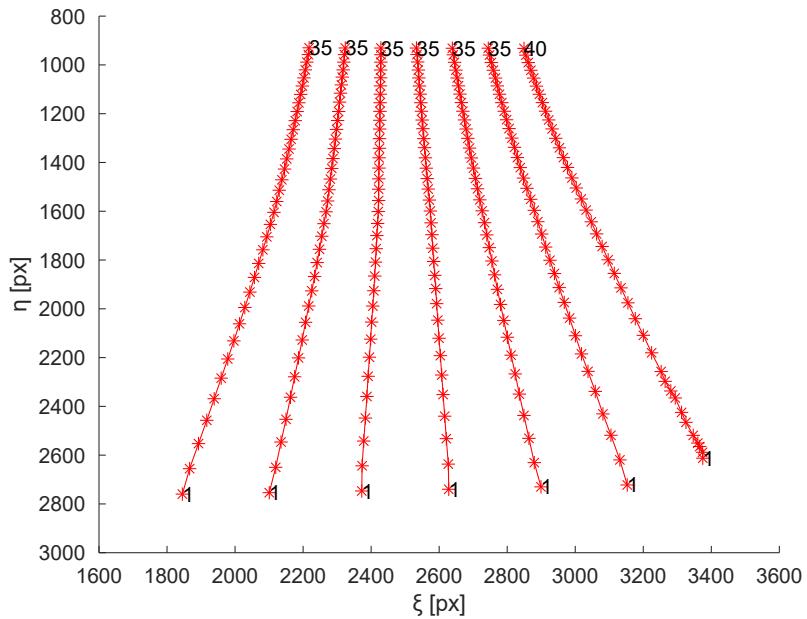


Figure 3.18: Points ordered by column using the angles estimated by the function `sort_point`

These deleted points can be seen on the top right column in red on the figure 3.19.

This algorithm allows to treat correctly 90% the images. The successfully treated images have 7

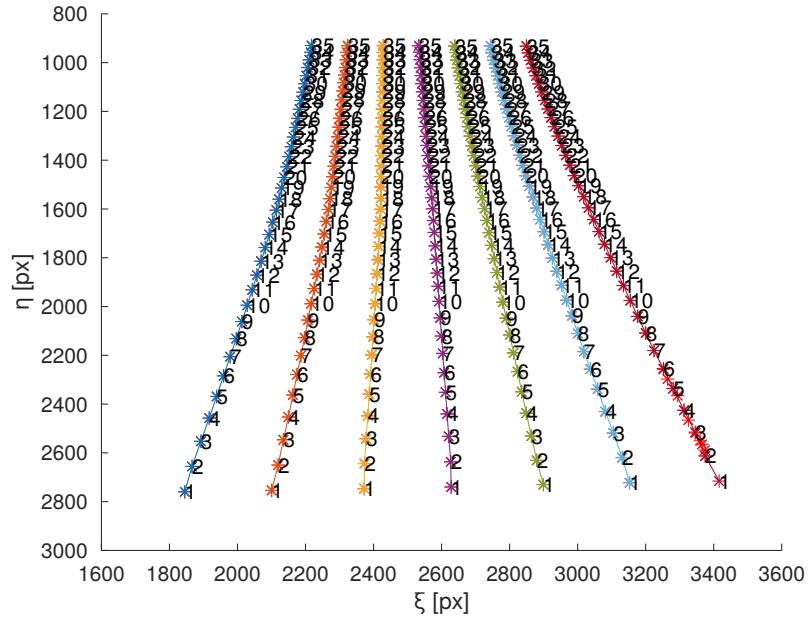


Figure 3.19: Points after the algorithm interpolating the missing points and deleting the noise points

columns of 35 points. The non-successfully treated images are the images with the lowest displacement. The projection of the chessboard square on the camera are really thin and the algorithm have difficulties to identify the corner in this cases. If too many corners are missing the algorithm cannot handle it.

3.6. Smoothing of the points

To smooth the points distribution and try do eliminate some errors on the points a smoothing is applied to the points. For each line, the algorithm computes a linear regression and then replaces each point by line by their interpolation. For each column, the algorithm computes a polynomial regression and then replaces each point by column by their polynomial interpolation.

3.7. Identification problem

Once all the points are identify the problem needs to be invert, i.e. from the projection on the camera in the coordinate (ξ, η) , identify the 3D coordinates (x, y, z) of the flexible plate.

In order to do that the algorithm goes line by line. The first line is at a distance h in the direction y and Z_0 in the direction z . From this first line the next line will be calculated and so on. The size of each square is 15 by 15 mm. For each point, using vertical distance between 2 points

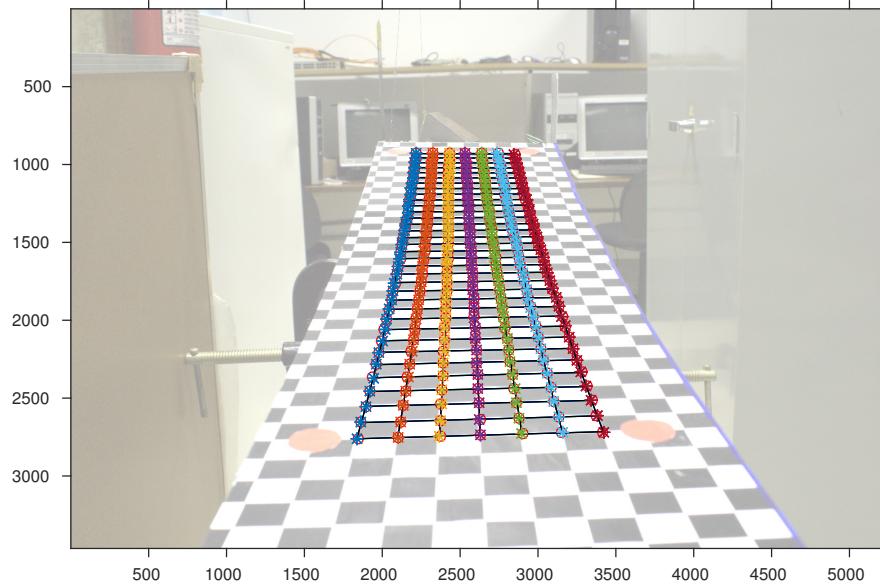


Figure 3.20: Final filtered points with the image in background

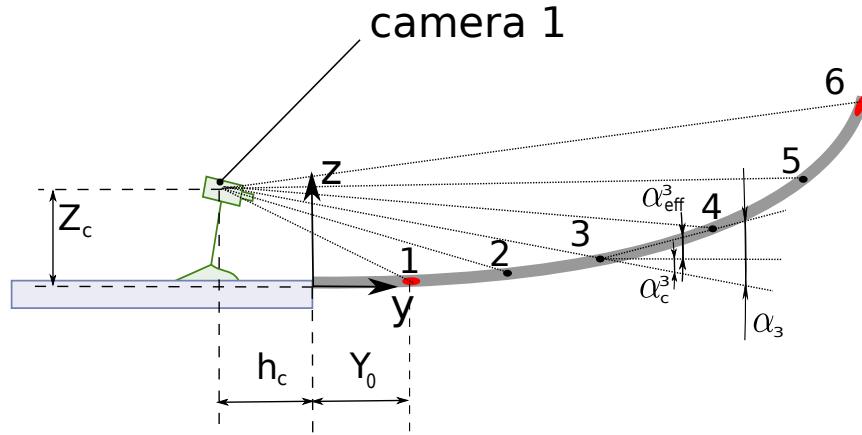


Figure 3.21: Sketch of the angles used to solve the invert problem

horizontally, local length scale d^i in pixels is obtained. Then the angle α^i can be calculated using the scale. The angle is given by the formula:

$$\alpha^i = \arcsin \left(\frac{\| \vec{A_i A_{i+1}} \|}{d^i} \right) \quad (3.6)$$

With the camera position, it is possible to evaluate the angle α_c^i . From this two angles, the

angle α_{eff}^i is calculated. It will give the increment in the direction y and z for the points $i + 1$ using the cosines and sinus formula.

Results

4.1. Comparison of the tip displacement

Using the procedure described before, the displacement of all the plane located in between the red points can be estimated. The displacement can be generated for 8 different cases using the average of two pictures. It is possible to compare the displacement at the tip of the plate with the measured values from the rulers during the experiment.

To calibrate the program, two experiments are necessary to delete the shift between the measured and calculated displacement and to determine a factor of correction for the calculated displacement. To do that the bigger and smaller displacement experiments are used and not considered in the following section for the calculation of the error.

For the invert problem, it is required to give four input:

Y_0 horizontal ordinate of the starting point.

Z_0 vertical ordinate of the starting point.

Y_c horizontal ordinate of the camera.

Z_c vertical ordinate of the camera.

Using the camera 2 it is possible to estimate the distance Z_0 of the flat plate at the first points for each case. These others parameters have been measured during the experiment or estimated from pictures of the setup.

The results of the tip displacement for the 6 cases are shown on the table 4.1.

Case	Displacement - Corner 1		Error [mm]	Displacement - Corner 2		Error [mm]
	Measured	Identified		Measured	Identified	
1	179.00	176.78	2.72	183.50	183.37	0.63
2	183.50	181.77	2.23	186.50	188.53	2.53
3	170.00	167.59	2.91	173.00	173.47	0.97
4	105.50	104.33	1.67	107.00	108.31	1.81
5	87.50	84.12	3.88	86.00	87.71	2.21
6	65.00	65.09	0.60	68.00	67.55	0.95

Table 4.1: Comparison between measured and estimated result where corner 1 is closest to the camera 2 and corner 2 is the farthest

4.2. Verification of the global displacement

One of the great advantage of this processing is that it gives the displacement for all displacement of all the flat plate. Using the camera 2 the displacement of the border 1 of the flat plate can be estimated using image processing.

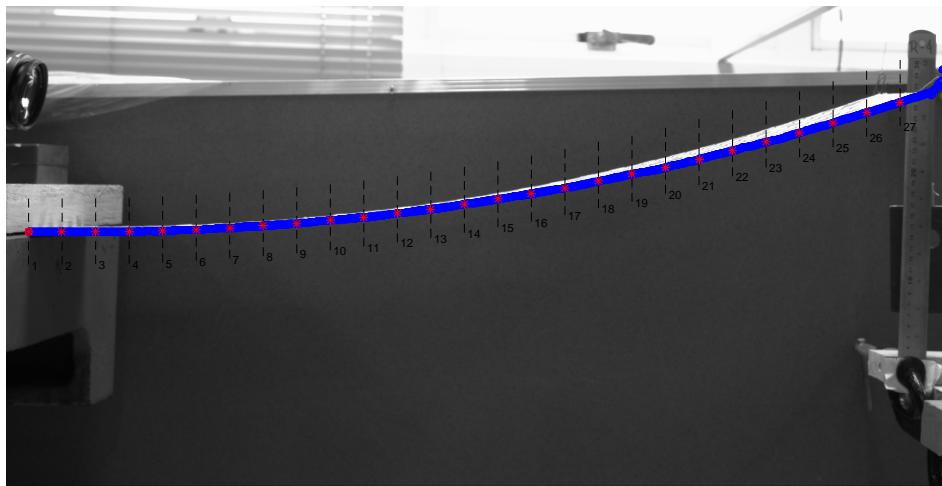


Figure 4.1: Estimation of flat plate border displacement using image from camera 2

Most of the identify points are following the flat plate border. But some points can be wrong at the tip of the plate. This is due to the background objects which generate a noise as shown in the picture 4.2.

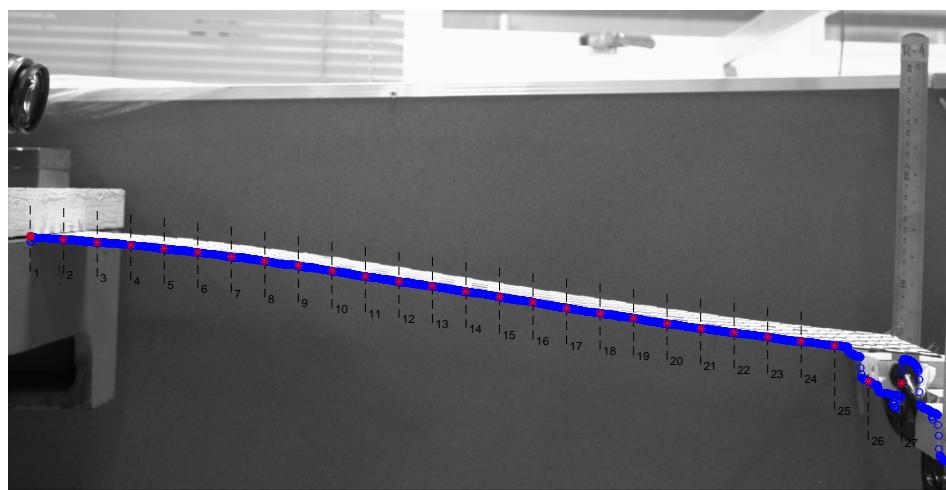


Figure 4.2: Estimation of flat plate border displacement using image from camera 2 with noisy background

To filter the result the resulting points are approximated by a polynomial function. If a point is too far from the polynomial approximation function, then the point should be inaccurate. A new estimation of the displacement is make a polynomial approximation of the others points and evaluate at this abscissa. The result are shown in the figure 4.3.

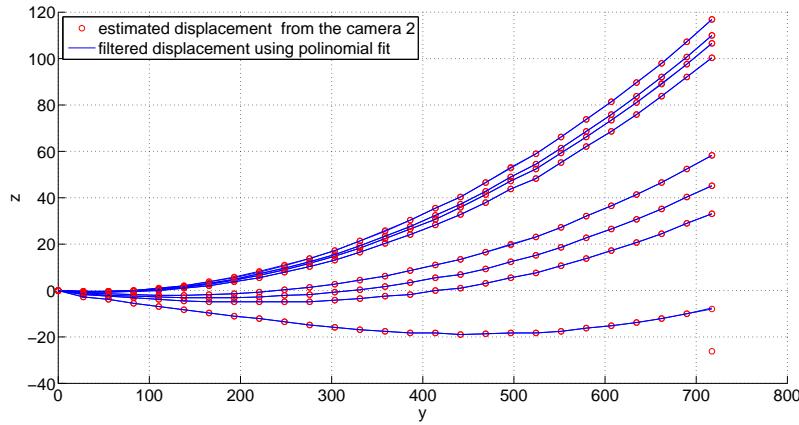


Figure 4.3: Flat plate border displacement estimation from camera 2 and corrected border displacement using polynomial approximation

4.2.1. Full displacement calibration

The present processing is done using the two pictures of the eight cases. From this 16 pictures an average is done to get the average for the eight cases. Then, the bigger and lower displacements are used to calibrate the image processing. The results for the 6 others cases are presented on the picture 4.4.

This methodology gives excellent precision. The maximum displacement error between the figures is less than 0.6 mm. This calibration allows to get the displacement field without a lens calibration. However, this method need the full displacement field of two different test.

4.2.2. Edge displacement calibration

The full displacement calibration might be complicate to be implemented in practice. Another methodology has been developed to use only the final displacement to calibrate the system. The result for the 6 others cases are presented on the picture 4.5.

The edge displacement calibration is need no lens calibration. This methodology needs only the edge displacement of two tests. However, the displacement error is bigger compared to the other methodology. A well-defined kinematic model need to be supposed.

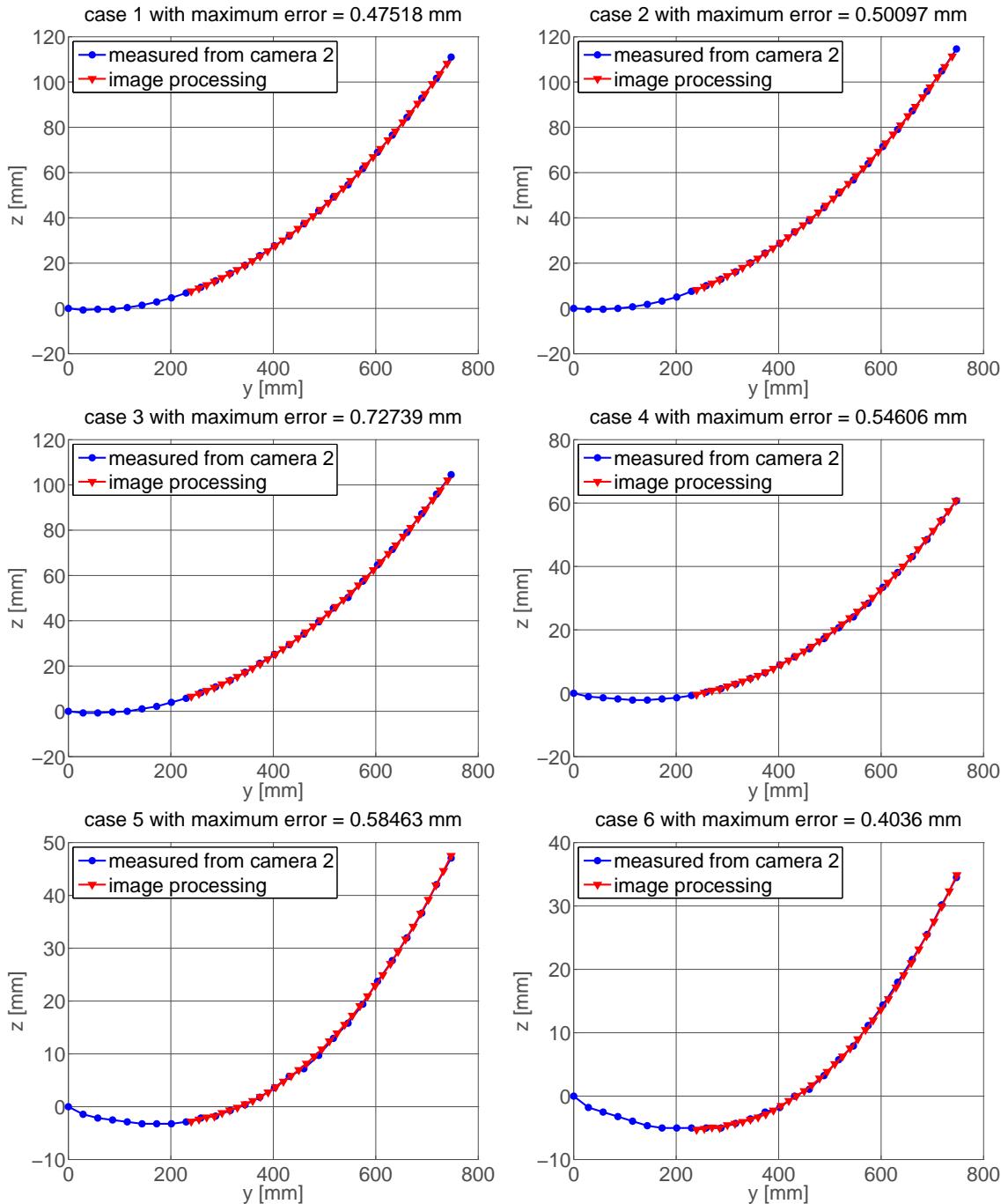


Figure 4.4: Displacement profile for the 6 cases of the front border from the image processing with a full displacement calibration and the measurement from camera 2

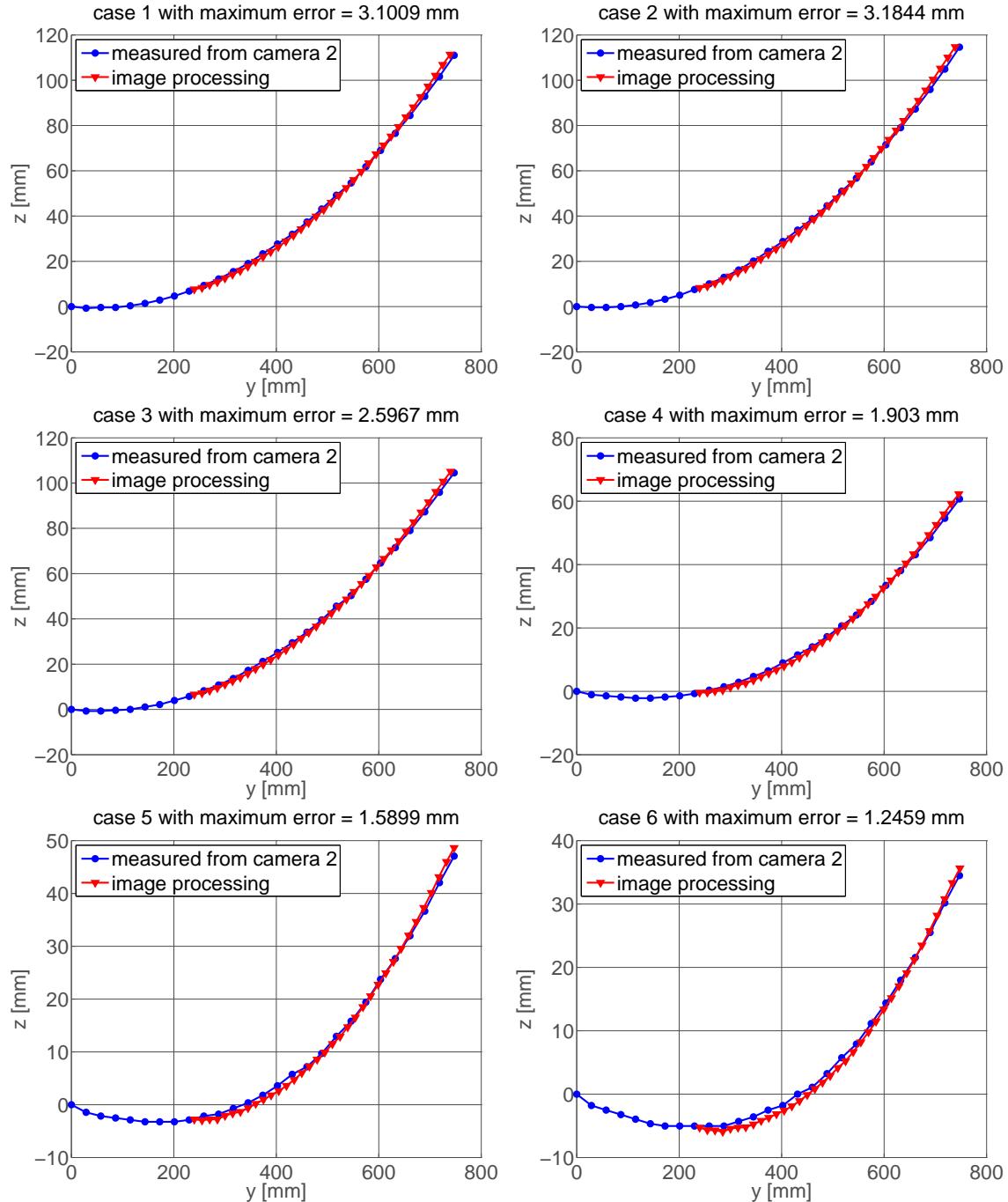


Figure 4.5: Displacement profile for the 6 cases of the front border from the image processing with a tip displacement calibration and the measurement from camera 2

4.2.3. *Without calibration*

Even a tip calibration can be hard to do in some situations. The results for the 6 previous cases are represented without any calibration. The results are shown on the figure 4.6.

Without any calibration the maximum error increases a little bit. The error might be decrease using a lens calibration and measuring with precision the distance Z_c and Y_c .

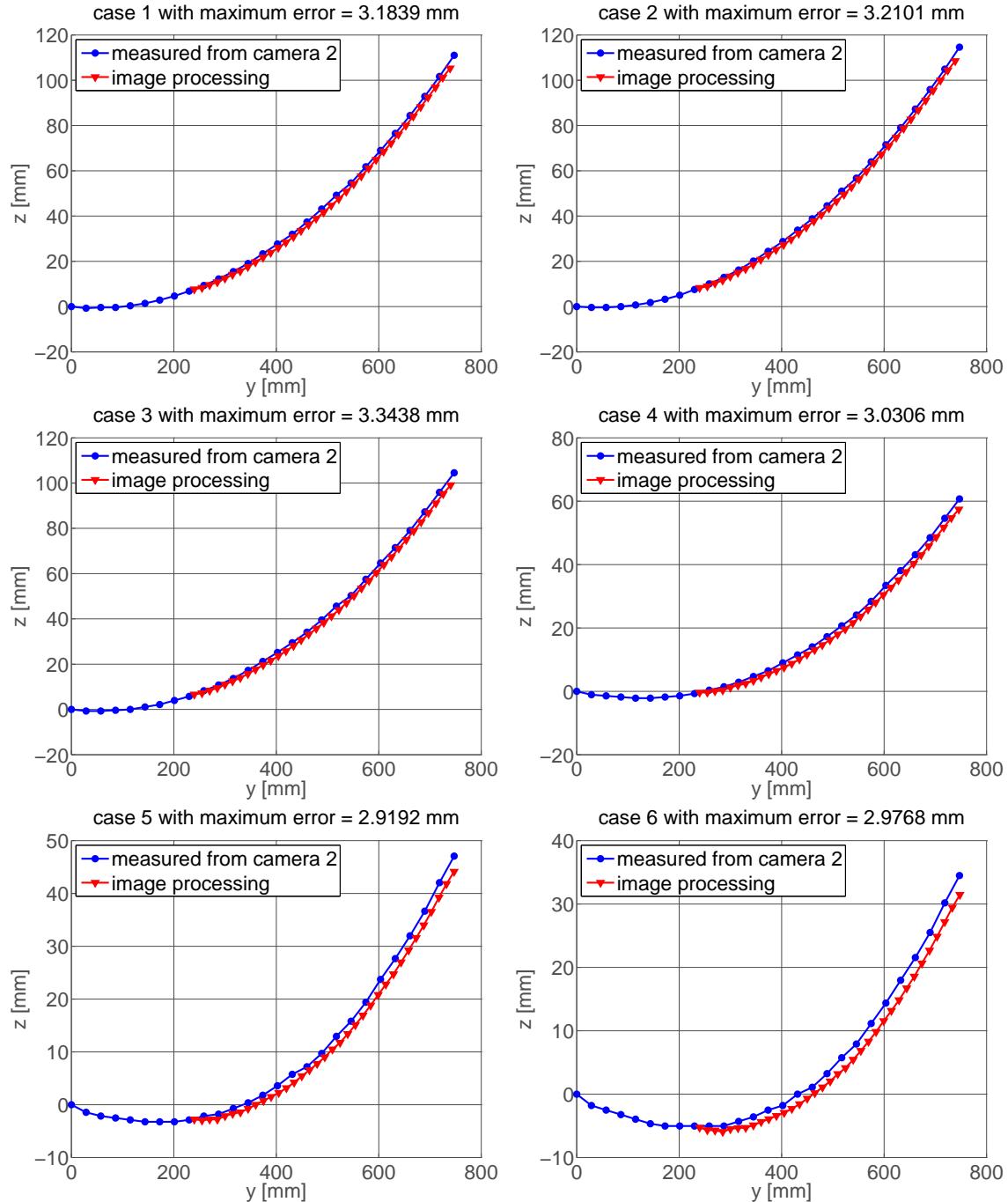


Figure 4.6: Displacement profile for the 6 cases of the front border from the image processing without calibration and the measurement from camera 2

Final remarks

5.1. Discussion of the results

This graduation project makes the proof that it is possible to measure the displacement of a flat plate using only one camera at the root of the flat plate. A first method needs two tests with the full displacement field to calibrate the displacement to get a high precision. With two tests and the tip displacements associated, it also possible to calibrate the estimated displacement field but with losing in accuracy. Without any calibration, the displacement field is still well approximated but present a lost in accuracy. However, this last methodology is easier to install and make it run.

5.2. Future work

The effect of the lens distortion on the might alter the result. It should be verified if it can play a significant role and if it is possible to eliminate it. If this lens distortion can be controlled, it would open the possibility to others camera with a lower quality lens system.

The test has been done with a flat plate. Maybe an airfoil profile will need to complement the code to take into account the thickness difference in the profile. This methodology will need to be tested with a wing profile.

Then the facility could be used to measure the displacement field in a wind tunnel. The camera at the root of the wings will allow easier installation in a wind tunnel and give the full displacement field of the wing.

5.3. Future application

This facility could be used to measure the displacement field in a wing tunnel but also can be easily adapt in an airplane. The camera will be install inside the fuselage and could give real time

information of the wing displacement field.

If this method is able to work with lower quality lens system, it is also possible to get implement this system on a small airplane. For example, a small camera from a Go-Pro could give the wing displacement field in radio commanded airplane which cannot fly with an expensive and heavy camera and lens equipment.

Bibliography

- [1] R. Albertani, B. Stanford, J. P. Hubner, and P. G. Ifju. Aerodynamic coefficients and deformation measurements on flexible micro air vehicle wings. *Experimental Mechanics*, 47(5):625–635, 2007.
- [2] K. Eskandary, M. Dardel, M. H. Pashaei, and A. K. Moosavi. Nonlinear aeroelastic analysis of high-aspect-ratio wings in low subsonic flow. *Acta Astronautica*, 70:6–22, 2012.
- [3] Joseph A Garcia. Numerical Investigation of Nonlinear Aeroelastic Effects on Flexible High-Aspect-Ratio Wings. *Journal of Aircraft*, 42(4):1025–1036, 2005.
- [4] Yunfeng Ji. A computer vision-based approach for structural displacement measurement. *Aerospace*, 7647:76473H–76473H–11, 2010.
- [5] Daniel Leonardo Braga Rodriguez Jurjo, Carlos Magluta, Ney Roitman, and Paulo Batista Gonçalves. Analysis of the structural behavior of a membrane using digital image processing. *Mechanical Systems and Signal Processing*, 54:394–404, 2015.
- [6] J. J. Lee and M. Shinozuka. Real-Time Displacement Measurement of a Flexible Bridge Using Digital Image Processing Techniques. *Experimental Mechanics*, 46:105–114, 2006.
- [7] Nguyen and I Tuzcu. Flight dynamics of flexible aircraft with aeroelastic and inertial force interactions. *American Institute of Aeronautics and Astronautics*, pages 1–23, 1990.
- [8] S.W. Park, Hyo Seon Park, J.H. Kim, and Hojjat Adeli. 3D displacement measurement model for health monitoring of structures using a motion capture system. *Measurement*, 59:352–362, 2015.
- [9] M J Smith, M J Patil, and D H Hodges. CFD-Based analysis of NONLINEAR AEROELASTIC BEHAVIOR OF HIGH-ASPECT RATIO WINGS. (c), 2001.
- [10] D. Tang and E. H. Dowell. Experimental and theoretical study of gust response for high-aspect-ratio wing. *AIAA Journal*, 40(3):419–429, 2002.
- [11] D M Tang and E H Dowell. Experimental and Theoretical Study on Aeroelstic Response of High-Aspect-Ratio Wings. *AIAA Journal*, 39(8):1430–1441, 2001.