Monocular Point Based Pose Estimation of Artificial Markers by Using Evolutionary Computing

Teuvo Heimonen and Janne Heikkilä

University of Oulu, Information Processing Laboratory, Linnanmaa, Po Box 4500, 90014 University of Oulu, Finland
teuvo.heimonen@ee.oulu.fi
http://www.ee.oulu.fi/mvg/mvg.php

Abstract. Evolutionary computation techniques are being increasingly applied to a variety of practical and scientific problems. In this paper we present a evolutionary approach for pose estimation of a known object from one image. The method is intended to be used in pose estimation from only a few model point - image point correspondences, that is, in cases in which traditional approaches often fail.

1 Introduction

Pose estimation based on model point - image observation correspondences is an essential problem in many computer and robot vision applications. In our special interest are applications of computer aided surgery, where pose of, for example, surgical instrument with respect to the patient need to be determined accurately and robustly. In these applications it has been noted sensible to attach reflective, spherical fiducials rigidly to the object and thus obtain reliable (point) observations from different imaging directions.

Several different approaches to the point correspondence based pose estimation problem has been reported both in photogrammetry and computer vision literature (see. e.g. [1,2]). The proposed methods can be roughly divided to two groups: analytical (see. e.g. [3,4,5]) and iterative (see e.g. [6,7,5,8]). General deficiency with the analytic methods is the high sensitivity to the noise [5]. Because this the pose estimation results may be far from true values. Iterative approaches, on the other hand, need typically a good initial pose estimate. Even if the initial estimate is near the true solution, which is not always the case, these methods may find only a local optimum of the highly multi-modal solution space of the pose estimation problem [9], and thus not produce reasonable solution.

Evolutionary algorithms (EA), have also been applied to the pose estimation from one image [10,11,12,13]. They are reported to offer the advantages like autonomy, robustness against diverging and local optimums, and better noise and outlier immunity [9,11]. Common to the previous EA-approaches is that they have used six genes (parameters) of a chromosome (solution candidate) to

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 122–131, 2007.

define pose: three for position and three for orientation. In the previous EA-based pose estimation papers the usage of minimum or near minimum number of point correspondences has not been discussed.

In order to robustly estimate the pose of an artificial marker comprising only a few, usually coplanar, point fiducial, we propose here a two-phase, geometrically constrained approach in which

1. Mathematical framework of the problem is formulated so that minimal number of genes are needed, and solution space constrains are inbuilt.
2. Initial search space of the genes is not strictly limited. The restriction of the search space is performed by the algorithm.

The correspondence of the model points and image observations is solved automatically in the first phase of the approach using index genes like in [13]. Similar to [11] we use real number presentation for the pose genes and apply Gaussian mutator and Blend crossover as the genetic operators. We also utilize kick-out genetic operator suggested in [12]. During our procedure no other minimization technique than the evolution is used.

The rest of the paper is organized as follows: In section 2 mathematical framework of our method is presented. In section 3 outline of our genetic algorithm approach is described and genetic operators are presented. In section 4 test and results of these test are presented and paper is concluded in section 5.

2 Mathematical Framework of the Method

The pose estimation problem can be formulated as follows (see also Fig. 1 a)): Determine the rigid transformation between camera coordinate frame (**C**) and model coordinate frame (**M**) when some number of image observations $p_i^C = [u_i, v_i, f]^T$ and corresponding model points $P_i^M = [X_i, Y_i, Z_i]^T$ are available. The coordinates of a model point in the camera frame $P_i^C = [x_i, y_i, z_i]^T$ can be obtained from the rigid transformation

$$P_i^C = RP_i^M - T, \tag{1}$$

where R is a 3x3 rotation matrix and T 3x1 translation vector. The collinear relation of a model point P_i^C and its image p_i^C can be presented by

$$k_i p_i^C = P_i^C, \tag{2}$$

where k_i is scalar coefficient. By combining (1) and (2) we can compute the actual image coordinates in **C** from the model coordinates in **M** with

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{f}{r^3 P_i^M - t_z} \begin{bmatrix} r^1 P_i^M - t_x \\ r^2 P_i^M - t_y \end{bmatrix}, \tag{3}$$

where f is the focal length known from the camera calibration (throughout of this paper we assume that intrinsic camera calibration parameters are known), r^i is the row i of R, and $[t_x, t_y, t_z]$ are the components of T.

According to (2) the coordinates of a model point P_i^C in camera frame can be, in a noise-free case, obtained from the image observations by multiplying the image observations by some factor k_i. The distance between two model points P_i and P_j should naturally be the same both in the model and in the camera frames:

$$D_{ij} + \eta = d_{ij} = \left\| k_i p_i^C - k_j p_j^C \right\| = \sqrt{k_i^2 (p_i^C)^T p_i^C - 2k_i k_j (p_i^C)^T p_j^C + k_j^2 (p_j^C)^T p_j^C}, \tag{4}$$

where $i = 1, .., n-1, j = 2, .., n, i \neq j$, D_{ij} is the known distance between the model points P_i^M and P_j^M, η is a noise term, and d_{ij} the distance between the computed points P_i^C and P_j^C. The distances D_{ij} are assumed to be exactly known. From n points $\frac{(n-1)n}{2}$ equations like (4) are obtained. The principal task of the pose estimation in this formulation is to solve coefficients k. Our approach is to fix one of the coefficients k and then solve the others from the equations (4). Fixing one of the coefficients k, let say k_i, gives us P_i^C (eq. 2) and two solutions for k_j (eq. 4). Related to the distance (d) between P_i^C and the ray from the projection center through the image point p_j^C to the infinity three cases for the solutions for k_j are possible (see Fig. 1b - 1d). If

1. $d < D_{ij}$, two different solutions for k_j are obtained. Other points available, say $P_{m \neq i,j}^C$, are considered in order to select the more feasible one of these two solutions.
2. $d = D_{ij}$, two equal solutions for k_j are obtained.
3. $d > D_{ij}$, two complex solutions for k_j are obtained. In this case we use $k_j = \frac{(u_i u_j + v_i v_j + f^2) k_i}{u_j^2 + v_j^2 + f^2}$ that yields minimum d_{ij} on condition $P_j^C = k_j \left[u_j, v_j, f\right]^T$.

Typically the 3D pose is defined with six parameters, three for position or translation $[t_x, t_y, t_z]$ and three for orientation or rotation $[\omega, \varphi, \kappa]$. In our approach these six pose parameters are extracted from the rotation matrix R and translation vector T of the rigid transformation (1) after the coefficients k_i and thus the model points P_i^C are obtained. We solve the rigid transformation

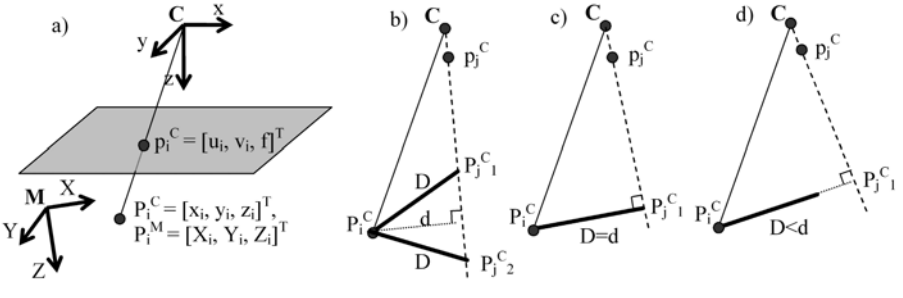

Fig. 1. a): Pose estimation framework. Because of the uncertainty in the image observations p_i^C and p_j^C b) two, c) one, or d) no real solution for the P_j^C is obtained depending on the distance D and the direction of the ray from **C** through p_j^C to infinity.

between the model coordinates in the model frame P_i^M and in the camera frame P_i^C by using method presented in [14].

3 Evolutionary Algorithm Approach

In evolutionary algorithms, a population of candidate solutions of the problem (chromosomes) is submitted repeatedly to the genetic operators (selection, reproduction, crossover and mutation) in order to create new generation of chromosomes with improved fitness with respect to the problem. We propose here a two-phase approach. In the first phase of our approach the correspondence of the points and a coarse estimate for k_1 are solved. In the second phase correspondence is assumed to be known and the final estimate for k_1 is searched for by letting also the image observations vary according to some predefined (noise) distribution. The chromosome encoding and the genetic operators, which are to be presented next, are partly different in the two phases.

3.1 Chromosome Encoding

In the first phase only the coefficient k_1 and the order of the image observations are revealed to the genetic operators. Thus we use chromosomes that comprise the coefficient (real number) and only an index list of the image observations. During the procedure each index (integers $1, 2, .., n$) may and should occur exactly once in the index list. So we have $n + 1$ genes in a chromosome (Fig. 2a). In the second phase the genes are the coefficient k and the image observations $[u_i, v_i]$. In this phase all of the genes can be varied. Thus the a chromosome includes $2n + 1$ real numbered genes (see Fig. 2b).

Fig. 2. Chromosome encoding a) in phase 1 and b) in phase 2

3.2 Initialization

In the first phase, the value of the first gene of each initial chromosome are randomly generated according to uniform distribution between such a values that the object can be anywhere between certain distance (along optical axis) from the camera center. The integer indexes of the image observations $[u_i, v_i]$ are initialized into random order. In the second phase, we limit the search base so that the gene values obey isotropic Gaussian distribution around the best solution from the first phase. Different standard deviation of the distribution may be used for different genes.

3.3 Genetic Operators

Cross-over. Cross-over occurs with a constant probability in both phases. The first gene in the first phase and all genes in the second phase of the off-springs resulting from the cross-over operation are linear combination of the corresponding genes of the parents (Blend cross-over). The weight of the linear combination (λ) is randomly generated real number between 0 and 1. Mathematically,

$$gene_i^{offspring} = (1 - \lambda) gene_i^{father} + \lambda gene_i^{mother}. \tag{5}$$

For other genes in the first phase (indexes) partially matched cross-over (PMX) is applied in order to guarantee that all indexes are found exactly once in each offspring.

Mutation. Mutation occurs with a constant probability in both phases. The first gene in the first phase of the off-springs resulting from the mutation operation are random values from uniform distribution used also in the initialization phase. There is no mutation operation for other genes of the first phase (indexes). In the second phase the value of the first gene of the off-springs resulting from the mutation operation obeys Gaussian distribution around the best solution found in the first phase. Other genes are from the Gaussian distribution around the initial image observations. The standard deviations of these distributions are predetermined constants.

Selection. Parents for the reproduction are selected by using fitness-proportional tournament selection. In this method two chromosomes picked randomly from the population compete and the fitter is selected. In every generation a certain percent of the fittest choromosomes in current population are selected to form a basis for the population of the next generation (elite selection). Also off-springs are evaluated and the fittest of them are selected to fill in the population for the next generation. In this step of the algorithm the population is also pruned so that any certain chromosome occurs no more than once in the population. If the size of the population is about to decrease because of this pruning, we add new chromosomes into it. These new chromosomes are such that their genes are a little and randomly variated (in maximum $\pm 1\%$ of the current search space dimension) from some randomly chosen chromosome already in the population.

Kick-out. Kick-out occurs if the best solution has not changed in certain number of generations. In the kick-out operation population is re-initialized.

3.4 Fitness

In the first phase the fitness of a chromosome is the inverse of the sum of squared differences of the distances between the model points (D_{ij}) and back-projected points (the right-hand side of eq. (4)), that is

$$fitness^{-1} = \sum \left(D_{ij} - \left\| k_i p_i^C - k_j p_j^C \right\| \right)^2. \tag{6}$$

In order to evaluate the fitness of a chromosome, other coefficients k_j needed to back-project all the image observations to model coordinates are first solved (see section 2).

In the second phase the fitness of a chromosome is the inverse of the sum of distances between the image observations p_i^C and the model points projected to the image plane with (3). In order to incorporate the possible uncertainty knowledge of the p_i^C:s, we use the Mahalanobis distance as a distance measure. So the fitness measure in the second phase is

$$fitness^{-1} = \sum \left((p_i^C - p_i'^C)^T \Sigma^{-1} (p_i^C - p_i'^C) \right), \tag{7}$$

where $p_i'^C$ are the projected image coordinates. Different uncertainty estimates (variances) can be used for every p_i^C and also for x- and y-components of any p_i^C using the covariance matrix Σ obtained e.g. from the image feature extraction procedures. In our experiments we used identity matrix as Σ.

3.5 Stop Criteria

We use two different criteria in order to stop the evolution. The first one is the number of generations evaluated: both minimum and maximum number of generations are limited. The second criterion is the fitness of the best solution candidate: if this fitness is better than a specific limit the evolution is ended. Different values for these parameters is used in the first and the second phase of the algorithm.

4 Experiments

The method presented in this paper was evaluated with both synthetic and real image data. In both cases the basic test procedure was the following: The marker was positioned in some position and orientation in the field-of-view of the camera. The image coordinates were either computed according to a pinhole camera model (synthetic data) or extracted from the image of the real camera (real data).

The noisy image coordinates were inputted to the "Extcal" pose estimation method implemented in the calibration toolbox [15] and to the EA-method presented in this paper. The Extcal-method uses DLT for the pose initialization and Levenberg-Marquardt minimization for the refinement, and it can be considered as a traditional bundle-adjustment approach for pose estimation. However, this method does not include correspondence determination and thus the pose estimation with this method was performed using correct point correspondences.

The parameters used in the EA-method are presented in table 1. Different empirically determined fitness limits for the stop criteria were used for different amount of points, for example limits of 0.8 and 0.00001 mm were used for the four point cases for the phase 1 and phase 2 stop criteria, respectively. The search space was limited with $k_1 = [80, 300]$ (absolute values) in the first phase and the standard deviations of 4 and 2*noise standard deviation for k_1 and u_i and v_i, respectively.

Table 1. EA parameters, * = same value for both phases was used

parameter	value
Population size*	150
Minimum number of generations*	20
Maximum number of generations: Phase 1	100
Maximum number of generations: Phase 2	300
Kick-out frequency: Phase 1	5
Kick-out frequency: Phase 2	20
Cross-over probability: Phase 1	0.5
Cross-over probability: Phase 2	0.9
Mutation probability: Phase 1	0.5
Mutation probability: Phase 2	0.5
Elite selection probability: Phase 1	0.3
Elite selection probability: Phase 2	0.5
Selection method*	Tournament
Population initialization distribution: Phase 1	Uniform
Population initialization distribution: Phase 2	Gaussian
Mutation distribution: Phase 1	Uniform
Mutation distribution: Phase 2	Gaussian

4.1 Synthetic Data

In the synthetic data tests a virtual marker with three, four, or five points was positioned randomly inside the field-of-view of the camera and a volume with limits t_x = [-500, 500] mm, t_y = [-500, 500] mm, and t_z = [2000, 8000] mm. The rotations of the marker were limited to ω = [-70, 70] deg, φ = [-70, 70] deg, and κ = [-180, 180] deg. The size of the virtual image sensor was [4.8, 3.6] mm.

Table 2. Average pose parameter errors with different methods. The number after the method name indicates the number of points used in the pose estimation. The symbol 4b stands for the case where 4 four points were used and the Extcal-method failed (The Extcal-estimate was doomed to be a failure when the sum of the distances between image observations and the image coordinates computed with the estimated pose parameters was more than 20 times bigger than the standard deviation of the noise). Tabulated values are mean errors of all measurements (60 repeats x 9 noise levels = 540 measurements) of test in question.

Test	t_x [mm]	t_y [mm]	t_z [mm]	ω [deg]	ϕ [deg]	κ [deg]
Extcal-3	0.917	1.078	28.511	1.941	1.778	1.196
EA-3	0.898	0.997	26.781	2.068	1.738	1.224
Extcal-4	0.277	0.305	7.650	0.264	0.177	0.246
EA-4	0.260	0.275	6.950	0.213	0.151	0.137
Extcal-4b	1.853	2.417	80.900	4.352	3.335	3.866
EA-4b	0.343	0.389	7.338	0.399	0.214	0.341
Extcal-5	0.263	0.300	7.496	0.217	0.188	0.231
EA-5	0.288	0.311	8.027	0.203	0.167	0.139

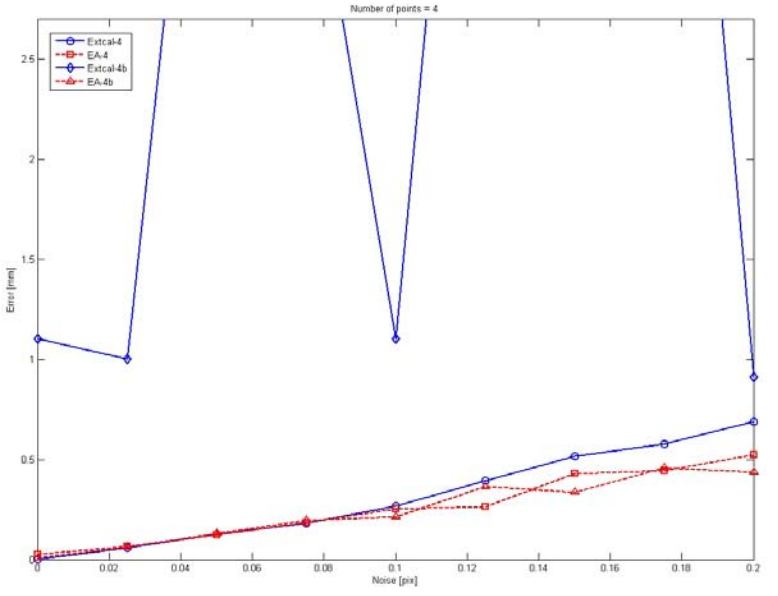

Fig. 3. Average errors in x and y directions with different methods. It should be noted that in the Extcal-4b and EA-4b cases the input data was chosen such that the Extcal-method failed. The EA-method succeed to remain robust also with these kinds of input data.

The model coordinates of the points in a marker were [0 0 0;0 50 0;113 0 0], [0 0 0;0 108 0;100 130 0;113 0 0], [0 0 0;0 108 0;100 130 0; 100 50 0;113 0 0] mm for three-, four-, and five-point marker, respectively. Model coordinates were projected to a image plane according to a pin-hole camera model with 25 mm focal length. Gaussian noise with standard deviations 0, 0.025, 0.05, ..., 0.2 pixels was added to the obtained image coordinates. For each noise level 50 random poses were evaluated.

From the results of this test it was observed that the methods succeeded almost equally in general, but in some special cases in which Extcal-method failed (the failure is caused by the faulty initial pose estimate obtained with the DLT), EA-method still performed satisfactorily. Compiled statistics of the results are presented in Table 2 and example plot is given in Fig. 3.

4.2 Real Images

A marker with four points (a passive planar rigid body from Northern Digital Inc.) was attached to a machine tool and moved to 19 poses. The poses were such that the marker was either translated in the x- direction (limits [-300, 300] mm) or rotated about y-axis ([-55, 40] deg). In every pose target was halted and 15 measurements were executed. One measurement comprised acquiring coordinates by using NDI Polaris tracking system and an image by using Sony

Table 3. Average errors in the position of the rotation axis and the rotation about it between different poses

Test	t_x [mm]	t_y [mm]	t_z [mm]	ω [deg]	ϕ [deg]	κ [deg]
Extcal-4	2.015	2.236	2.763	0.412	1.498	0.277
EA-4	2.036	2.089	2.776	0.461	1.490	0.187

XCD-X710 camera with Rainbow TV zoom lens. The model coordinates of the points in a marker were [0 0 0;0 108 0;100 130 0;113 0 0] mm.

From the images the centroids of the spherical markers were extracted by simply thresholding the image and by computing the center of masses of the spheres. The repeatability of the image formation and feature extraction was computed by comparing the image coordinates extracted from the images of same pose. The standard deviation of the extracted image coordinates was 0.030 pix.

The pose of the marker was estimated from every image using again both Extcal- and EA-methods. The input parameters of the EA-method were same as in synthetic data tests (see Tables 1), except that 0.030 pix for the noise standard deviation was used. The motion between of different poses was determined and compared to the known values and the motion information obtained with the Polaris tracker.

As in the synthetic data tests also here almost similar performance was observed (Table 3). The significantly larger errors especially in t_x, t_y, and ϕ in these tests than in the synthetic data tests was observed to be caused by the inaccuracy of the camera calibration and thus deficient correction of the geometric distortion.

5 Discussion

We proposed here a pose estimation method based on evolutionary computing. The method proved to be robust and reliable. It does not need accurate initial guess and finds point correspondences automatically.

We presented some examples of the test results with three, four, and five coplanar points. It should be noted that the proposed method can be directly used with 3D model points as well. The performance of the method is similar than we have demonstrated with the coplanar points.

As the evolution based methods in general, also the method proposed here is slow compared to analytical or more traditional, iterative pose estimation methods (computation of a new generation of ten chromosomes in our EA-method is about equal to the computation of one iteration in the Extcal-method). However, we believe that evolution based pose estimation is a feasible option for pose estimation in any application without a strict demand for real-time performance.

References

1. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, MA (1993)
2. Horaud, R., Dornaika, F., Lamiroy, B., Christy, S.: Object pose: The link between weak perspective, paraperspective and full perspective. International Journal of Computer Vision 22(2), 173–189 (1997)
3. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
4. Horaud, R., Conio, B., Leboulleux, O., Lacolle, B.: An analytic solution for the perspective 4-point problem. CVGIP 47, 33–44 (1989)
5. Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., Kim, M.: Pose estimation from corresponding point data. IEEE Transactions on Systems, Man. and Cybernetics 19(6), 1426–1446 (1989)
6. Oberkampf, D., DeMenthon, D.F., Davis, L.S.: Iterative pose estimation using coplanar feature points. Journal of Computer Vision and Image Understanding 63(3), 495–511 (1996)
7. Lowe, D.: Fitting parameterized three-dimensional models to images. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(5), 441–450 (1991)
8. Heikkilä, J.: Geometric camera calibration using circular control points. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1066–1077 (2000)
9. Ji, Q., Zhang, Y.: Camera calibration with genetic algorithms. SMC-A. 31(2), 120–130 (2001)
10. Toyama, F., Shoji, K., Miyamichi, J.: Model-based pose estimation using genetic algorithm. In: ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition, vol. 1, p. 198. IEEE Computer Society, Washington DC, USA (1998)
11. Hati, S., Sengupta, S.: Robust camera parameter estimation using genetic algorithm. Pattern Recogn. Lett. 22(3-4), 289–298 (2001)
12. C.Rossi, M.Abderrahim, J.C.Díaz: Evopose: A model-based pose estimation algorithm with correspondences determination. In: ICMA 2005: Proceedings of the IEEE International Conference on Mechatronics and Automation 2005, Niagara Falls, Canada, pp. 1551–1556 (2005)
13. Yu, Y., Wong, K., Chang, M.: Pose estimation for augmented reality applications using genetic algorithm. SMC-B. 35(6), 1295–1301 (2005)
14. Arun, K., Huang, T., Bolstein, S.: Least-Squares Fitting of Two 3-D Point Sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 698–700 (1987)
15. Heikkilä, J.: Camera calibration toolbox for matlab, `http://www.ee.oulu.fi/~jth/calibr/` (2000)