Abstract

In this paper, we present a method to calibrate the omnidirectional sensor used in our laboratory and named SYCLOP (“Conic SYstem for LOcalization and Perception”). This system, which is able to get a panoramic image of a $2\pi$ radian field, is made of a CCD camera and a vertically oriented conic shaped reflector. In order to have a better precision than that obtained in classical applications using this kind of sensors, we bring to the fore the importance of calibration for the whole sensor. After having briefly recalled the theoretical framework used in hard calibration, we design the different transformations made between world object, cone reflector and pictures; as well as the different types of relationship between the world, the cone, the camera and the image coordinates. The last part of this paper presents results obtained with the SYCLOP simulator and an experimenting phase.

1 Introduction

Localization is the most important thing in autonomous mobile robotics. In fact, if we want to execute movements without mishap, it is imperative for us to know the trajectory of the RMA (at least the attitude and longitude for land robots). This information is obtained using different data sources proceeding from internal sensor types to localize it relatively to an initial position. In opposition, this information is obtained using external sensors to localize the RMA relatively to a known environment.

More and more robotic applications combine the use of different kinds of sensors with fusion data algorithms, which improve localization results to the detriment of computing time that is not compatible with real time.

To limit the number of sensors to be introduced in applications, a new type of sensor (“Omnidirectional sensor”) is in used in our laboratory. This kind of sensor can give a 360° sensorial model of the environment in one acquisition.

The using of omnidirectional sensors in robotic applications is an attractive idea. Cao in 1986 [1] was the first researcher who used this technique with fish-eye lenses. He obtained a distorted panoramic image. Ishiguron et al in 1993 [2] proposed another method using four cameras looking in four directions. Now, omnidirectional sensors are based on the conjoined use of a monocular camera and a conic reflector [3][4][5]. The advantage of Distortions that are generated lead users of these sensors to use particular features of the environment which are the vertical straight lines. These kind of sensor project them in radial lines into images [3][4][5][6]. In all these applications, without any exception, the calibration of the sensor had not been done, because the influence of a small non-alignment between the optical axis of the camera and cone axis on the one hand and the radial and tangential distortions on the other hand are minor on the detection and localization of the radial straight-lines in the image.

In this paper, we propose not to limit the detection to the radial straight lines, but to take into account all aligned sets of points which can be detected in the image. This can be expressed by landmark detection other than vertical straight lines. To improve the detection results, we propose to establish a model of the whole omnidirectional vision system relying on the principle used in classical camera calibration (or hard calibration). In the first part we remind readers of the basic mechanisms at work to calibrate monocular vision systems. We have integrated in the model the necessary and useful distortion coefficients to obtain a good image reconstruction. In the second part, we complete the previous model by integrating the conic reflector transformation based on the use of virtual point and the crossing matrix between the cone and the world system coordinate. The whole model resolution is done in two stages:

The first one allows to estimate intrinsic camera parameters and extrinsic parameters which expresses the rigid movement between the camera and the cone system coordinate.

The second one allows from the previous estimation to compute the set of parameters using the Levenberg & Marquardt algorithm.

To validate our method, we present in the simulation section the results we obtained with the SYCLOP simulator created from the TSAI simulator and fit to our sensor.

In the experimental section, the same configuration is employed but using real data with INRIA markup. We validate our work by computing, with whole sensor model, new image points not used during the system resolution.

2 Camera Model
The camera used in the experimentation is a pinhole model as shown in Figure 1.

We obtain the transformation matrix M:

\[
M = P_{\text{int}} \cdot P_{\text{ext}}
\]

This matrix consists of two kinds of parameters. The first set deals with the intrinsic parameters that provide the internal camera geometry and optical characteristics, the other one contains the extrinsic parameters giving the camera position and orientation.

\[
M = \begin{bmatrix}
\alpha_u & 0 & u_0 & 0 \\
0 & \alpha_v & v_0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \cdot \begin{bmatrix}
P_{Xw} \\
P_{Yw} \\
P_{Zw}
\end{bmatrix}
\]

(1)

In the intrinsic matrix, \(\alpha_u\) and \(\alpha_v\) are the scale factors (vertical, horizontal) multiplied by the focal length and \((u_0,v_0)\) are the coordinates of the optical center projection on the image plane; In the extrinsic matrix, the \(r_{ij}\) set represents an orthogonal rotation matrix \(R\) and \((t_x,t_y,t_z)^t\) is a translation vector \(T\).

We will note rotation angles \(\alpha, \beta, \gamma\) on the X, Y, Z axis.

With this camera model, the calibration problem is expressed in the following terms:

Given a sufficient number of visible points whose world coordinates \((P_{Xw_i}, P_{Yw_i}, P_{Zw_i})\) are known with a high precision, as well as their corresponding observed pixel positions \((u_i, v_i)\), we can estimate the value of the intrinsic and extrinsic parameters. Generally speaking, observed pixel locations are not equal to locations resulting from (1) because of acquisition, spatial digitization noises, point extraction and different kinds of deformations. As a result of several types of imperfections in the design and assembly of the lenses composing the camera optical system, geometrical distortion concerns the position of image points in the image plane. There are two kinds of distortions: radial and tangential [7]. For each kind of distortion, an infinite series is required, but as Tsai [8] and Beyer [9], we have noticed that only radial distortion needs to be considered, and only one term is needed [10].

In fact, we use the following model:

\[
\begin{align*}
\hat{u} &= \alpha_u - \frac{t_x}{f} + r_{11}P_{Xw} + r_{12}P_{Yw} + r_{13}P_{Zw} + t_x \\
\hat{v} &= \alpha_v - \frac{t_y}{f} + r_{21}P_{Xw} + r_{22}P_{Yw} + r_{23}P_{Zw} + t_y \\
\end{align*}
\]

(3)

with

\[
\hat{u} = (u-u_0)\quad \hat{v} = (v-v_0) \quad \text{and} \quad r = \sqrt{u^2 + v^2}.
\]

3 SYCLOP Model

3.1 Problem description

Our sensor, as the one used by Yagi in [4], is made of a conic mirror and a CCD camera (Figure 2).

Nowadays, this panoramic vision part only allows us to detect all the vertical elements on a \(2\pi\) radian domain because they generate a set of radial straight lines converging to the center of the cone through a 2D projection. To extend the detection to other lines, we have to calibrate this sensor. First, we have to determine a mathematical model of the transformation. An object of the world will be reflected on the conic mirror and projected on the image plane. Figure 3 shows the sensor geometry.

![Figure 3: SYCLOP geometry.](image-url)
The transformation contains a conic reflection that we compute with a virtual point notion.

As shown in Figure 4, a point P', projected of P on the image plane through the conic mirror, can be the result of the projection of any point of half line [IP], but it can be the projected of any point located on the half line [IP']. Thus, any point located on half line [IP'] can be interpreted as the projection of point V on the same plane. This last half line includes the set of the virtual points which are symmetrical of P according to the straight-line \( \Delta \). So, the projection of point P on the image plane can be computed with a virtual point notion.

3.1 Determination of the tangent of the cone

We know that \( \Delta \), the intersection between the cone and a particular plane \( \Pi \). This plane contains the points P, M, V, P' and the axis OZk. In fact, the plane \( \Pi \) can be expressed as:

\[
\alpha X_k + \beta Y_k = 0
\]

simplified by the following expression:

\[
X_k + \gamma Y_k = 0 \quad \text{with} \quad \gamma = \frac{\alpha}{\beta}
\]

(4)

Given \((P_{Xk}, P_{Yk}, P_{Zk})\), the coordinates of P in the cone frame, we know that P is located on plane \( \Pi \), so (4) becomes:

\[
X_k = \frac{P_{Xk}}{P_{Yk}} Y_k = 0
\]

(5)

We know that the tangent equation is:

\[
\begin{align*}
X_k &= aZ_k \\
Y_k &= bZ_k
\end{align*}
\]

(6)

As \( \Delta \) belongs to plane \( \Pi \), from (4) and (5) we can deduce:

\[
aZ_k - \frac{P_{Xk}}{P_{Yk}} bZ_k = 0
\]

With \( Z_k \) in factor, we obtain:

\[
a = \frac{P_{Xk}}{P_{Yk}} b
\]

(7)

The cone equation is:

\[
(hX_k)^2 + (hY_k)^2 - (rZ_k)^2 = 0
\]

(8)

\( \Delta \), also belongs to the cone, so (6) and (8) result in:

\[
(haZ_k)^2 + (hbZ_k)^2 - (rZ_k)^2 = 0
\]

By developing and with \( Z_k^2 \) in factor, we obtain:

\[
(a^2 + b^2) = \frac{r^2}{h^2}
\]

(9)

With equations (7) and (9), we can write:

\[
\frac{P_{Xk}}{P_{Yk}} b^2 + b^2 = \frac{r^2}{h^2} \quad \Rightarrow \quad b^2 = \frac{r^2}{h^2} \times \frac{P_{Yk}}{P_{Xk}^2 + P_{Yk}^2}
\]

So,

\[
b = \frac{r}{h} \times \frac{P_{Yk}}{\sqrt{P_{Xk}^2 + P_{Yk}^2}}
\]

(10)

With (7) and (10), we have:

\[
a = \frac{r}{h} \times \frac{P_{Xk}}{\sqrt{P_{Xk}^2 + P_{Yk}^2}}
\]

(11)

According to (6), (10) and (11), the equation of \( \Delta \) is:

\[
\begin{align*}
X_k &= \frac{r}{h} \times \frac{P_{Xk}}{\sqrt{P_{Xk}^2 + P_{Yk}^2}} \times Z_k \\
Y_k &= \frac{r}{h} \times \frac{P_{Yk}}{\sqrt{P_{Xk}^2 + P_{Yk}^2}} \times Z_k
\end{align*}
\]

(12)

3.2 The virtual point reckoning

Virtual point V is computed from real point P by performing an axial symmetry according to the straight line \( \Delta \). The intersection between the perpendicular straight line to the tangent of the cone containing point P and \( \Delta \) is point M. Finding point M is equivalent to finding the minimum distance between real point P and tangent \( \Delta \). Given \((M_{Xk}, M_{Yk}, M_{Zk})\), the coordinates of point M in the cone frame, this distance is:

\[
d(P, \Delta) = \sqrt{(M_{Xk} - P_{Xk})^2 + (M_{Yk} - P_{Yk})^2 + (M_{Zk} - P_{Zk})^2}
\]

(13)

Finding the minimum of \( d(P, \Delta) \) is equivalent to finding the minimum of \( d^2(P, \Delta) \). We obtain:

\[
d^2(P, \Delta) = M_{Xk}^2 - 2M_{Xk}P_{Xk} + P_{Xk}^2 + M_{Yk}^2 - 2M_{Yk}P_{Yk} + P_{Yk}^2 + M_{Zk}^2 - 2M_{Zk}P_{Zk} + P_{Zk}^2
\]

(14)

As M belongs to \( \Delta \), we can write:

\[
\begin{align*}
M_{Xk} &= aM_{Zk} \\
M_{Yk} &= bM_{Zk}
\end{align*}
\]

(15)
By writing \( d^2(P, \Delta P) = F(M_{Zk}) \), with (14) and (15):

\[
F(M_{Zk}) = a^2 M_{Zk} - 2a M_{Zk} P_{xk} + P_{xk}^2
+ b^2 M_{Zk} - 2b M_{Zk} P_{yk} + P_{yk}^2
+ M_{Zk}^2 - 2M_{Zk} P_{zk} + P_{zk}^2
\]

But finding the minimum of \( F(M_{Zk}) \) is equivalent to resolving \( F'(M_{Zk}) = 0 \) as:

\[
F'(M_{Zk}) = 2a M_{Zk} - 2a P_{xk} + 2b P_{yk} + 2M_{Zk} - 2P_{zk}
\]

so:

\[
M_{Zk} = \frac{(a P_{xk} + b P_{yk} + P_{zk})}{(a^2 + b^2 + 1)}
\]  

With (15) we find:

\[
M_{xk} = a \left( a^2 + b^2 + 1 \right)
\]

\[
M_{yk} = b \left( a^2 + b^2 + 1 \right)
\]

Given \((V_{xk}, V_{yk}, V_{zk})\), the coordinates of virtual point \( V \) in the cone frame, as \( M \) is located in the middle of segment \([PV]\), we can deduce the coordinates of point \( V \):

\[
V_{xk} = 2M_{xk} - P_{xk}
\]

\[
V_{yk} = 2M_{yk} - P_{yk}
\]

\[
V_{zk} = 2M_{zk} - P_{zk}
\]

By replacing the variables and simplifying the equations, we obtain:

\[
V_{xk} = P_{xk} \left( r^2 - h^2 \right) \left( P_{xk}^2 + P_{zk}^2 \right) + 2rh P_{zk} \left( P_{xk}^2 + P_{zk}^2 \right)
\]

\[
V_{yk} = P_{yk} \left( r^2 - h^2 \right) \left( P_{yk}^2 + P_{zk}^2 \right) + 2rh P_{zk} \left( P_{yk}^2 + P_{zk}^2 \right)
\]

\[
V_{zk} = \frac{2rh P_{zk} \left( P_{xk}^2 + P_{yk}^2 + P_{zk} \left( h^2 - r^2 \right) \right)}{r^2 + h^2}
\]  

Now, we are able to compute the set of virtual points corresponding with the set of real points.

### 3.3 The complete model

The transformation from a real Point \( P \) to its projected point \( P' \) consists in:

- a change from the world coordinate system to the cone coordinate system,
- the conic reflection,
- a change from the cone coordinate system to the camera coordinate system,
- the perspective projection.

So, we obtain the following model:

\[
u = a \alpha_1 V_{xw} + a \alpha_2 V_{yw} + a \alpha_3 V_{zw} + t_x + u_0 + k_u r^2
\]

\[
v = a \alpha_1 V_{xw} + a \alpha_2 V_{yw} + a \alpha_3 V_{zw} + t_y + u_0 + k_v r^2
\]

\[
v = a \alpha_1 V_{xw} + a \alpha_2 V_{yw} + a \alpha_3 V_{zw} + t_y + u_0 + k_v r^2
\]

With \((15)\) we find:

\[
\begin{align*}
\alpha_1 & = 2 \left( \frac{P_{xw}^2 + P_{yw}^2}{P_{xw}^2 + P_{yw}^2} \right) \\
\alpha_2 & = 2 \left( \frac{P_{xw}^2 + P_{zw}^2}{P_{xw}^2 + P_{zw}^2} \right) \\
\alpha_3 & = 2 \left( \frac{P_{yw}^2 + P_{zw}^2}{P_{yw}^2 + P_{zw}^2} \right) \\
\end{align*}
\]

\[
\begin{align*}
u & \approx \alpha_1 V_{xw} + \alpha_2 V_{yw} + \alpha_3 V_{zw} + t_x + u_0 + k_u r^2 \\
v & \approx \alpha_1 V_{xw} + \alpha_2 V_{yw} + \alpha_3 V_{zw} + t_y + u_0 + k_v r^2 \\
v & \approx \alpha_1 V_{xw} + \alpha_2 V_{yw} + \alpha_3 V_{zw} + t_z + u_0 + k_v r^2
\end{align*}
\]

### 4 Simulation

#### 4.1 Validation

In order to validate our model, we have used a simulator based on the Tsai method that had been modified to calculate the set of virtual points. According to the large number of parameters, we have decided to calibrate our sensor in two stages. Firstly, we place a two-plane calibration pattern on the top of the cone to estimate the internal parameters of the camera and the change from the cone frame to the camera coordinate system (Figure 5). These values are used as initial values in the second stage in which we place a two perpendicular plane calibration pattern to estimate all parameters (Figure 6).

Each stage has been simulated and we have generated two sets of synthetic data without adding noises, except radial distortion.

The results of this simulation are summarized in Table 1. The first line represents the values used to generate the synthetic data.

<table>
<thead>
<tr>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
<th>( \gamma_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>950.02</td>
<td>963.86</td>
<td>384.00</td>
<td>288.00</td>
<td>2.13e-7</td>
<td>87.50</td>
<td>93.83</td>
</tr>
<tr>
<td>949.91</td>
<td>963.98</td>
<td>376.96</td>
<td>296.02</td>
<td>2.13e-7</td>
<td>88.20</td>
<td>94.58</td>
</tr>
<tr>
<td>0.10</td>
<td>0.20</td>
<td>1.00</td>
<td>-0.17</td>
<td>-0.02</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>0.55</td>
<td>0.26</td>
<td>209.98</td>
<td>0.50</td>
<td>1.00</td>
<td>-89.96</td>
<td>599.70</td>
</tr>
<tr>
<td>0.10</td>
<td>0.20</td>
<td>1.00</td>
<td>-0.17</td>
<td>-0.02</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>0.55</td>
<td>0.26</td>
<td>209.98</td>
<td>0.50</td>
<td>1.00</td>
<td>-89.96</td>
<td>599.70</td>
</tr>
<tr>
<td>0.10</td>
<td>0.20</td>
<td>1.00</td>
<td>-0.17</td>
<td>-0.02</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1**: Estimation of model parameters for synthetic data having a radial distortion.

We have used the estimated values (in bold in Table 1) to simulate a new image. In this way, we can calculate the error on each calibration point. The mean error is less than \( 2 \times 10^{-4} \).

#### 4.2 Noises robustness

In order to determine the noises robustness, we have added different noises in accordance with Mohr [11]. We have noted that the main source of errors is the noise due to the CCD matrix. Uncertainties on the world point locations are much attenuated by the conic reflection. Also, we have seen that over a standard deviation of 0.3 pixels for the noise, due to the CCD matrix, parameter estimations were not accurate (and did not permit to rebuild points which had not been used in the calibration).
5 Experimentation

5.1 First Stage

First, as shown in Figure 5, we have acquired a multitude of images in order to calculate the mean of each parameter set in accordance with Puget and Skordas in [12]. In each image, we apply a Sobel Edge Detector in U and V directions to determine the intersection of the straight lines to obtain sub-pixel accuracy. Results obtained in this part are summarized in Table 2.

![Figure 5: The first stage of the calibration.](image)

<table>
<thead>
<tr>
<th>(\alpha_u)</th>
<th>(\alpha_v)</th>
<th>(\alpha_0)</th>
<th>(v_0)</th>
<th>(r)</th>
<th>(h)</th>
<th>(\alpha_1)</th>
<th>(\beta_1)</th>
<th>(\gamma_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>998.97</td>
<td>997.36</td>
<td>411.71</td>
<td>297.34</td>
<td>-2.3e-7</td>
<td>87.5</td>
<td>93.83</td>
<td>0.52</td>
<td>0.17</td>
</tr>
<tr>
<td>1133.6</td>
<td>1053.7</td>
<td>568.63</td>
<td>281.03</td>
<td>-2.3e-7</td>
<td>87.5</td>
<td>93.83</td>
<td>4.89</td>
<td>-13.35</td>
</tr>
<tr>
<td>0.52</td>
<td>0.17</td>
<td>0.27</td>
<td>-2.26</td>
<td>-0.46</td>
<td>206.07</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Means results of the first stage.

5.2 Second Stage

As shown in Figure 6, we have placed the pattern calibration to obtain several images (Figure 7). In each image, we have applied a Sobel Edges Detector in horizontal and vertical directions. In this way, we can estimate the straight line and calculate the intersection to obtain a calibration point set with a sub-pixel accuracy (Figure 8).

![Figure 6: The second stage of the calibration.](image)

![Figure 7: Some pictures of the pattern calibration through the conic sensor.](image)

![Figure 8: The points calibration set of an image.](image)

The first image (top-left on Figure 7) gives results summarized in Table 3. The first line represents initial values of the first stage and values measured during the experimentation.

<table>
<thead>
<tr>
<th>(\alpha_u)</th>
<th>(\alpha_v)</th>
<th>(u_0)</th>
<th>(v_0)</th>
<th>(k_1)</th>
<th>(r)</th>
<th>(h)</th>
<th>(\alpha_1)</th>
<th>(\beta_1)</th>
<th>(\gamma_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>998.97</td>
<td>997.36</td>
<td>411.71</td>
<td>297.34</td>
<td>-2.3e-7</td>
<td>87.5</td>
<td>93.83</td>
<td>0.52</td>
<td>0.17</td>
<td>0.27</td>
</tr>
<tr>
<td>1133.6</td>
<td>1053.7</td>
<td>568.63</td>
<td>281.03</td>
<td>-2.3e-7</td>
<td>87.5</td>
<td>93.83</td>
<td>4.89</td>
<td>-13.35</td>
<td>4.60</td>
</tr>
<tr>
<td>(t_{x_1})</td>
<td>(t_{y_1})</td>
<td>(t_{z_1})</td>
<td>(\alpha_2)</td>
<td>(\beta_2)</td>
<td>(\gamma_2)</td>
<td>(t_{x_2})</td>
<td>(t_{y_2})</td>
<td>(t_{z_2})</td>
<td></td>
</tr>
<tr>
<td>-2.26</td>
<td>-0.49</td>
<td>200.7</td>
<td>0</td>
<td>0</td>
<td>-13.5</td>
<td>584</td>
<td>0</td>
<td>-15.5</td>
<td></td>
</tr>
<tr>
<td>-34.43</td>
<td>0.34</td>
<td>200.99</td>
<td>-0.35</td>
<td>-1.10</td>
<td>-131.01</td>
<td>574.83</td>
<td>39.14</td>
<td>-12.54</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Parameter estimation on the first image.

In order to estimate the quality of the parameters estimation, we calculate the projection of the middle of the segment of the pattern calibration. On Figure 9, we can see that points are well projected in the image. From the results shown in Table 3, we can notice that internal parameters of the camera are really poorly estimated. This is due to an existing correlation between different parameters. Florou and Mohr have shown the existence of a correlation between \(u_0\) and \(t_x\) in [11]. Moreover, Brassart in [13] has shown that the estimation...
of the image center localization \((u_0, v_0)\) is dependent on the set of calibration points. On the first image, this set is located on the right. So, we can deduce that the estimation of \(u_0\) will be greater than the true value. Furthermore, \(u_0\) and \(t_{X_1}\) are correlated, so \(t_{X_1}\) will be badly estimated too. In this way, considering all images, we have placed the pattern calibration to have a homogeneous calibration point set. So, we estimate all parameters with each image, then we calculate the mean of each set to obtain the model parameters summarized in Table 4.

![Figure 9: The projection of the middle of segments of the pattern calibration in an image.](image)

<table>
<thead>
<tr>
<th>(\sigma_u)</th>
<th>(\sigma_v)</th>
<th>(\bar{u}_0)</th>
<th>(\bar{v}_0)</th>
<th>(\bar{k}_1)</th>
<th>(\bar{\rho})</th>
<th>(\bar{h})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1097.4</td>
<td>1090.2</td>
<td>379.58</td>
<td>289.86</td>
<td>-2.33 e-7</td>
<td>87.5</td>
<td>93.83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\bar{\alpha}_1)</th>
<th>(\bar{\beta}_1)</th>
<th>(\bar{\gamma}_1)</th>
<th>(\bar{\xi}_X)</th>
<th>(\bar{\xi}_Y)</th>
<th>(\bar{\xi}_Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.58</td>
<td>0.21</td>
<td>-5.21</td>
<td>2.24</td>
<td>-7.78</td>
<td>224.67</td>
</tr>
</tbody>
</table>

Table 4: SYCLOP parameters.

6 Conclusion

In this study we have described the process of SYCLOP calibration to establish the relationship between 2D points in the image coordinate system and 3D points in the world coordinate system. In the first stage, we have based our work on the utilization of the hard camera calibration to estimate the initial parameters. Before that, we had done a study to determine the significance distortion coefficients to introduce them in our model. According to [11], only the first coefficient had been hold regard on physical camera used in our application. In the second stage, we have calibrated the whole sensor.

To validate our model, we have proceeded in two parts: a simulation part in which we show, with the SYCLOP simulator, the influence of noise on detection points in world and image coordinate systems. The added noise in the 3D points and on their corresponding 2D points in the image permits us to assess the influence on parameter determination. A standard deviation superior to 0.3 pixel in the CCD matrix does not permit to compute with a great accuracy the parameters and to rebuilt 2D points from 3D points. These same standard deviations applied on 3D points have no influence on the parameter computation.

In the second part we have experimented with real data proceeding of SYCLOP sensor. The obtained results permit us to rebuild 2D points from 3D points not used in the parameters calculation. The error of projection is less than 0.3 pixels.

The future work is to integrate this calibration in our localization system for mobile robots in a verification stage.

7 References