Table of Contents

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

CaliPSO: Omnidirectional camera calibration based on Particle Swarm Optimization

Daniele Marzorati, Domenico G. Sorrenti, and Leonardo Vanneschi

Universitá di Milano - Bicocca, Italy

Abstract. In this paper we present a method for calibrating a catadioptric omnidirectional camera that is based on Particle Swarm Optimization (PSO). Like for other known approaches, only a few images, of a known calibration pattern, are required. First the projection model is shortly reviewed, and then it is shown that our proposed method can cope with the realistic noise acting during calibration. The experiments are presented in simulation, so that a comparison with ground truth is available, to characterize the performance of the method.

1 A short introduction

Omnidirectional vision systems provide a 360° panoramic view about the optical axis, like traditional perspective ones, but with a much larger vertical field of view. A common approach is to obtain such result by means of the combination of a catoptric (mirror) and a dioptric (lenses) part. Such devices, called catadioptric omnidirectional cameras, play an important role in robotics, and can be divided in two categories, as defined in [2]. On one hand, we have the devices where the Single View Point (SVP) property holds, e.g., hyperbolic mirror under perspective projection. The SVP property, provided the projection parameters are known, allow the un-warping of the panoramic image to perspective one. On the other, we have non-SVP devices, where the designer degrees of freedom are much more, and more complex application requirements might be satisfied, at the price of dealing with a non perspective image.

The calibration of the projection parameters of SVP catadioptric omnidirectional cameras have been the topic of relevant literature contributions, like [18], [15], [11], [16], [9], and [5]. Most notable, in our view, are [18] and [15] as the accompanying software toolboxes are available to the general public for use.

In general the number of parameters and the choice of the initialization for the optimization required in the calibration of such cameras make the task difficult, citing from [15] "too many parameters make the equations difficult to minimize because of the numerous local minima, the need for a lot of data and the numerical instability introduced into the Jacobian". The authors of this work approaches the problem by reducing the parameters, in particular leveraging on the assumption that the errors due to the assembly of the system are small, which we believe not being appropriate for most real systems. For instance, catadioptric lenses produced by VStone, a well established company in the field, are not

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

rigid with respect to the solicitations onboard a mobile robot, also when moving indoor; even worse is the fact that such systems tend to show some plasticity, under mechanical solicitations, so that, for high accuracy, continuous calibration could be a definitive solution. We are not tackling the continuous calibration problem in this work, as we are indeed dealing with the problem of robustness to local minima, noise, under a complete or even redundant, parametrization.

In [18] a toolbox is presented that allows to estimate the intrinsic parameters of the camera, the pose of the mirror and also the mirror shape (out of a family of polynomials). One can observe that the procedure is quite complex, with many steps independently performed, while the problem would be naturally formulated as a single optimization problem. Moreover, here and there in the paper, one can find dependencies on initial guesses, which are required to be reasonably good. PSO, on the other hand, allows to tackle the problem really as a single optimization and is quite robust to large errors in the initialization.

In [15] a method, and source code, is presented that allow to compute the parameters of the projection model. The model is simpler than appropriate, in our view, as it disregards misalignments between the optical axis and the mirror axis, taken as in the design, by fully trusting the assembly process. This method also requires accurate initializations, in order for the estimates to converge to the appropriate values. They rely on the mirror border to determine the principal point, a limitation in building a catadioptric system because it requires to waste a part of the image (i.e., {full image area} - {image of the mirror area}).

Like in [18] we compute the unknowns using only the points the user selects, without the need to observe the circular outer boundary of the mirror. Differently from others, beside basing on PSO, we sum distances between calibration points and interpretation lines of the calibration points. Anyway, the main contribution of this paper is to prove that a simple definition of the problem, i.e., a criterium for the evaluation of the solutions, and the PSO approach can solve the problem of estimating the parameters of catadioptric camera calibration, and allow an easy and accurate implementation of the procedure. On the contrary, other approaches tend to rely on fragmenting the problem and then tackling each sub-problem with a different method.

2 A short introduction to Particle Swarm Optimization

Many contributions have recently been dedicated to Particle Swarm Optimization (PSO) [12, 19, 4], since it is easily implementable, it does not rely on gradient information, and it is able to solve a wide array of optimization problems efficiently [8]. Furthermore, as reported in [1], PSO features reduced memory requirements and fast convergence with respect to other evolutionary algorithms (EAs).

PSO basic formulation works by establishing two attractors (normally the best local and global positions so far); besides that, the swarm behavior is influenced by parameters that control global exploration and local exploitation, and try to prevent the particles from prematurely converging to local minima [17].

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

Starting from the work presented in [21], several papers have been recently published which analyze and improve the performance of the PSO, trying to find the best possible values for these parameters. Several recent interesting researches in the literature describe techniques aimed at improving the performances of the PSO with different settings, which focus on the optimization of parameters such as the inertia weight, and the constriction and acceleration coefficients (see for instance [3, 1, 21, 22]).

Another interesting variant of PSO original formulation consists in establishing a given "structure" (or "topology") to the swarm. Among others, Kennedy and coworkers evaluate different kinds of topologies, finding that good performance is achieved using random and Von Neumann neighborhoods [13]. Nevertheless, the authors also indicate that, selecting the most efficient neighborhood structure is in general a problem-dependent task. In [6], Oltean and coworkers evolve the structure of an asynchronous version of the PSO algorithm. They use a hybrid technique that combines PSO with a genetic algorithm (GA), in which each GA chromosome is defined as an array which encodes an update strategy for the particles of the whole swarm. Such an approach works at macro and micro levels, that correspond, respectively, to the GA algorithm used for structure evolution, and to the PSO algorithm that assesses the quality of a GA chromosome at the macro level. The authors empirically show that the evolved PSO algorithm performs similarly and sometimes even better than standard approaches for several benchmark problems. They also indicate that, in structure evolution, several features, such as particle quality, update frequency, and swarm size influence the overall performance of PSO [7].

Many improvements based on the conjunction of EAs and PSO have been proposed, for example considering self-update mechanisms [20] or formation of 3D complex patterns in the swarm [14], to increase convergence speed and performance in the problems under consideration. Recently, a modified genetic PSO has been defined by Jian and colleagues [10], which takes advantage of the crossover and mutation operators, along with a differential evolution (DE) algorithm which enhances search performance, to solve constrained optimization problems.

Other work, aimed at solving global non-linear optimization problems is presented by Kou and colleagues in [23]. They have developed a constraint-handling method in which a double PSO is used, together with an induction-enhanced evolutionary strategy technique. Two populations preserve the particles of the feasible and infeasible regions, respectively. A simple diversity mechanism is added, allowing the particles with good properties in the infeasible region to be selected for the population that preserves the particles in the feasible region. The authors state that this technique could effectively improve the convergence speed with respect to plain PSO.

As reported PSO has had such a fast development and growing popularity in the last few years that it can now be considered a widely accepted method of optimization of continuous parameters. It thus comes natural to us to employ

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

PSO for the omnidirectional camera calibration, where the local minima and the noise contribute to make the optimization difficult.

3 Omnidirectional Camera Model

In this section, we present the geometric model adopted to describe catadioptric imaging systems. In Figure 1 we see a schematic drawing of a typical omndirectional camera system. The reference system in the middle, indicated with C, represents the camera (camera reference frame $[x_C, y_C, z_C]^T$). Note that this reference frame has a precise physical meaning, as its origin is the center of projection. It depends on the camera intrinsic parameters. In front of the camera there is an hyperbolic mirror. In the rest of the paper, this mirror will be represented by a quadric Q.

We can define a reference frame local to the mirror whose z-axis corresponds to the hyperbolic axis (mirror reference frame $[x_M, y_M, z_M]^T$). The mirror is in a generic position w.r.t. the camera: its axis is not parallel to the optical axis and it does not pass through the pinhole. The roto-translation between the camera reference frame and the mirror will be represented with $\mathbf{R}_C^M$ (position of the camera w.r.t. the mirror).

Moreover, we introduce a further reference frame $[x_W, y_W, z_W]^T$, called world reference frame. All calibration data (points on the calibration pattern) will be referred w.r.t. this reference system. The roto-translation from the mirror to the world reference frame is $\mathbf{R}_M^W$ (position of the mirror w.r.t. the world). This transformation will be the extrinsic parameters of the omnidirectional camera system.

4 PSO Calibration of a catadioptric omnidirectional camera

The purpose of the calibration procedure is to estimate the values of the intrinsic parameters of the camera M and the coefficients of the roto-translations $\mathbf{R}_C^M$ and $\mathbf{R}_M^W$. To compute these parameters we use a set of calibration points and the corresponding image points whose coordinates are known. In particular, the calibration points are on the calibration pattern in the world reference frame while the corresponding image points are referred w.r.t. the camera reference frame.

Using the intrinsic parameters M (unknown) and the coordinates of the corresponding image points we can compute the interpretations lines w.r.t the camera system. The roto-translation between camera and mirror $\mathbf{R}_C^M$ (unknown) allows to compute the reflected interpretation lines w.r.t the mirror reference system. Finally, if we know the transformation between mirror and world reference system $\mathbf{R}_M^W$ (unknown) we can compute the intersections between the interpretation lines and the calibration points.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

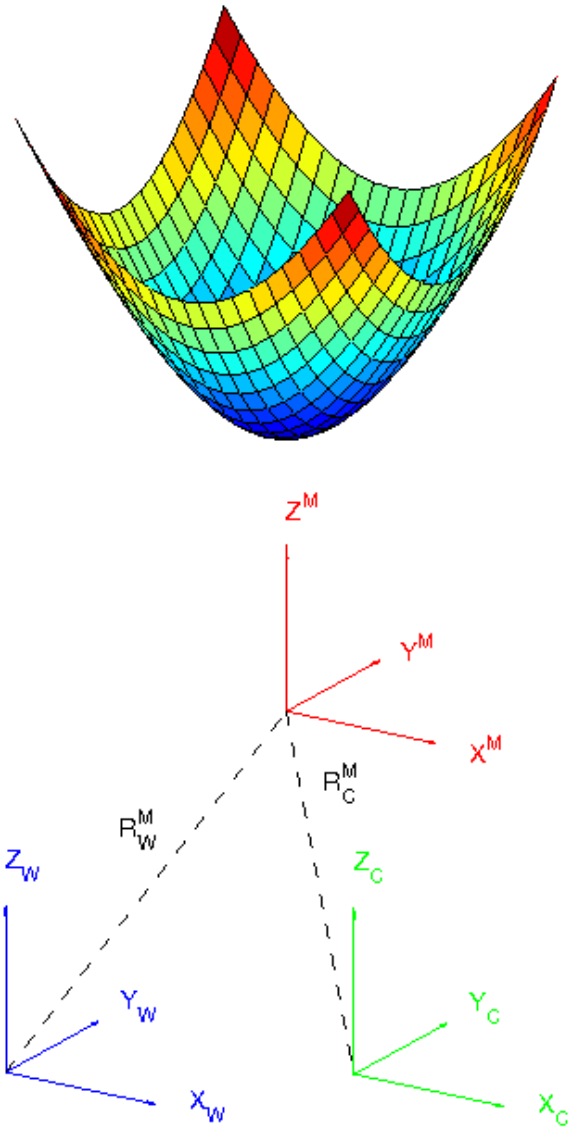

Fig. 1. Reference frames: in red the camera reference frame, in green the mirror reference frame and in blue the world reference frame.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

Therefore, the estimation of the parameters of the omnidirectional camera model can be treated as a optimization problem, in which the aim is to minimize the following point to line distance:

$$D = \sum_{i \leq m*n}^{i} \|p_i - k_i - v_i \hat{t}_i\| \tag{1}$$

where p_i is the i-th 3D point, $q_i = k_i - v_i t_i$ is the parametric form of the line and $\hat{t}_i = v_i^T (p_i - k_i)$.

This equation describe the distance between a calibration 3D point p_i in the mirror reference frame and the interpretation line q_i. This interpretation line is computed in different steps.

- roto-translation of the interpretation line w.r.t the mirror reference frame
- computation of the intersection between this interpretation line and the mirror
- computation of the normal plane to the surface w.r.t. intersection point
- computation of the reflected interpretation line

Thus, let Q be a symmetric, non-zero matrix representing the quadric (mirror). The locus of points belonging to the quadric surface is given by $F = x^T Q x = 0$ where $\mathbf{x} = [x, y, z, 1]^T$.

Given a point $\mathbf{k} = [k_x, k_y, k_z]^T$ and a direction $\mathbf{v} = [v_z, v_y, v_z]^T$, let $\mathbf{q}(t) = k + vt$ be the parametric equation of the line passing through $\mathbf{k}$ with direction $\mathbf{v}$, whose intersection with the quadric we want to compute.

This line is computed using the following relation:

$$\mathbf{p}_{2D}^C = \mathbf{M}\mathbf{R}_M^C \mathbf{p}_{3D}^M \tag{2}$$

By substituting the line into the quadric, grouping and rewriting as a function of t we obtain $At^2 + Bt + C = 0$ which is a second degree polynomial in t.

We solve it to find t_1 , t_2 and substitute them into the line to find the intersection points $\mathbf{P}_1$, $\mathbf{P}_2$ of the line with the quadric. Only one of the two intersections has a physical meaning, while the other is a point either behind the camera or outside the mirror or also occluded by the mirror itself.

Once the intersection point $\mathbf{P}_i$ is known, we compute the normal to the quadric in $\mathbf{P}_i$ as:

$$\hat{n}_i = \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial x_i} \\ \frac{\partial \mathbf{F}}{\partial y_i} \\ \frac{\partial \mathbf{F}}{\partial z_i} \end{bmatrix} \tag{3}$$

computed in $\mathbf{P}_i$.

Finally, we apply the law of reflection to the line q obtaining the direction vector r as:

$$\mathbf{r} = 2\left(\hat{n}^T v\right)^T \hat{n} - \mathbf{v} \tag{4}$$

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

These equations are obtained using the projected image points $\mathbf{p}_{2D}$, the intrinsic camera parameters $\mathbf{M}$ and the roto-translation between the camera and the mirror $\mathbf{R}^{\mathbf{C}}_{\mathbf{M}}$.

For each image point we have an interpretation line. All the reflected interpretation lines are so defined:

$$\hat{q}_i = \mathbf{r}_i t + \mathbf{P}_i \tag{5}$$

They pass through their corresponding 3D scene points, i.e., the distance between a line and its point is zero. These points needed to be represented w.r.t. the mirror reference frame by the roto-translation $\mathbf{R}^M_W$.

Our aim is to find the following quantities:

- roto-translation between the camera and the mirror reference frame $\mathbf{R}^C_M$
- roto-translation between the mirror reference frame and the world reference frame $\mathbf{R}^M_W$
- the camera intrinsic parameters

that allows the zeroing of the distances between the calibration points and their reflected interpretation lines.

Actually, because of noise, the lines will only pass close to the scene points, i.e., we can only find quantities that minimizes the sum of the squares of the calibration points distances to their interpretation lines.

We choose to use a Particle Swarm Algorithm to solve this minimization problem. The algorithm starts with a set of random particles. The idea is to minimize the objective function $f(x)$, with $f \in \Re^n \rightarrow \Re$, using a swarm of dynamic particles which explore the parameter space $x \in \Omega \subset \Re^n$ searching for the minima.

For each step, each particle evaluates the function and keeps track of the best solution it has found so far. The current best solution among all the particles is also tracked using a global best parameter. In this way, during the evolution the particles are directed to the good solution while maintaining some randomness to explore the search space trying to find the best solution. The PSO algorithm:

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

Initialize x_i and v_i . The value of x_i is randomly drawn from an uniform distribution among $[x_{min}, x_{max}]$ and the value of v_i is randomly drawn from an uniform distribution among $[v_{min}, v_{max}]$. Evaluate $f(x_i)$ and set $g = argminf(x_i)$ and $\hat{x}_i$. **while** $g > threshold$ **do**
  foreach x_i **do**
    $v_i \leftarrow \omega v_i + c_1\dot{r}_1(\hat{x}_i x_i) + c_2\dot{r}_2(g x_i)$
    if $v_i > v_{max}$ **then**
      $v_i = v_{max}$
    else if $v_i < v_{min}$ **then**
      $v_i = v_{min}$
    $tmp \leftarrow x_i + v_i$
    if $tmp > x_{max}$ **then**
      $v_i \leftarrow -v_i$
      $x_i \leftarrow x_{max}$
    else if $tmp < x_{min}$ **then**
      $v_i \leftarrow -v_i$
      $x_i \leftarrow x_{min}$
    else
      $x_i \leftarrow tmp$
    if $f(x_i) < f(\hat{x}_i)$ **then**
      $\hat{x}_i = x_i$
    if $f(x_i) < f(g)$ **then**
      $g = x_i$
  end
end

Each particle x_i represents, in our case, a vector containing: the camera intrinsic parameters $\mathbf{M}$, the roto-translation $\mathbf{R}_M^C$ and the roto-translation $\mathbf{R}_W^M$. The best position up to now is $\hat{x}_i$ and the global best is g. At any given time, the velocity of particle i, v_i, is updated to point towards $\hat{x}_i$ and g, up to a random factor. The inertia coefficient is represented by ω, while c_1 and c_2 are the ... coefficients. r_1 and r_2 are random vectors uniformly distributed among $[0, 1]$.

5 Experimental Results

In this section we present the capabilities of our system using simulated data. The simulator allows to compute the projection of a 3D point in the camera image plane through the mirror. Given a point in the scene (from the calibration pattern), the position of the camera w.r.t. the mirror and the position of the reference system of the calibration pattern w.r.t. the mirror, we simulate the image formation on the device. This process includes the reflection of the interpretation line of this 3D point on the mirror and, subsequently, the projection of them on the camera image plane. The motivation for using a simulated environment to test the proposed model is to have access to the ground truth and therefore to compare different runs using the same data.

The simulated world is a set of planar calibration patterns. Each pattern is composed by n calibration points. Parameters used for the simulated camera

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

are: image resolution of 640x480 pixels, focal length of 650 pixels and projection center of 320 pixels.

The mirror is an hyperbolic surface with equation:

$$-\frac{x^2}{a^2} - \frac{y^2}{b^2}\frac{z^2}{c^2} = 1 \tag{6}$$

where $a = 50$, $b = 50$ and $c = 50$.

ϕ	γ	θ	x	y	z
10.000000	2.000000	4.000000	50.000000	20.000000	100.000000
7.000000	3.000000	6.000000	0.000000	40.000000	200.000000
4.000000	4.000000	8.000000	-50.000000	60.000000	300.000000

Table 1. GroundTruth of the calibration pattern position w.r.t. the mirror reference frame: angles terms are in degrees and the traslation terms are in cm

ϕ	γ	θ	x	y	z
-1.800000	0.000000	4.500000	0.700000	1.500000	30.300000

Table 2. GroundTruth of the rototraslation between the camera reference frame and the mirror reference frame: angles terms are in degrees and the traslation terms are in cm

f_x	f_y	c_x	c_y
650.000000	320.000000	650.000000	240.000000

Table 3. GroundTruth of the camera intrisic parameters

In the experiment we use $m = 3$ planes. The positions of the reference frame of these planes w.r.t. the mirror reference frame are shown in table 5. A Gaussian noise with 0 mean and $\sigma = 0.3pixels$ standard deviation is added to each projected image point.

The parameters of the PSO algorithm are: $w = 0.6$, $c1 = 1.19$ and $c2 = 1.19$. We use 100 particles and the cost function is described in section 4. The algorithm stops when the fitness falls above a termination threshold.

In table 5, the estimated camera parameters are compared with the ground truth : in particular we represent the relative error for the focal length f_x and f_y while absolute error for the image center c_x and c_y.

In table 5 and 5 we show the differences (as absolute error) between both the estimated position of the camera and the computed location of the patterns with the expected values.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

ϕ	γ	θ	x	y	z
10.013215	2.030693	4.059546	48.927297	20.036825	100.026407
6.993828	3.014812	6.063330	-1.830115	40.145591	200.236215
4.000911	4.012032	8.055714	-52.240358	59.977741	300.409171

Table 4. Estimated rototraslation between the calibration pattern position w.r.t. the mirror reference frame: angles terms are in degrees and the traslation terms are in cm

ϕ	γ	θ	x	y	z
-1.767626	0.053566	4.540408	0.554351	1.483190	30.328784

Table 5. Estimated rototraslation between the camera reference frame and the mirror reference frame: angles terms are in degrees and the traslation terms are in cm

f_x	f_y	c_x	c_y
650.128196	650.255554	319.341323	240.321775

Table 6. Estimated camera intrisic parameters

f_x rel.err.	f_y rel.err.	c_x abs.err.	c_y abs.err.
0.02%	0.04%	0.658677px	0.321775px

Table 7. Relative and absolute error w.r.t. the groundtruth in the estimated camera intrisic parameters

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.032374	0.053566	0.040408	0.145649	0.01681	0.028784

Table 8. Error in the estimated rototraslation between the camera reference frame and the mirror reference frame

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.013215	0.030693	0.059546	1.072703	0.036825	0.026407
0.006172	0.014812	0.063330	1.830115	0.145591	0.236215
0.000911	0.012032	0.055714	2.240358	0.022259	0.409171

Table 9. Error in the estimated rototraslation between the calibration pattern position w.r.t. the mirror reference frame

f_x rel.err.	f_y rel.err.	c_x abs.err.	c_y abs.err.
0.0496%	0.0746%	3.3477px	0.8653px

Table 10. Relative and absolute error w.r.t. the ground truth in the estimated camera intrinsic parameters with image error σ=1px

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.0018	0.0040	0.0006	0.0590	0.0398	0.1259

Table 11. Error in the estimated rototraslation between the camera reference frame and the mirror reference frame with image error σ=1px

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.0003	0.0001	0.0017	2.0834	0.1381	0.3521
0.0004	0.0007	0.0014	3.5876	0.3140	0.4953
0.0005	0.0005	0.0008	3.9270	0.2103	0.2663

Table 12. Error in the estimated rototraslation between the calibration pattern position w.r.t. the mirror reference frame with image error σ=1px

f_x rel.err.	f_y rel.err.	c_x abs.err.	c_y abs.err.
0.1318%	0.1166%	2.8844px	1.5356px

Table 13. Relative and absolute error w.r.t. the groundtruth in the estimated camera intrisic parameters with image error σ=1.5px

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.0014	-0.0044	0.0005	0.2614	0.1012	0.8455

Table 14. Error in the estimated rototraslation between the camera reference frame and the mirror reference frame with image error σ=1.5px

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
0.0012	0.0018	0.0019	2.8576	3.9626	2.5486
0.0012	0.0026	0.0026	5.1842	3.9093	1.8180
0.0005	0.0006	0.0000	4.0907	7.7427	2.5765

Table 15. Error in the estimated rototraslation between the calibration pattern position w.r.t. the mirror reference frame with image error σ=1.5px

ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.	f_x rel.err.	f_y rel.err.	c_x abs.err.	c_y abs.err.	run
0.000190	0.001040	0.000327	0.044153	0.036161	0.508815	0.046286	0.017978	0.729404	0.169850	1
0.009352	0.002027	0.000714	0.481637	0.308773	0.212802	0.145977	0.225087	2.961853	3.054735	2
0.004604	0.002318	0.001048	0.285156	0.112183	0.555148	0.009929	0.060829	1.589454	2.965404	3
0.000733	0.002632	0.000446	0.291984	0.123066	0.295119	0.043363	0.043161	2.327046	0.913163	4
0.003679	0.002344	0.000829	0.039905	0.090441	0.171289	0.079702	0.056704	2.610476	2.051325	5
0.006699	0.001757	0.000114	0.108475	0.123923	0.785393	0.173032	0.037745	1.147003	4.531109	6
0.003531	0.001731	0.000062	0.073676	0.059392	0.006993	0.023689	0.080116	1.238297	1.555388	7
0.007434	0.006689	0.000024	0.011204	0.055356	0.572381	0.153693	0.116142	4.914833	4.757583	8
0.001438	0.002958	0.000278	0.107102	0.022454	0.139206	0.026949	0.040314	2.055081	1.031710	9
0.010494	0.002108	0.000283	0.042087	0.127189	0.839445	0.145986	0.008466	1.712255	6.088573	10
0.001755	0.001126	0.000554	0.056696	0.024599	0.142849	0.083002	0.057012	1.082364	0.968800	11
0.004761	0.004047	0.000146	0.233204	0.067994	0.473782	0.069670	0.057046	2.726377	2.746185	12
0.007255	0.001481	0.004183	0.372138	0.092056	1.200014	0.455616	0.480298	1.899352	4.653922	13
0.010905	0.000232	0.000883	0.295697	0.128164	0.914409	0.003506	0.298597	1.030696	4.958919	14
0.002234	0.005052	0.000541	0.076094	0.062677	0.378180	0.024563	0.103262	3.683589	1.565450	15
0.013720	0.000501	0.000271	0.038796	0.358644	0.873706	0.136085	0.066848	0.068380	8.851165	16

Table 16. Error in the estimated rototraslation between the camera reference frame and the mirror reference frame and camera parameters using the camera-mirror positions reported in 5

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

run	ϕ abs.err.	γ abs.err.	θ abs.err.	x abs.err.	y abs.err.	z abs.err.
1	-0.104720	0	0	0	0	30
2	0.104720	0	0	0	0	30
3	0	-0.104720	0	0	0	30
4	0	0.104720	0	0	0	30
5	0	0	-0.104720	0	0	30
6	0	0	0.104720	0	0	30
7	0	0	0	-1	0	30
8	0	0	0	1	0	30
9	0	0	0	0	-1	30
10	0	0	0	0	1	30
11	0	0	0	0	0	27
12	0	0	0	0	0	33
13	-0.078540	0.078540	0	0	0	30
14	0.078540	0	0.078540	0	0	30
15	0	0	0	-1.500000	1.500000	30
16	0	0	0	-1.500000	0	31.800000

Table 17. Camera-mirror positions used in the experiment 5.

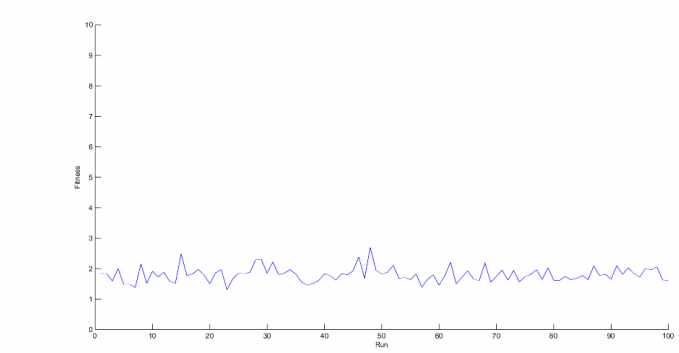

Fig. 2. Fitness values after 100 independent trials

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

6 CONCLUSIONS AND FUTURE WORKS

We presented a method, based on Particle Swarm Optimization, for the calibration of the parameters of a Single View Point catadioptric omnidirectional camera. The method is quite robust to noise local minima, and has been evaluated in simulation, to obtain a comparison with with the ground truth, and gave good results. The next steps for us is to compare the method with other approaches, both with real and simulated data. Then to insert it in a proper image interface, so to make the work a toolbox, and not a prototype.

7 ACKNOWLEDGMENTS

This work has been partially supported by the EEC Project RAWSEEDS, Contract Number FP6-045144.

References

1. M. Senthil Arumugam and M.V.C. Rao. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (rms) variants for computing optimal control of a class of hybrid systems. *Journal of Applied Soft Computing*, 8:324–336, 2008.
2. Simon Baker and Shree K. Nayar. A theory of catadioptric image formation. In *ICCV*, pages 35–42, 1998.
3. S. Cagnoni, L. Vanneschi, A. Azzini, and A.G.B. Tettamanzi. A critical assessment of some variants of particle swarm optimization. In *European Workshop on Bio-inspired algorithms for continuous parameter optimisation, EvoNUM'08*, pages 565–574. Springer Verlag, 2008.
4. M. Clerc, editor. *Particle Swarm Optimization*. ISTE, 2006.
5. Alberto Colombo, Matteo Matteucci, and Domenico G. Sorrenti. On the calibration of non single viewpoint catadioptric sensors. In *RoboCup*, volume 4434 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2006.
6. L. Diosan and M. Oltean. Evolving the structure of the particle swarm optimization algorithms. In *European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP'06*, pages 25–36. Springer Verlag, 2006.
7. L. Diosan and M. Oltean. What else is evolution of pso telling us? *Journal of Artificial Evolution and Applications*, 1(5):1–12, 2008.
8. A.A.A. Esmin, G. Lambert-Torres, and A.C. Zambroni de Souza. A hybrid particle swarm optimization applied to loss power minimization. *IEEE Transactions on Power Systems*, 20:859–866, 2005.
9. C. Greyer and K. Daniilidis. Paracatadioptric camera calibration. *IEEE Transaction on PAMI*, 24(5):687 – 695, May 2002.
10. L. Zhiming W. Cheng L. Jian. Solving contrained optimization via a modified genetic particle swarm optimization. In *Workshop on Knowledge Discovery and Data Mining, WKDD'08*, pages 217–220. IEEE Computer Society, 2008.
11. Sing Bing Kang. Catadioptric self-calibration. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2000.

12. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. conf. on Neural Networks*, volume 4, pages 1942–1948. IEEE Computer Society, 1995.
13. J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *IEEE Congress on Evolutionary Computation, CEC'02*, pages 1671–1676. IEEE Computer Society, 2002.
14. H. Kwong and C. Jacob. Evolutionary exploration of dynamic swarm behaviour. In *IEEE Congress on Evolutionary Computation, CEC'03*, page 367374. IEEE Press, 2003.
15. C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, 2007.
16. B. Micusik and T. Pajdla. Autocalibration & 3d reconstruction with non-central catadioptric cameras. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2004.
17. A. Ratnaweera, S.K. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, 2004.
18. Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *IROS*, 2006.
19. Y. H. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proc. IEEE Int. Conference on Evolutionary Computation*, pages 69–73. IEEE Computer Society, 1998.
20. D. Srinivasan and T. H. Seow. Particle swarm inspired evolutionary algorithm (ps-ea) for multi-objective optimization problem. In *IEEE Congress on Evolutionary Computation, CEC03*, page 22922297. IEEE Press, 2003.
21. Y. Del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, and R.G. Harley. Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation*, 12(2):171–195, 2008.
22. Z. Wu and J. Zhou. A self-adaptive particle swarm optimization algorithm with individual coefficients adjustment. In *Proc. IEEE International Conference on Computational Intelligence and Security, CIS'07*, pages 133–136. IEEE Computer Society, 2007.
23. X.L. You, S.Y. Liu, and W. Zheng. Double-particle swarm optimization with induction-enhanced evolutionary strategy to solve constrained optimization problems. In *IEEE International Conference on Natural Computing, ICNC'07*, pages 527–531. IEEE Computer Society, 2007.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272

Subject Index

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 256-272