
	<p style="text-align: center;">ROSETTA LANDER</p> <p style="text-align: center;">ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 1/16</p>
---	--	--

**RAPPORT DE MESURE DE DISTORSION GEOMETRIQUE DES
CAMERAS CIVA-P QM**

	Name	Function	Date	
Prepared by	N. Charvin	ÇIVA-P Engineer		
Approved by	A. Soufflot	ÇIVA Project Manager		

	<p style="text-align: center;">ROSETTA LANDER</p> <p style="text-align: center;">ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 2/16</p>
---	--	--


1 INTRODUCTION.....3

2 MESURES.....3

3 RÉSULTATS.....4

4 CALIBRATION STÉRÉO5

5 FICHIERS UTILISÉS12

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 3/16</p>
---	--	--

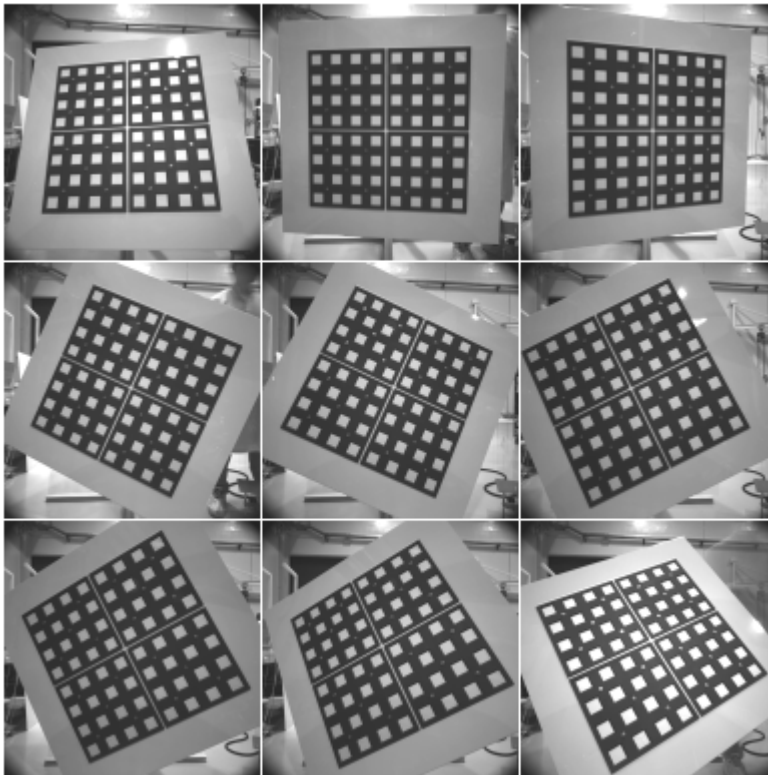
1 INTRODUCTION

Ces tests ont pour objectif de mesurer les paramètres de distorsion géométrique pour les caméras QM. On a utilisé pour cela la méthode décrite dans le document "**Etalonnage des caméras CIVA à l'aide de mires**", rédigé par JL Reynaud, du LAS.

2 MESURES


Les tests ont eu lieu entre le 8/03/2001 et le 14/03/2001, à la station d'étalonnage de l'IAS. Les mesures ont été effectuées à température ambiante.

La méthode utilisée requiert la prise d'une dizaine d'image de mire. La mire en question est constitué d'un réseau de 15x15 carrés blancs, chacun mesurant 55 mm de côtés, et espacé régulièrement de 55 mm, en X et en Y. La mire est articulée, ce qui permet de prendre les images sous différentes orientations angulaires.



Sur chacune de ces images, nous allons détecter les coins des carrés, en utilisant la CALIBRATION TOOLBOX, développée par JY Bouguet, au Caltech (http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)

L'éclairage doit être particulièrement soignée pour permettre une détection précise des motifs de la mire. L'idéal serait bien évidemment un éclairage de face, avec deux spots de chaque côté de la mire. Ceci

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 4/16</p>
---	--	--

n'a pas été possible pendant nos mesures, pour des raisons d'encombrement. Nous avons donc utilisé les plafonniers au sodium, installés dans le hall de la station d'étalonnage.

Après traitement, nous pouvons déterminer les paramètres suivants, pour chacune des caméras:

- La distance focale (ou plutôt le tirage) de l'objectif.
- La position dans l'image de l'intersection de l'axe de visée avec le CCD, appelé *point principal*.
- Les coefficients de distorsion de l'objectif
- Le '*skew factor*', qui représente l'angle entre les axes X et Y des pixels (90° normalement).

Le logiciel permet également de déterminer les positions des caméras stéréos, l'une par rapport à l'autre, en translation et en rotation.

Tous ces résultats sont exprimés dans le chapitre suivant.

Notons que le '*skew factor*' n'a pas été estimé, ce qui explique qu'il est alors spécifié égal à 90°.

3 RESULTATS

3.1 Résultats bruts, issus de la Calibration Toolbox

Les valeurs de distance focale sont exprimées en "dimension de pixels" (soit N*14 microns)


QM04	<p>Focal Length: fc = [889.56696 890.52765] ± [1.97525 1.91913] Principal point: cc = [524.60541 516.99475] ± [1.16378 1.14454] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [-0.01002 -0.01210 0.00042 0.00185 0.00000] ± [0.00343 0.00911 0.00035 0.00035 0.00000] Pixel error: err = [0.28036 0.30450]</p>
------	--

QM05	<p>Focal Length: fc = [882.97825 883.64567] ± [2.28467 2.14883] Principal point: cc = [508.32872 519.47148] ± [0.96449 1.30708] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [-0.00325 -0.02591 0.00045 -0.00122 0.00000] ± [0.00323 0.00854 0.00037 0.00037 0.00000] Pixel error: err = [0.28036 0.30450]</p>
------	---

QM06	<p>Focal Length: fc = [886.17917 886.60248] ± [1.69696 1.56294] Principal point: cc = [510.52243 513.25832] ± [1.00828 1.27773] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [-0.00829 -0.01519 -0.00054 0.00039 0.00000] ± [0.00384 0.01195 0.00039 0.00039 0.00000] Pixel error: err = [0.28036 0.30450]</p>
------	---

QM07	<p>Focal Length: fc = [886.39381 886.02127] ± [1.54838 1.64852] Principal point: cc = [496.43231 512.43151] ± [1.33501 1.00390] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [-0.01152 -0.00237 0.00018 -0.00146 0.00000] ± [0.00324 0.01010 0.00040 0.00039 0.00000] Pixel error: err = [0.28142 0.26253]</p>
------	---

QM09	<p>Focal Length: fc = [887.64757 887.94479] ± [3.13971 3.10558] Principal point: cc = [517.67592 515.98634] ± [1.18781 1.30690]</p>
------	--

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 5/16</p>
---	--	--

<p>Skew: => angle of pixel axes = Distortion: Pixel error:</p>	<p>alpha_c = [0.00000] ± [0.00000] 90.00000 ± 0.00000 degrees kc = [-0.01369 -0.00744 -0.00029 0.00018 0.00000] ± [0.00412 0.01281 0.00039 0.00040 0.00000] err = [0.24205 0.40275]</p>
---	---

<p>QM11</p>	<p>Focal Length: fc = [889.84095 889.60585] ± [1.64101 1.72557] Principal point: cc = [512.87654 524.52184] ± [1.43343 1.11205] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [-0.01179 -0.00226 0.00074 0.00172 0.00000] ± [0.00304 0.00765 0.00043 0.00039 0.00000] Pixel error: err = [0.28945 0.27530]</p>
-------------	--

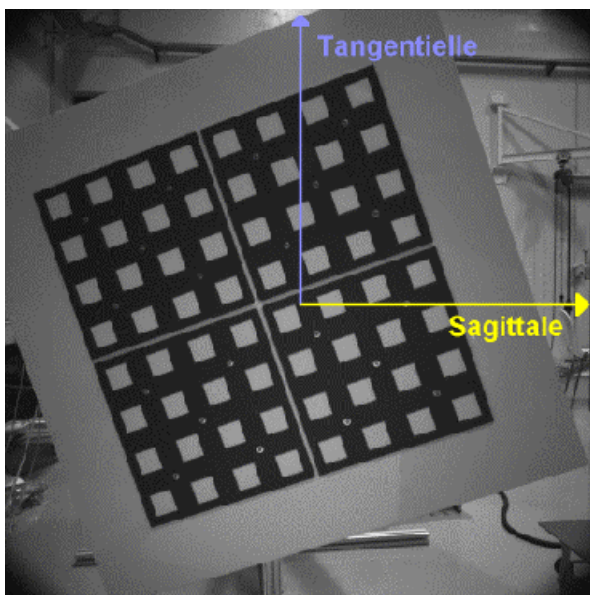
<p>QM12</p>	<p>Focal Length: fc = [890.90033 891.21899] ± [2.79021 2.75571] Principal point: cc = [516.89671 513.96647] ± [1.33156 1.49142] Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees Distortion: kc = [0.00565 -0.05355 -0.00004 0.00049 0.00000] ± [0.00539 0.01864 0.00047 0.00048 0.00000] Pixel error: err = [0.26624 0.46185]</p>
-------------	--

3.2 Résultats interprétés

A l'aide des résultats bruts, et en suivant les notations présentées en annexe, on peut présenter les résultats sous la forme suivante.

3.2.1 Distorsion axiale

Ces figures présentes les distorsions sagittale et tangentielle, pour les différentes caméras.





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

Authors : N. Charvin

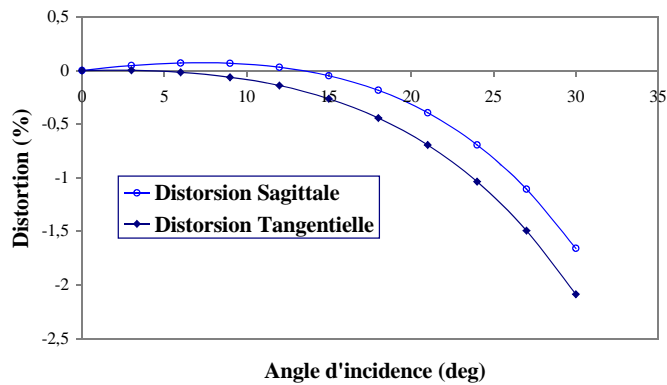
Edition : 0

Revision : 0

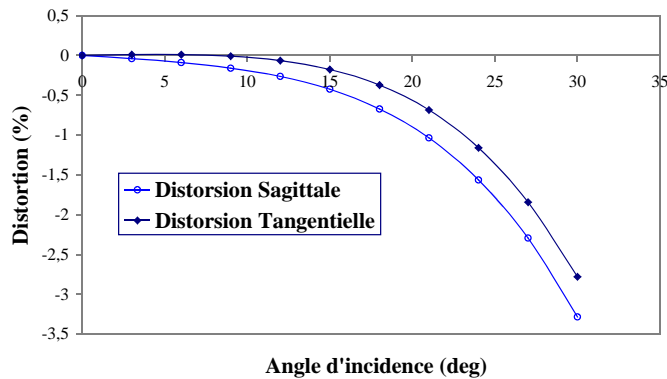
Date Rev. : 12/12/2001

Page : 6/16

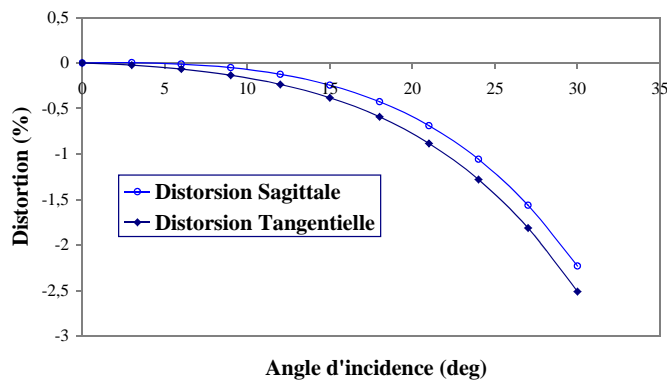
Distorsion QM04



Distorsion QM05



Distorsion QM06





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

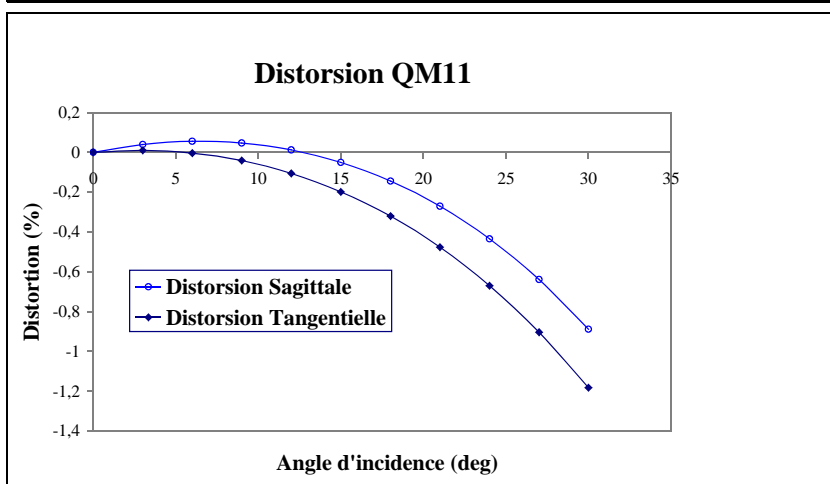
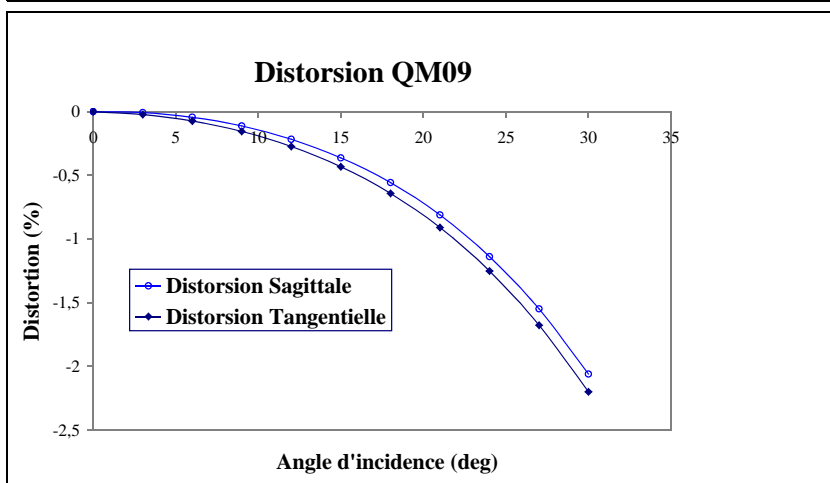
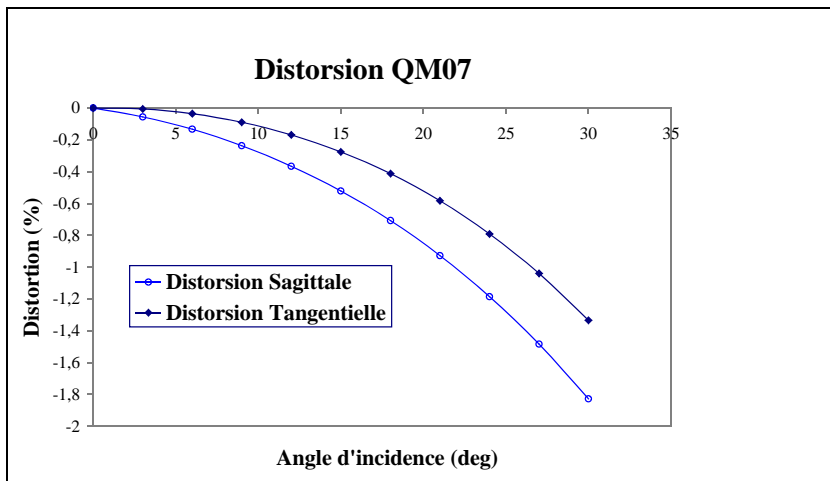
Authors : N. Charvin

Edition : 0

Revision : 0

Date Rev. : 12/12/2001

Page : 7/16





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

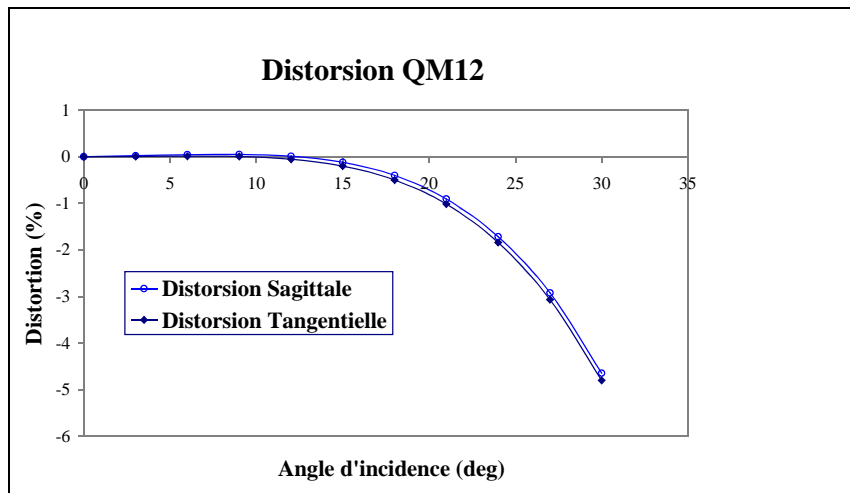
Authors : N. Charvin

Edition : 0

Revision : 0

Date Rev. : 12/12/2001

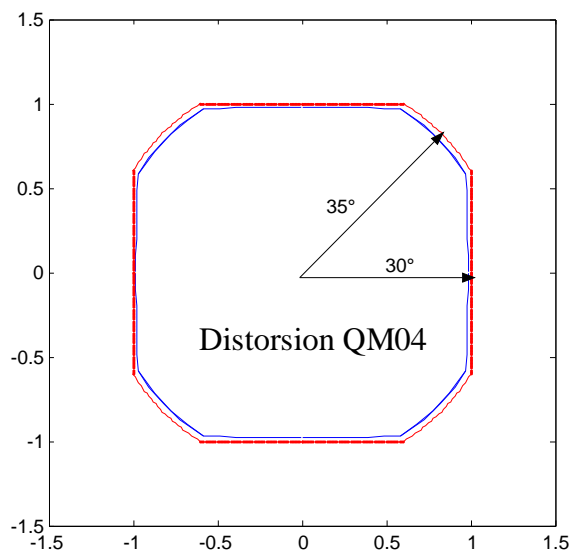
Page : 8/16



3.2.2 *Distorsion sur le plan focal*

Les figure suivantes montrent la carte de distorsion sur le plan focal, pour chacune des caméras. Le champ de vue est dimensionné de la façon suivante : sur l'axe 30° , sur la diagonale, 35° . Il est représenté en rouge sur les figures.

Leurs coordonnées axiales sont normalisées, la valeur 1 correspond au champ de 30° (Angle d'incidence de l'objet). La courbe rouge présente la limite de champ de vue de caméra, et la courbe bleue présente l'image réelle de cette limite. La déformation résulte la distorsion de l'objectif de la caméra.





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

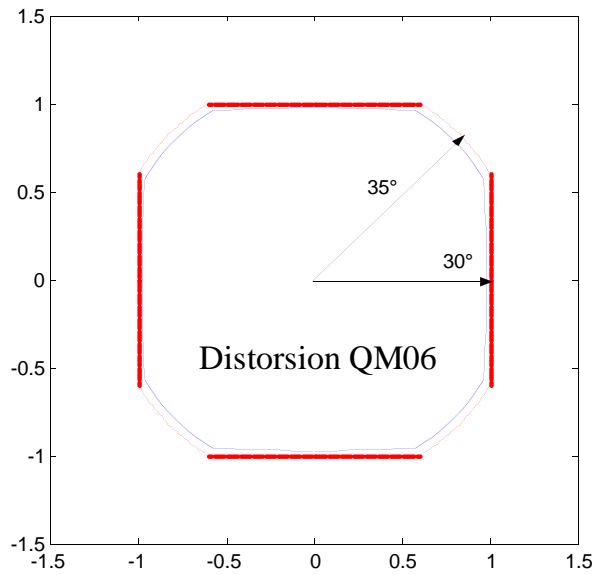
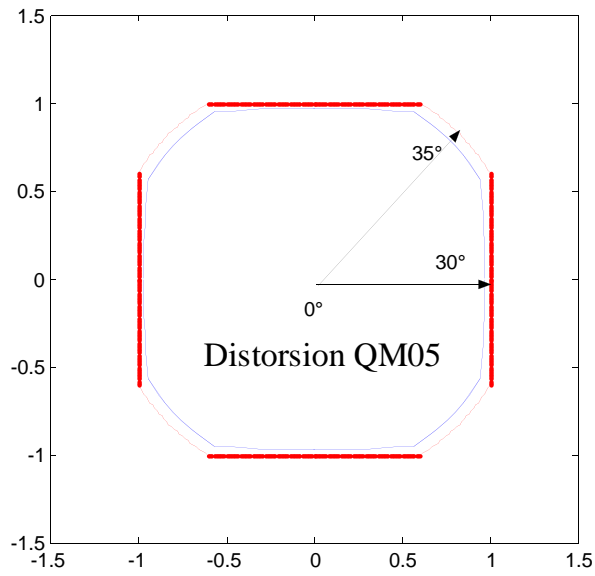
Authors : N. Charvin

Edition : 0

Revision : 0

Date Rev. : 12/12/2001

Page : 9/16





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

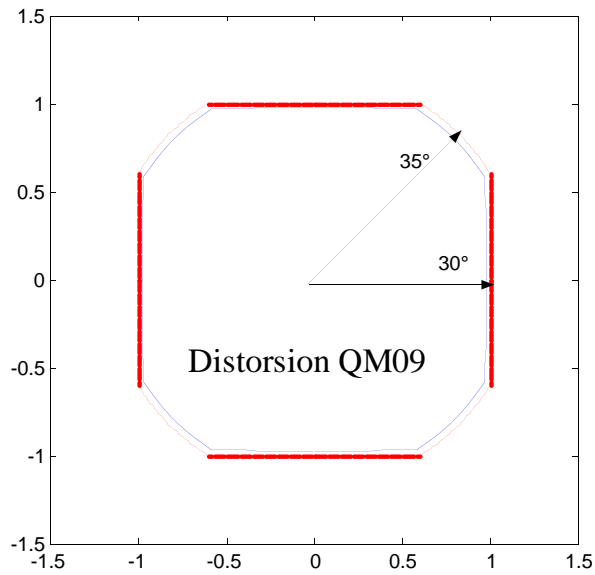
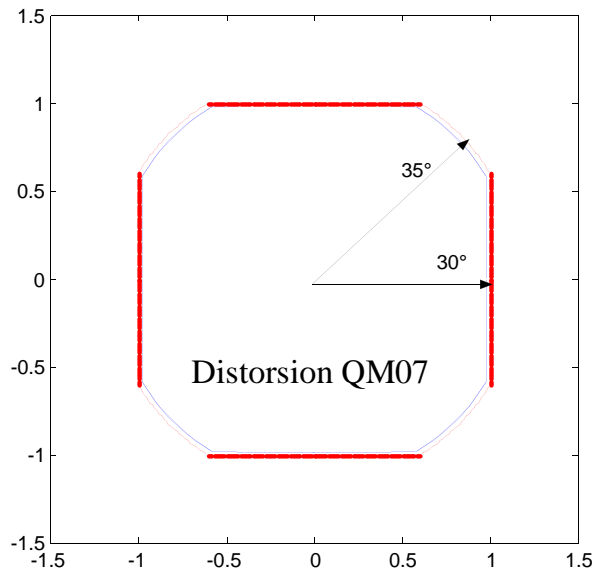
Authors : N. Charvin

Edition : 0

Revision : 0

Date Rev. : 12/12/2001

Page : 10/16





ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS

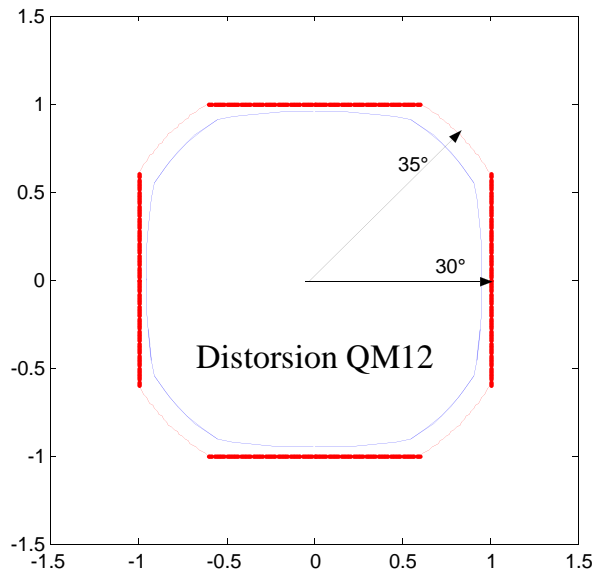
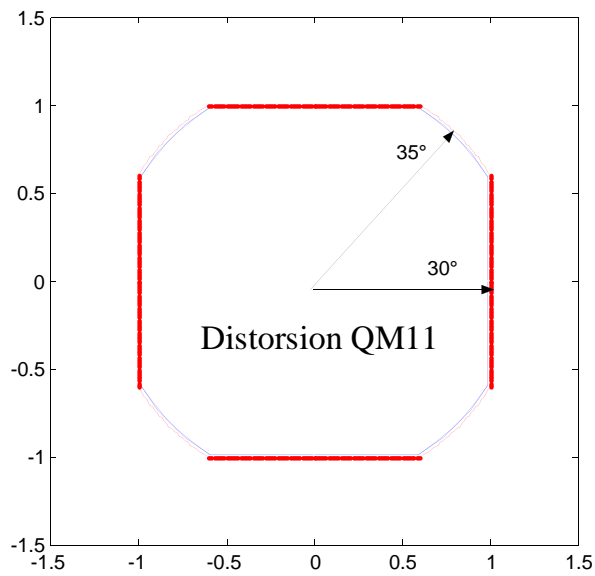
Authors : N. Charvin

Edition : 0

Revision : 0


Date Rev. : 12/12/2001

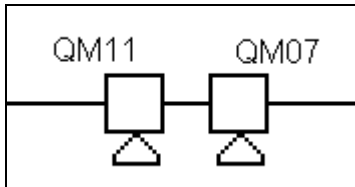
Page : 11/16



4 CALIBRATION STEREO

Les deux caméras QM07 et QM11 sont utilisées pour former la paire stéréo de CIVA-P.

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 12/16</p>
---	--	---



Intrinsic parameters of left camera:

```
Focal Length:      fc_left = [ 886.71445  886.47294 ] ± [ 1.59269  1.64454 ]
Principal point:   cc_left = [ 498.51480  512.14538 ] ± [ 1.41015  1.24271 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000 ]
=> angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:       kc_left = [ -0.01073  -0.00950  0.00007  -0.00104  0.00000 ]
                  ± [ 0.00531  0.01923  0.00050  0.00048  0.00000 ]
```

Intrinsic parameters of right camera:

```
Focal Length:      fc_right = [ 889.99649  889.62345 ] ± [ 1.61278  1.65212 ]
Principal point:   cc_right = [ 510.67236  523.72641 ] ± [ 1.41779  1.29511 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000 ]
=> angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:       kc_right = [ -0.01301  -0.00169  0.00048  0.00094  0.00000 ]
                  ± [ 0.00384  0.01007  0.00052  0.00045  0.00000 ]
```

Extrinsic parameters (position of right camera wrt left camera):

(les valeurs suivantes sont exprimées en millimètres)

```
Rotation vector:   om = [ 0.00109  -0.00305  0.00550 ] ± [ 0.00188  0.00176  0.00015 ]
Translation vector: T = [ 2.10740  -100.21617  -0.03749 ] ± [ 0.61401  0.62258  2.01152 ]
```

5 FICHIERS UTILISES

Les fichiers utilisés pour les mesures de distorsion géométrique portent les noms suivants:


QMxx_disto_nn.CIV

Avec xx: le numéro de la caméra, et nn: le numéro de l'image.

6 ANNEXE

6.1 Documentation de la Calibration Toolbox

On trouvera ci-dessus, un extrait de la documentation de la Calibration Toolbox, développée par Jean-Yves Bouguet, au Caltech. La documentation complète et originale se trouve à l'adresse suivante: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS</p> <p>Authors : N. Charvin</p> <p>Edition : 0</p> <p>Revision : 0</p> <p>Date Rev. : 12/12/2001</p> <p>Page : 13/16</p>
---	--	---

Description of the calibration parameters

After calibration, the list of parameters may be stored in the matlab file **Calib_Results** by clicking on **Save**. The list of variables may be separated into two categories: Intrinsic parameters and extrinsic parameters.

Intrinsic parameters (camera model):

The internal camera model is very similar to that used by Heikkilä and Silven at the University of Oulu in Finland. Visit their online calibration page, and their publication page. We specifically recommend their CVPR'97 paper: A Four-step Camera Calibration Procedure with Implicit Image Correction.

The list of internal parameters:

- **Focal length:** The focal length in pixels is stored in the 2x1 vector **fc**.
- **Principal point:** The principal point coordinates are stored in the 2x1 vector **cc**.
- **Skew coefficient:** The skew coefficient defining the angle between the x and y pixel axes is stored in the scalar **alpha_c**.
- **Distortions:** The image distortion coefficients (radial and tangential distortions) are stored in the 5x1 vector **kc**.

Definition of the intrinsic parameters:

Let **P** be a point in space of coordinate vector $\mathbf{XX}_c = [X_c; Y_c; Z_c]$ in the camera reference frame. Let us project now that point on the image plane according to the intrinsic parameters (**fc, cc, alpha_c, kc**). Let \mathbf{x}_n be the normalized (pinhole) image projection:

$$\mathbf{x}_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\text{Let } r^2 = x^2 + y^2.$$


After including lens distortion, the new normalized point coordinate \mathbf{x}_d is defined as follows:

$$\mathbf{x}_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \cdot (1 + \mathbf{kc}(1)*r^2 + \mathbf{kc}(2)*r^4 + \mathbf{kc}(5)*r^6) + \mathbf{dx}$$

where **dx** is the tangential distortion vector:

$$\mathbf{dx} = \begin{bmatrix} 2*\mathbf{kc}(3)*x*y + \mathbf{kc}(4)*(r^2 + 2*x^2) \\ \mathbf{kc}(3)*(r^2 + 2*y^2) + 2*\mathbf{kc}(4)*x*y \end{bmatrix}$$

Therefore, the 5-vector **kc** contains both radial and tangential distortion coefficients (observe that the coefficient of 6th order radial distortion term is the fifth entry of the vector **kc**). It is worth noticing that this distortion model was first introduced by Brown in 1966 and called "Plumb Bob" model (radial polynomial + "thin prism"). The tangential distortion is due to "decentering", or

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 14/16</p>
---	--	---

imperfect centering of the lens components and other manufacturing defects in a compound lens. For more details, refer to Brown's original publications listed in the reference.

Once distortion is applied, the final pixel coordinates $\mathbf{x}_{\text{pixel}} = [x_p; y_p]$ of the projection of \mathbf{P} on the image plane is:

$$\begin{aligned} x_p &= \mathbf{fc}(1) * (x_d(1) + \mathbf{alpha_c} * x_d(2)) + \mathbf{cc}(1) \\ y_p &= \mathbf{fc}(2) * x_d(2) + \mathbf{cc}(2) \end{aligned}$$

Therefore, the pixel coordinate vector x_{pixel} and the normalized (distorted) coordinate vector x_d are related to each other through the linear equation:


$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{KK} \cdot \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix}$$

where \mathbf{KK} is known as the camera matrix, and defined as follows:

$$\mathbf{KK} = \begin{bmatrix} \mathbf{fc}(1) & \mathbf{alpha_c} * \mathbf{fc}(1) & \mathbf{cc}(1) \\ 0 & \mathbf{fc}(2) & \mathbf{cc}(2) \\ 0 & 0 & 1 \end{bmatrix}$$

In matlab, this matrix is stored in the variable \mathbf{KK} after calibration. Observe that $\mathbf{fc}(1)$ and $\mathbf{fc}(2)$ are the focal distance (a unique value in mm) expressed in units of horizontal and vertical pixels. Both components of the vector \mathbf{fc} are usually very similar. The ratio $\mathbf{fc}(2)/\mathbf{fc}(1)$, often called "aspect ratio", is different from 1 if the pixel in the CCD array are not square. Therefore, the camera model naturally handles non-square pixels. In addition, the coefficient $\mathbf{alpha_c}$ encodes the angle between the x and y sensor axes. Consequently, pixels are even allowed to be non-rectangular. Some authors refer to that type of model as "affine distortion" model.

Currently manufactured cameras do not always justify this very general optical model. For example, it is now customary to assume rectangular pixels, and thus assume zero skew ($\mathbf{alpha_c}=0$). It is in fact a default setting of the toolbox (the skew coefficient not being estimated). Furthermore, the very generic (6th order radial + tangential) distortion model is often not considered completely. For standard field of views (non wide-angle cameras), it is often not necessary (and not recommended) to push the radial component of distortion model beyond the 4th order (i.e. keeping $\mathbf{kc}(5)=0$). This is also a default setting of the toolbox. In addition, the tangential component of distortion can often be discarded (justified by the fact that most lenses currently manufactured do not have imperfection in centering). The 4th order symmetric radial distortion with no tangential component (the last three component of \mathbf{kc} are set to zero) is actually the distortion model used by Zhang. Another very common distortion model for good optical systems or narrow field of view lenses is the second order symmetric radial distortion model. In that model, only the first component of the vector \mathbf{kc} is estimated, while the other four are set to zero. This model is also commonly used when a few images are used for calibration (too little data to estimate a more complex model). Aside from distortions and skew, other model reductions are possible. For example, when only a few images are used for calibration (e.g. one, two or three images) the principal point \mathbf{cc} is often very difficult to estimate reliably. It is known to be one of the most difficult part of the native perspective projection model to estimate (ignoring lens distortions). If this is the case, it is sometimes better (and recommended) to set the principal point at the center of the image ($\mathbf{cc} = [(\mathbf{nx}-1)/2; (\mathbf{ny}-1)/2]$)

	<p>ROSETTA LANDER</p> <p>ÇIVA EXPERIMENT</p>	<p>Reference : LCI-RP-052-3184-IAS Authors : N. Charvin Edition : 0 Revision : 0 Date Rev. : 12/12/2001 Page : 15/16</p>
---	--	---

and not estimate it further. Finally, in few rare instances, it may be necessary to reject the aspect ratio $fc(2)/fc(1)$ from the estimation. Although this final model reduction step is possible with the toolbox, it is generally not recommended as the aspect ratio is often 'easy' to estimate very reliably. For more information on how to perform model selection with the toolbox, visit the page describing the first calibration example.

Important Convention:

Pixel coordinates are defined such that $[0;0]$ is the center of the upper left pixel of the image. As a result, $[nx-1;0]$ is center of the upper right corner pixel, $[0;ny-1]$ is the center of the lower left corner pixel and $[nx-1;ny-1]$ is the center of the lower right corner pixel where nx and ny are the width and height of the image (for the images of the first example, $nx=640$ and $ny=480$). One matlab function provided in the toolbox computes that direct pixel projection map. This function is **project_points2.m**. This function takes in the 3D coordinates of a set of points in space (in world reference frame or camera reference frame) and the intrinsic camera parameters ($fc,cc,kc,alpha_c$), and returns the pixel projections of the points on the image plane. See the information given in the function.

The inverse mapping:

The inverse problem of computing the normalized image projection vector x_n from the pixel coordinate x_{pixel} is very useful in most machine vision applications. However, because of the high degree distortion model, there exists no general algebraic expression for this inverse map (also called normalization). In the toolbox however, a numerical implementation of inverse mapping is provided in the form of a function: **normalize.m**. Here is the way the function should be called: $x_n = \text{normalize}(x_{pixel},fc,cc,kc,alpha_c)$. In that syntax, x_{pixel} and x_n may consist of more than one point coordinates. For an example of call, see the matlab function **compute_extrinsic_init.m**.

[...]

Extrinsic parameters:

- **Rotations:** A set of n_{ima} 3x3 rotation matrices $Rc_1, Rc_2, \dots, Rc_{20}$ (assuming $n_{ima}=20$).
- **Translations:** A set of n_{ima} 3x1 vectors $Tc_1, Tc_2, \dots, Tc_{20}$ (assuming $n_{ima}=20$).

Definition of the extrinsic parameters:

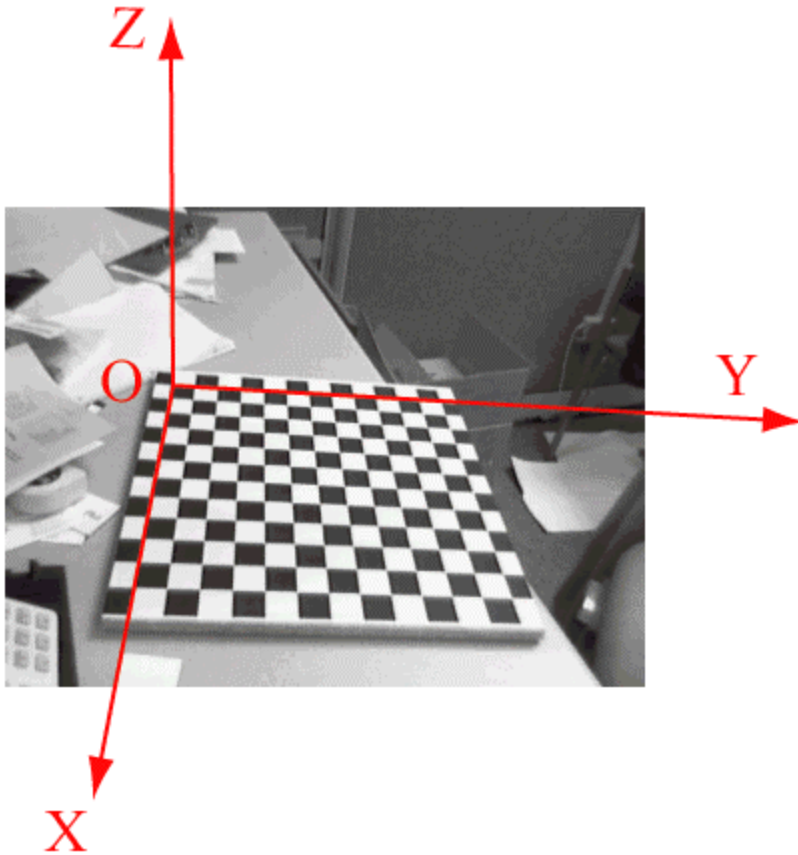
Consider the calibration grid # i (attached to the i th calibration image), and concentrate on the camera reference frame attached to that grid. Without loss of generality, take $i = 1$. The following figure shows the reference frame (O,X,Y,Z) attached to that calibration grid.



ROSETTA LANDER

ÇIVA EXPERIMENT

Reference : LCI-RP-052-3184-IAS
 Authors : N. Charvin
 Edition : 0
 Revision : 0
 Date Rev. : 12/12/2001
 Page : 16/16



Let \mathbf{P} be a point space of coordinate vector $\mathbf{XX} = [\mathbf{X};\mathbf{Y};\mathbf{Z}]$ in the grid reference frame (reference frame shown on the previous figure). Let $\mathbf{XX}_c = [\mathbf{X}_c;\mathbf{Y}_c;\mathbf{Z}_c]$ be the coordinate vector of \mathbf{P} in the camera reference frame. Then \mathbf{XX} and \mathbf{XX}_c are related to each other through the following rigid motion equation:

$$\mathbf{XX}_c = \mathbf{Rc}_1 * \mathbf{XX} + \mathbf{Tc}_1$$

In particular, the translation vector \mathbf{Tc}_1 is the coordinate vector of the origin of the grid pattern (\mathbf{O}) in the camera reference frame, and the third column of the matrix \mathbf{Rc}_1 is the surface normal vector of the plane containing the planar grid in the camera reference frame. The same relation holds for the remaining extrinsic parameters $(\mathbf{Rc}_2,\mathbf{Tc}_2)$, $(\mathbf{Rc}_3,\mathbf{Tc}_3)$, ... , $(\mathbf{Rc}_{20},\mathbf{Tc}_{20})$.

Once the coordinates of a point is expressed in the camera reference frame, it may be projected on the image plane using the intrinsic camera parameters.

The vectors $\mathbf{omc}_1, \mathbf{omc}_1, \dots, \mathbf{omc}_{20}$ are the rotation vectors associated to the rotation matrices $\mathbf{Rc}_1, \mathbf{Rc}_1, \dots, \mathbf{Rc}_{20}$. The two are related through the rodrigues formula. For example, $\mathbf{Rc}_1 = \mathbf{rodrigues}(\mathbf{omc}_1)$