Self-calibration from Planes Using Differential Evolution

Luis Gerardo de la Fraga

Cinvestav, Computer Science Department,
Av. Instituto Politécnico Nacional 2508, 07360 Mexico City, Mexico
`fraga@cs.cinvestav.mx`

Abstract. In this work the direct self-calibration of a camera from three views of a unknown planar structure is proposed. Three views of a plane are sufficient to determine the plane structure, the view's positions and orientations and the camera's focal length. This is a non-linear optimization problem that is solved using the heuristic Differential Evolution. Once an initial structure is obtained, the bundle adjustment can be used to incorporate more views and estimate other camera intrinsic parameters and possible lens distortion. This new self-calibration method is tested with real data.

Keywords: Computer Vision, camera self-calibration, self-calibration from planes, differential evolution.

1 Introduction

Self-calibration is defined in [1, Chap. 19] as “the computation of metric properties of the cameras and/or the scene from a set of uncalibrated images”. Camera (self-)calibration is one of the most important problems in computer vision. Its purpose is to obtain through a camera, an estimation of the parameters to transform a point in the real world to a point in an image. Self-calibration avoids the tedious process of calibrating cameras using special calibration objects. Using self-calibration, a camera can be calibrated on-line, i.e. every time a zoom is made. Therefore, a self-calibration technique must be used if camera's zoom is changed or auto-focus is active.

In the self-calibration process is assumed only image features correspondences to be known, and also it is possible to obtain a three-dimensional reconstruction up to an unknown similarity transformation (also called Euclidean reconstruction), but it is necessary to have some additional information about either the cameras' intrinsic parameters, the extrinsic parameters or the viewed object in order to obtain the desired Euclidean reconstruction [2,3].

Calibration using planes is a very flexible task, because it is very easy to produce high quality planar patterns (like a chessboard) easily made and printed in a laser printer. For this purpose, the Zhang [4] or Sturm [5] techniques can be used. For self-calibration with planes, no information about the planar pattern is needed, and a technique using metric rectification is available [6], but this

technique needs at least four images to work. Other technique, with a different approach but also solving a global optimization task, is presented in [7].

In this paper, a novel form to solve the plane based self-calibration problem, using the heuristic Differential Evolution (DE) [8] is proposed. This new method obtains an initial plane reconstructed from three different images of that plane. DE is used to estimate the focal length and the three orientation angles and the three camera positions of each view. This is a non-linear optimization problem that is solved with DE by minimizing the reprojection error. To add more images and to correct possible camera's lens distortion, the standard bundle adjustment can be used.

The paper is organized as follows: in Sec. 2 the problem of self-calibration from planes is established. In Sec. 3 DE is briefly described. In Sec. 4 the experiment and results are shown. A briefly discussion is in Sec. 5. And finally, conclusions of this work are drawn in Sec. 6.

2 Self-calibration from Three Planes

Camera model: A point over an image is represented by $\mathbf{p} = [u, v, 1]$, a three-dimensional point is represented by a vector $\mathbf{P} = [x, y, z, 1]$. The relation between the two points is the so called pinhole camera model which is:

$$\lambda \mathbf{p} = K[R|\mathbf{t}]\mathbf{P}, \quad (1)$$

where λ is a scale factor, K a 3×3 matrix of the camera's intrinsic parameters, R a 3×3 rotation matrix and $\mathbf{t}$ a vector $[t_1, t_2, t_3]^T$. A rotation matrix depends on only three parameters: three rotations around the main axes. The pinhole camera model in (1) represents the projection transformation of a 3D scene, by K , of a view obtained by rotating and translating the scene with respect to the world coordinates.

In the case of planes, a three-dimensional point can be also represented by the vector $\mathbf{P} = [x, y, 1]$, where the coordinate z is made equal to zero (without lost of generality, the plane is supposed to be on the xy -plane). Thus, the relation between the two points is:

$$\lambda \mathbf{p} = K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]\mathbf{P}, \quad (2)$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two columns of matrix R in (1). In this work, K is defined as:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where f is the camera's focal length and (u_0, v_0) are the principal point coordinates. The principal point is the intersection point of the camera's optical axis with the image plane. The camera model in (3) assumes that pixels are squares and their xy axes are perpendicular, a supposition that can be made in modern cameras.

2.1 The Proposed Method

As input, three different views of a unknown planar pattern are needed. It is supposed that from these three images, points of correspondences have been extracted (by a method that is not part of this discussion); these points mark the position of a same characteristic (for example, corner points) viewed on the three images.

Suppose also that we have one estimation of the focal length, $\hat{f}$, the principal point, $(\hat{u}_0, \hat{v}_0)$, and the orientation, $(\hat{\theta}_1^i, \hat{\theta}_2^i, \hat{\theta}_3^i)$, and position, $(\hat{t}_1^i, \hat{t}_2^i, \hat{t}_3^i)$, for each of the three views, $i = 1, 2, 3$. Then, it is possible to calculate a model plane, this is, a set of points $\hat{\mathbf{P}}_j$ using (2), where j is equal to the number of correspondence points on the three initial images. From (2), points $\hat{\mathbf{P}}_j = [\hat{x}_j, \hat{y}_j, 1]^T$ can be calculated by removing the scale factor λ , thus for a point (u, v) over one image, two equations as the following can be formed:

$$\begin{aligned}(um_{31} - m_{11})x + (um_{32} - m_{12})y &= m_{13} - um_{33} \\ (vm_{31} - m_{21})x + (vm_{32} - m_{22})y &= m_{23} - vm_{33}\end{aligned}$$

where m_{ij} are the elements of matrix $M = K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]$ (see Eq. (2)). Therefore, six equations can be formed, and this overdetermined system is solved using normal equations.

Here an important point: in calibration methods as [5,4] the plane model is given, in the self-calibration method the plane model is assumed to be unknown. Calibration methods are based on homographies and the self-calibration in [6] is based on inter-image homographies. An homography is the transformation mapping between two plains and it is represented by a 3×3 matrix. Homography is invariance to the scale and therefore it has only eight degrees of freedom. The proposed method here does not use homographies, therefore to impose scale invariance, the reconstructed plane is centered at its centroid and normalized arbitrarily to 0.01σ , where σ is the RMS standard deviation of all x and y values of the reconstructed model plane points.

Once the model plane is obtained and normalized, the reprojection error can be calculated as:

$$\sum_{i=1}^3 \sum_{j=1}^n \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}(\hat{f}, \hat{u}_0, \hat{v}_0, \hat{\theta}_1^j, \hat{\theta}_2^j, \hat{\theta}_3^j, \hat{t}_1^j, \hat{t}_2^j, \hat{t}_3^j, \hat{\mathbf{P}}_j)\|^2 \quad (4)$$

where $\mathbf{p}_{ij}$ is the given point j , $1 \leq j \leq n$, on image i , $1 \leq i \leq 3$, and $\hat{\mathbf{p}}_i$ is the estimated point obtained from (2).

The described problem is solved with the evolutionary algorithm called Differential Evolution (DE). As an evolutionary algorithm, DE works with a population, or a set of individuals; this population evolve by mutating and selecting the best individuals, and the process stops after a given number of iterations or when other stop condition is reached. Each individual codes a possible solution for the problem, in this problem an individual is a vector of real values of size 18, which store the parameters to estimate: f , $(\theta_1^i, \theta_2^i, \theta_3^i)$ and (t_1^i, t_2^i, t_3^i) , for

$i = 1, 2, 3$. The principal point (v_0, u_0) , can not be estimated from three images, therefore it is fixed in the image center coordinates.

To fix the orientation of the reconstructed plane, all three image orientations are respect to $\theta_1^1 = 0$, thus this parameter is not part of an individual.

From two projections of a plane, a common rotation axis exists. The dihedral angle can be fixed if a third projection of that plane is available. This is the reason why it is necessary three images of a plane to obtain its reconstruction. This form to obtain a reconstruction is a old procedure used in computer tomography [9].

3 Differential Evolution

The population of DE is composed by a set of individuals or vectors of real numbers. All vectors are initialized with random numbers with an uniform distribution within the search bounds of each parameter.

There are several version of DE. Here the rand/1/bin version of DE is used because it is robust and provides the best results for different kind of benchmarks and real optimization problems [10].

Nowadays exists lot of literature about DE, reader can use [11] for a good starting point about deeply DE details.

Algorithm 1. Differential evolution algorithm (rand/1/bin version)

Require: The search domain and the value s for the stop condition. The values for population size, μ ; maximum number of generations, M ; difference and recombination constants, F and R respectively.

Ensure: A solution of the minimization problem

```

1: initialize( $X = \{\mathbf{x}_1, \dots, \mathbf{x}_\mu\}$ )
2: evaluate( $X$ )
3:  $k = 0$ 
4: repeat
5:   for  $j = 1$  to  $\mu$  do
6:     Let  $r_1, r_2$  and  $r_3$  be three random integers in  $[1, \mu]$ , such that  $r_1 \neq r_2 \neq r_3$ 
7:     Let  $i_{\text{rand}}$  be a random integer in  $[1, n]$ 
8:     for  $i = 1$  to  $n$  do
9:        $x'_{i,j} = \begin{cases} x_{i,r_3} + F(x_{i,r_1} - x_{i,r_2}) & \text{if } U(0, 1) < R \text{ or } i = i_{\text{rand}} \\ x_{i,j} & \text{otherwise} \end{cases}$ 
10:     $x'_{n+1,j} = \text{evaluate}(\mathbf{x}'_j)$ 
11:    if  $x'_{n+1,j} < x_{n+1,j}$  then
12:       $\mathbf{x}_j = \mathbf{x}'_j$ 
13:   $\min = x_{n+1,1}, \max = x_{n+1,1}$ 
14:  for  $i = 2$  to  $n$  do
15:    if  $x_{n+1,i} < \min$  then
16:       $\min = x_{n+1,i}$ 
17:    if  $x_{n+1,i} > \max$  then
18:       $\max = x_{n+1,i}$ 
19:   $k \leftarrow k + 1$ 
20: until  $(\max - \min) < s$  or  $k > M$ 

```

The pseudocode of DE is shown in Algorithm 1. The core of DE is in the loop on lines 8-12: a new individual is generated from three different individuals chosen randomly; each value of the new vector (it represents a new individual) is calculated from the first father, plus the difference of the other two fathers multiplied by F , the difference constant; the new vector value is calculated if a random real number (between zero and one) is less than R , the DE's recombination constant. To prevent the case when the new individual is equal to the first father, at least one vector's component is forced to be calculated from their fathers values, it is in line 9 of the pseudocode, when $i = i_{\text{rand}}$, and i_{rand} is a integer random number between 1 and n . Then the new individual is evaluated, if it is better than the father (in lines 11-12), then the child replaces its father. The stop condition used here is: if the number of iterations is greater than 10,000, or when the difference in the objective function values of the worst and best individuals is less than 0.001. This stop condition is called *diff* criteria in [12], and is the recommended for a global optimization task.

According to the test in CEC 2005 conference [13], DE is the second best heuristic to solve real parameter optimization problems, when the number of parameters is around 10. The best heuristic is a Evolution Strategy called G-CMA-ES [14]. DE was chosen because it has a better execution time and it is very easy to implement

4 Experiments and Results with Real Data

Three experiments are carry on public available dataset of Zhang [15]. It consist of five images of a planar calibration pattern of 256 corners, taken with a CCD camera (resolution of 640×480). For each image, the corners positions found by Zhang are available. This data set is rather challenging for plane-based self-calibration. Indeed, there are few images (five), small rotations around optical axis, the plane orientation does not vary much, and a significant lens distortion is present.

Fist experiment, is performed with Zhang's images with lens distortion corrected. The camera parameters given in Zhang's paper [4] are $f = 832.53$, $(u_0, v_0) = (303.96, 206.59)$ and $k_1 = -0.288$, $k_2 = 0.190$ for the fist two terms of lens radial distortion. With this information a new set of five corners positions, but now without lens distortion, were generated (considering that the aspect ratio is equal to 1.0).

The search bounds for DE are $-90^\circ \leq \theta_1, \theta_2 \leq 90^\circ$, $-180^\circ \leq \theta_3 \leq 180^\circ$, $-50 \leq t_1, t_2 \leq 50$, $100 \leq t_3 \leq 1000$, $100 \leq f \leq 2000$, (u_0, v_0) is fixed to $(320, 240)$ (the image center). Here the convention $R = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ is used.

From the five sets of corners points, each one corresponding to one image, there exists 10 combinations of three images. The mean and standard deviation for f for each three images calculated with the proposed algorithm is shown in Table 1. The shown statistics are for 40 executions. On each execution seven runs of the algorithm is made and the median, according the reprojection error, is taken. For this problem a population of 50 individuals, difference constant equal to 0.7, and recombination constant of 0.9 are used.

Table 1. Results for the first experiment with real data. s.d. is standard deviation, r.e. reprojection error. Numbers on first column are three indexes corresponding to each image.

Triplet	f	f s.d.	r.e.	r.e. s.d.
123	816.63	0.38	0.0623	7×10^{-6}
124	836.94	9.90	0.0895	0.13
125	842.09	0.24	0.0262	9×10^{-6}
134	820.86	0.34	0.0451	6×10^{-6}
135	821.56	0.28	0.0491	7×10^{-6}
145	847.04	1.00	0.0175	5×10^{-6}
234	828.71	0.18	0.0681	1×10^{-5}
235	829.78	0.24	0.0745	1×10^{-5}
245	848.83	1.10	0.0173	1×10^{-5}
345	833.40	0.45	0.0552	7×10^{-6}
Total	832.58	11.06	0.0505	0.0478

Table 2. Results for the second experiment. The third column is the value of f obtained with the bundle adjustment of all the five images.

Triplet	Initial f	Final f	% error
123	816.57	828.03	-0.54
124	842.01	826.73	-0.70
125	842.14	826.96	-0.67
134	820.04	828.11	-0.53
135	821.68	828.69	-0.46
145	847.19	827.27	-0.63
234	828.31	827.18	-0.64
235	829.97	827.38	-0.62
245	848.38	827.80	-0.57
345	832.91	827.57	-0.60

The second experiments is using the five images. From every triplet of images in Tab. 1, the other two images are added minimizing (4) for $i = 4, 5$ also with DE. The orientation and position of these two new images are calculated fixing both f and the reconstructed plane ($\hat{\mathbf{P}}_j$ in (4)). To solve this problem the same conditions are used except that the number of individuals is set to 30. Now all the parameters are refined using Bundle Adjustment: this is, (4) is minimized with $i = 1, 2, \dots, 5$ with the Levenberg-Marquardt method provided by MINPACK (lmdif1 function) as Zhang suggested in his paper [4]. the initial solution obtained with DE is a local optimum. In order to obtain the global optimum, t_1^i and t_2^i , $i = 1, 2, \dots, 5$, with values less than 10.0 are set to 0.0 and the bundle adjustment (five steps, or when RMS reprojection error is less than 0.075) is executed. The result is shown in Table 2. The error in the estimation of the focal length is less than 0.7 %. All the bias in error are negatives, and this could be due the skew was made equal to 0.0 instead its real value in this experiment.

The third experiment performs the same calculations that the second experiment but using the raw original data and starting with the principal point fixed at the images center. In the bundle adjustment, the correction of the lens distortion is added, and the principal point values are also refined. Results and errors vs the values obtained by Zhang are shown in Tab. 3.

The camera focal length is estimated with an error less than 1.3% from the results of the third experiment. But the principal point position has an error less than 8%, and error is even more for the lens distortion parameters. According to the simulation results in [6], the focal length is not affected by the principal point position, but the orientation and positions of the involved plane views are affected. Therefore, from this result, it can be seen that the principal point position can not be estimated with accuracy if a reference plane model –which is a calibration method– is not used.

Table 3. Results of the third experiment. Initial values for the principal point are $(u_0, v_0) = (320, 240)$, and distortion parameters $k_1 = 0$, $k_2 = 0$. Final values were obtained with all five images.

Triplet	Initial value f	Final values					Percentage in error				
		f	u_0	v_0	k_1	k_2	f	u_0	v_0	k_1	k_2
123	894.96	843.21	294.07	208.58	-0.2145	0.2704	1.28	-3.25	0.96	-5.93	42.34
124	814.28	837.39	316.71	206.26	-0.2154	0.1865	0.58	4.19	-0.16	-5.51	-1.86
125	824.02	841.15	294.75	196.13	-0.1863	0.1926	1.04	-3.03	-5.07	-18.30	1.35
134	878.45	838.20	293.54	195.75	-0.2117	0.2671	0.68	-3.43	-5.25	-7.17	40.60
135	879.89	834.27	292.22	195.65	-0.2183	0.2620	0.21	-3.86	-5.29	-4.27	37.88
145	854.62	836.63	295.17	196.71	-0.2208	0.3021	0.49	-2.89	-4.78	-3.17	58.98
234	854.62	840.37	293.87	192.25	-0.2039	0.2410	0.94	-3.32	-6.94	-10.59	26.84
235	857.77	833.33	293.66	190.93	-0.2167	0.2559	0.10	-3.39	-7.58	-4.95	34.68
245	803.48	836.32	294.60	196.86	-0.2217	0.3003	0.46	-3.08	-4.71	-2.76	58.04
345	832.75	833.38	291.79	193.28	-0.2171	0.2546	0.10	-4.00	-6.45	-4.79	34.02

5 Discussion

The convergence of a heuristic is at most linear. Convergence of algorithms based on derivatives is quadratic. Other heuristics could be used to solve non-linear optimization problems, but their theoretical convergence limit is always linear. This argumentation lead to the idea that an heuristic should not be used instead a conventional numerical algorithm. In the methodology proposed in this work, the heuristic differential evolution solves the difficult task of finding an initial reconstructed plane, from three images; and then this reconstructed plane is used to add more images, and perhaps, to correct the camera lens distortion. For these last two tasks, conventional bundle adjustment was used.

From the result of the third experiment is not clear that the proposed method can be used to correct the lens distortion. But the method can be applied on images taken with a modern digital camera where distortion can be ignored [6].

The proposed method solves directly a non-linear optimization problem, and its performance is good under noisy conditions (1-2 pixels) in points positions.

A run of the DE algorithm with the same conditions used in experiments (50 individuals, $s = 0.001$) takes approximately 18 sec ¹.

6 Conclusions

A method for direct (without using derivatives) self-calibration of a camera from three images of an unstructured plane was presented. This method uses directly the pinhole camera model to estimate the positions, the orientation and the camera parameters of the three views, and also obtains the reconstructed plane. The reprojection error is minimized. This non-linear optimization problem was solved using the heuristic Differential Evolution. Moreover, more images can

¹ Time was measured in a 1.8 GHz PowerPC G5 Mac with Mac OSX and gcc compiler.

be incorporated, using the reconstructed plane as a model plane. The whole structure then is refined with a conventional bundle adjustment.

The solution with a heuristic has a cost: running time is high compared with a conventional algorithm such as Levenberg-Marquardt, but the advantage is that a starting solution, near to the optimal solution, is not necessary to solve the non-linear problem.

Acknowledgments. This work was partially supported by CONACyT, México under grant 80965.

References

1. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge Uni. Press, Cambridge (2003)
2. Heyden, A., Astrom, K.: Flexible calibration: minimal cases for auto-calibration. In: ICCV-1999, pp. 350–355. IEEE Press, Los Alamitos (1999)
3. Hemayed, E.E.: A survey of camera self-calibration. In: AVSS 2003: Proc. IEEE Conf. on Advanced and Signal Based Surveillance. IEEE Press, Los Alamitos (2003)
4. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on Patt. Anal. & Mach. Intel. 22, 1330–1334 (2000)
5. Sturm, P.F., Maybank, S.J.: On plane-based camera calibration: A general algorithm, singularities, applications. In: CVPR, pp. 432–437 (1999)
6. Menudet, J.F., Becker, J.M., Fournel, T., Mennessier, C.: Plane-based camera self-calibration by metric rectification of images. Image and Vision Computing 26, 913–934 (2008)
7. Bocquillon, B., Gurdjos, P., Crouzil, A.: Towards a guaranteed solution to plane-based self-calibration. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 11–20. Springer, Heidelberg (2006)
8. Storn, R., Price, K.V.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11(4), 341–359 (1997)
9. Herman, G.T.: Image Reconstruction from Projections: The Fundamentals of Computerized Tomography. Academic Press, London (1980)
10. Mezura, E., Velázquez, J., Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: GECCO 2006, New York, NY, USA, pp. 485–492. ACM Press, New York (2006)
11. Chakraborty, U.K.: Advances in Differential Evolution. Studies in Computational Intelligence. Springer, Heidelberg (2008)
12. Zielinski, K., Laur, R.: Stopping criteria for differential evolution in constrained single-objective optimization. In: Advances in Differential Evolution. Springer, Heidelberg (2008)
13. Hansen, N.: Comparisons results among the accepted papers to the special session on real-parameter optimization at CEC 2005 (2006), http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/compareresults.pdf
14. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proceedings of the Congress on Evolutionary Computation CEC-2005, pp. 1769–1776 (2005)
15. Zhang, Z.: <http://research.microsoft.com/~zhang/calib/>