Evolutionary Parameter Optimization for Visual Obstacle Detection

Thomas Bergener [a], Carsten Bruckhoff [b], Christian Igel [c] *
Institut für Neuroinformatik, Ruhr-Universität Bochum 44780 Bochum, Germany
Tel: +49 234 700 7969 [a] / 5570 [b] / 5558 [c]
Fax: +49 234 709 4209
{Thomas.Bergener, Carsten.Bruckhoff, Christian.Igel}@neuroinformatik.ruhr-uni-bochum.de

Abstract

In this paper we employ an Evolutionary Algorithm (EA) to improve the parameters of a visual obstacle detection method called Inverse Perspective Mapping. *We show that the EA leads to a better parameter setting than the one found by an expert. The obstacle detection method is successfully implemented on our autonomous mobile robot* ARNOLD *to navigate in an unknown and dynamically changing environment in a fast and reliable manner.*

1 Introduction

The control of an autonomous mobile robot in an unknown and dynamically changing environment by an active stereo vision system needs a fast an reliable obstacle detection. For this task we use the *Inverse Perspective Mapping* (IPM) (Mallot et al., 1991; Bergener and Bruckhoff, 1999). This algorithm is based on a camera model. The quality of the IPM depends on the hard to determine internal and external camera parameters. Evolutionary algorithms (EAs), i.e. the family of algorithms that mimic mechanisms of natural evolution, have proven to be powerful tools for solving multi-modal, non-linear and non-differentiable optimization tasks (Bäck and Schwefel, 1993). In this article we employ an elaborated EA to find the parameters of our camera model with the required accuracy needed for the IPM.

The IPM has been implemented on our autonomous mobile Robot ARNOLD (see Figure 1) developed in the framework of NEUROS[1] (NEUral RObot Skills) (Bergener et al., 1999). We experimentally show that the evolved results are better than the results found by an expert who has implemented the IPM.

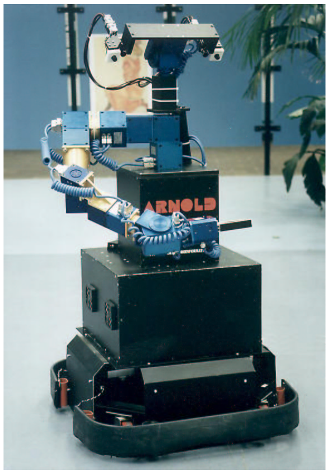


Figure 1: Arnold

In the next section, the IPM is introduced. In Section 3 we describe the employed EA. The results are shown in Section 4. The last section draws a conclusion.

2 Inverse Perspective Mapping

Visual obstacle detection for mobile robots can be achieved by observing the floor in front of the robot from two different camera perspectives and comparing the image contents. A solution of this correspondence problem by image correlation is suitable to construct a 3D representation of the scene and thus to detect obstacles. At the same time the method is computational expensive. Since we are not interested in a complete 3D representation but only in a segmentation into free and blocked areas of the floor plane, we can assume an obstacle free floor in front of the robot. A suitable mathematical camera model and complete knowledge of all internal and external parameters of the

[*] Authors are in alphabetical order.

[1] NEUROS is a research project supported by the German Ministry of Education and Research through grant number FKZ01IN504F

stereo camera system then allows to reconstruct the texture of the floor from the camera image. In the same way this texture can be mapped back to the camera image by a sequence of image transformations. Therefore, it is possible to predict the contents of the left camera image from the right camera image, at least for the so called binocular range, i.e. the part of the floor that is visible in both camera images. A comparison of the predicted content of the left camera image with an acquired image by simply subtracting the grey-level values shows deviations at all points where the hypothesis of a free way is violated. Thresholding the difference image leads to a segmentation of the camera image into obstacles and free space. This technique of a purposeful change of perspective is called *Inverse Perspective Mapping* (IPM) and is described in detail by Bergener and Bruckhoff (1999).

In practice it is crucial to find good values for the required camera parameters that are needed for the mapping, camera positions and rotations, exact focal length, optical image center, pixel size, horizontal scan ratio of camera and frame grabber and the coefficients of a radial lens distortion. All of these are hard to determine for off-the-shelf cameras and active vision heads. Especially when the system is stressed mechanically, e.g. by vibrations of the moving robot, a regular re-tuning of the parameters is necessary to achieve a small remaining error in the image mapping. An experienced user is able to improve the mapping by observing the difference image and fine tuning the several parameters since he qualitatively knows the effect of the different parameters. This work is unsatisfactory and time-consuming. Standard optimization methods like estimated gradient descent or simulated annealing turned out to be ineffective. This makes an alternative automatic and systematic technique for finding good parameter settings very attractive.

The obstacle detection module based on the IPM is successfully used in a navigation task (Bruckhoff and Dahm, 1998) where the robot moves in a cluttered indoor environment avoiding obstacles.

3 Evolution Strategy with Derandomized Self-Adaptation

3.1 Outline of the Algorithm

A typical EA starts with a parent population of individuals each representing a trial solution of the problem at hand. Each individual is assigned a fitness that is determined by the quality of the solution it represents. Here test pictures are used to calculate the fitness of a set of camera parameters. The quality of a solution is equal to the number of correctly matched pixels (see next section).

New solutions are generated by randomly altering the existing individuals. A selection mechanism that prefers solutions with better fitness values determines which individuals form the next parent apopulation. This (easily to parallelize) loop of creating new individuals from the parents, fitness evaluation, and selection is iterated until a termination criterion is fulfilled, e.g. a suitable solution is found or a certain amount of computation time has been consumed (see the work of Rudolph (1997) for an overview of general convergence properties of EAs).

A variety of EAs exists, in our application an *evolution strategy* (ES) is employed (Schwefel, 1995; Rechenberg, 1994). Each individual represents a real-valued vector, in this case 14 parameters (see section 4.1 for a description of the parameters) of our camera model. These so called object variables are altered (mutated) by adding a normally distributed random vector with zero mean. The covariance matrix of the mutation vectors is adapted to improve the search process. The so called *strategy parameters* that determine the mutation distribution are incorporated into the representation of each individual. *Derandomized self-adaptation of individual step-sizes and one direction* is used (Ostermeier, Gawelczyk, and Hansen, 1995; Hansen, Ostermeier, and Gawelczyk, 1995). This adaptation scheme has mainly the following advantages. Firstly, the change of the strategy parameters is based on the same realization of a random variable as the mutation of the object variables. It is guaranteed that the selection of a small or large mutation step-size leads to a corresponding decrease or increase of the step-size parameter. Secondly, each individual uses history information about its evolution path stored in the strategy parameters. Thirdly, the step-size variations are damped before they are transmitted to the descendants in order to avoid random fluctuations of the step-size parameters.

We use a (μ, λ)-selection scheme with one elitist. This means, the new parent population consists of the best individual found so far and the $\mu - 1$ best of the λ offsprings.

3.2 Details of the EA

The ES as proposed by Hansen, Ostermeier, and Gawelczyk (1995) works as follows:

1. An initial parent population of μ individuals is randomly created and evaluated.

2. From the parent population λ offsprings are created. For each $k = 1, \ldots, \lambda$:

   (a) The object variables are mutated. For each $i = 1, \ldots, n$:

$$x_i^{N_k} = x_i^{E\zeta_k} + \overbrace{\sigma_i^{E\zeta_k} z_i^k}^{\text{uncorrelated mutation}} + \overbrace{\sigma_r^{E\zeta_k} z_r^k r_i^{E\zeta_k}}^{\text{line mutation}} \tag{1}$$

   (b) The individual step sizes are adapted. For each $i = 1, \ldots, n$:

$$\boldsymbol{s}^{N_k} = (1-c) \cdot \boldsymbol{s}^{E\zeta_k} + c \cdot (c_u \boldsymbol{z}^k) \tag{2}$$

$$\sigma_i^{N_k} = \sigma_i^{E\zeta_k} \cdot \overbrace{\exp\{\beta(||\boldsymbol{s}^{N_k}|| - \hat{\chi}_n)\}}^{\text{“global” step-size adaptation}} \cdot \overbrace{\exp\{\beta_{\text{ind}}(|\boldsymbol{s}_i^{N_k}| - \hat{\chi}_1)\}}^{\text{individual step size-adaptation}} \tag{3}$$

   (c) The preferred direction vector is adapted:

$$s_r^{N_k} = \max\{0, (1-c) \cdot s_r^{E\zeta_k} + c \cdot (c_u z_r^k)\} \tag{4}$$

$$\boldsymbol{r}' = (1-c_r) \cdot \sigma_r^{E\zeta_k} \boldsymbol{r}^{E\zeta_k} + c_r \cdot (\boldsymbol{x}^{N_k} - \boldsymbol{x}^{E\zeta_k}) \tag{5}$$

$$\boldsymbol{r}^{N_k} = \boldsymbol{r}'/||\boldsymbol{r}'|| \tag{6}$$

$$\sigma_r^{N_k} = \max\{\alpha \cdot ||\boldsymbol{\sigma}^{N_k}||, \sigma_r^{E\zeta_k} \cdot \exp\{\beta_r(|s_r^{N_k}| - \hat{\chi}_1)\}\} \tag{7}$$

3. The offsprings are evaluated.

4. The best individual found so far and the $\mu - 1$ best of the λ offsprings are selected to form the new parent population.

5. If the termination criterion is not fulfilled go to step 2.

An explanation of the various parameters and symbols is given in Table 1. Each individual encodes one set of camera model parameters which are used to calculate the IPM. We have seven real-valued parameters for each camera, so the individuals encode points $\boldsymbol{x} \in \mathbb{R}^{14}$.

Two different terms contribute to the mutation of the object variables, see Eq. (1). Firstly, each objective variable is mutated independently. The n individual step-sizes are stored in a vector $\boldsymbol{\sigma}$. Each individual stores information about the history of the realizations of the random vector $\boldsymbol{z}$ in a n-dimensional vector $\boldsymbol{s}$ needed for a kind of momentum effect, see Eq. (2). Correlated mutations are realized by an additional line mutation along an adaptive preferred direction $\boldsymbol{r}$. The line mutation has another step-size σ_r and stores its history in the parameter s_r (see Eq. (4)) and the normalized (see Eq. (6)) n-dimensional vector $\boldsymbol{r}$ (see Eq. (5)). If an actual step is larger (smaller) than its expectation, the corresponding step-size is increased (decreased), see Eqs. (3) and (7).

Altogether, the employed self-adaptation mechanism uses $3n+2$ strategy parameters. Not any arbitrary normal distribution with zero mean can be produced by this mutation scheme (if $n > 2$), but in practice this algorithm has proven to be a good compromise between adapting too many parameters (which can describe any covariance matrix but may take too long to adapt) and a too strong restriction of the possible mutation distributions.

Instead of initializing $\boldsymbol{s}$ and s_r with zero(s) (Hansen, Ostermeier, and Gawelczyk, 1995) we suggest to replace Eqs. (2) and (4) in the *first* iteration by

$$\boldsymbol{s}^{N_k} = \boldsymbol{z}^k \qquad \text{and} \qquad s_r^{N_k} = z_r^k \quad , \tag{8}$$

respectively. So in every generation s_r ($\boldsymbol{s}^{N_k}$) can be viewed as a realization of a normally distributed random variable (vector) with variance one — which is the underlying assumption for using $\hat{\chi}_1, \hat{\chi}_n$ and the choice of c_u. This modification is just of theoretical interest, in practice its effect can be neglected, because of the exponential decay of its influence.

4 Experiments

4.1 Setup

The experiments shown were done using the camera setup of the autonomous service robot ARNOLD (Bergener et al., 1997).

The internal camera parameters were determined using an implementation of the calibration algorithm described by Tsai (1987). Knowing the internal parameters we could mechanically adjust the cameras on the head with simple calibration patterns making the nodal lines and image rows parallel for a vergence angle of zero. This gives a good estimation of the three external camera parameters. Since we map the image of one camera to the perspective of the other, we have to optimize the parameters of both cameras. We have chosen the seven most important parameters for optimization, these are the radial lense distortion, the focal length, the image center and the three external rotation parameters of the cameras, pan, tilt and yaw. The initial values of the parameter set is calculated by the calibration scheme described above.

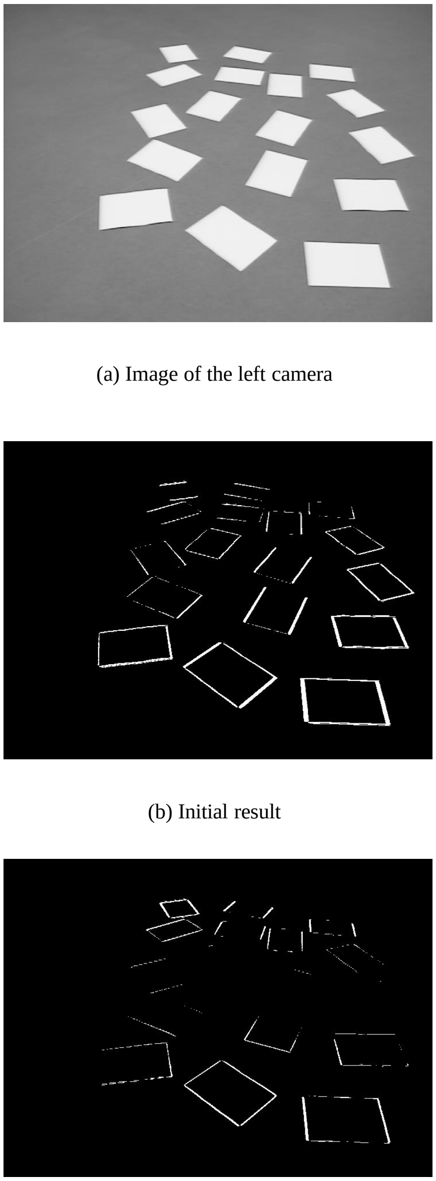
To generate an error measure needed by the EA, we put sheets of paper on the floor to get a structured horizontal plane $z = 0$ in the world coordinate system. After calculating the mapping, the number of pixels exceeding the threshold should be close to zero, since the assumption of an obstacle free plane is fulfilled. We compute the number of pixels exceeding the threshold for five different pictures. The sum of these values serves as the error measure which equals the negative fitness. We would like to stress that the error cannot vanish, because the pinhole-camera model can only approximate real cameras up to a certain extend. Furthermore, only the seven most important parameters for each camera are optimized. This was done because there is a trade-off between computational effort and accuracy.

4.2 Results

We performed 10 independent runs of the EA described in the previous section. We used a parent population size of $\mu = 10$ and the number of offsprings in each generation was set to $\lambda = 150$. The μ/λ ratio was chosen after monitoring the probability of fitness improvement (Schwefel, 1995) which constantly dropped below 0.075. The population size is comparatively large for this kind of ES. Smaller values for μ seemed to decrease the performance, but the corresponding trials were not comprehensive.

The results of the evolutionary parameter optimization have to be compared with the results achieved by an expert.

After calibration the initial mean error was 9869.8 per image. The adjustment by hand took several hours and reduced the error to 5217.8. It has to be stressed that it needs a lot of experience and exact knowledge of the mapping, i.e. the influence of the parameters, to achieve such good results. The average error of our ten trials of the $(10, 100)$-ES was 5122.6 and the best run achieved a final error value of 4422 after 212 generations, see Figure 3.



(a) Image of the left camera

(b) Initial result

(c) Result after optimization

Figure 2: Result of the evolutionary optimization of the parameters. The white pixels in Figs. (b) and (c) indicate mismatches.

Figure 2 shows one test image of the left camera (a), the initial result (b) and the result after optimization (c).

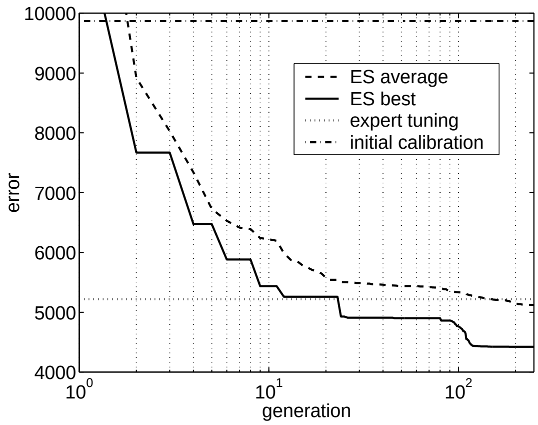


Figure 3: Results for 250 generations of the $(10, 150)$-ES, based on 10 trials

5 Conclusion

Evolutionary optimization has proven to be a suitable means to optimize the parameters of a camera model used for visual obstacle detection. It can be successfully employed to automate the setting of the parameters for the IPM. The results shown here were even on average superior to the settings suggested by an expert. The enhancement of the best parameter setting was more than 50% in comparison to the initial parameter estimation and about 20% compared to the tuning by the expert.

The IPM is used to control the mobile robot ARNOLD in an unknown and dynamically changing environment (Bruckhoff and Dahm, 1998). After optimization the obstacle detection leads to better results so that the robot can move faster and more reliable.

We are confident that the results of the mapping can be further enhanced by optimizing all camera model parameters. Furthermore, we also believe that there is room for further refinements of the EA we used in this study.

Acknowledgments

We would like to thank Martin Kreutz for providing the EALib software package and Bernhard Sendhoff for proofreading the manuscript.

References

Bäck, T. and H.-P. Schwefel (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation 1*(1), 1–23.

Bergener, T. and C. Bruckhoff (1999). Compensation of non-linear distortions in inverse-perspective mappings. Technical Report IRINI99-04, Institut für Neuroinformatik, Lehrstuhl für Theoretische Biologie, Ruhr-Universität Bochum.

Bergener, T., C. Bruckhoff, P. Dahm, H. Janßen, F. Joublin, and R. Menzner (1997). Arnold: An anthropomorphic autonomous robot for human environments. In *SOAVE'97, Selbstorganisation von adaptivem Verhalten*, pp. 25–34.

Bergener, T., C. Bruckhoff, P. Dahm, H. Janßen, F. Joublin, R. Menzner, A. Steinhage, and W. von Seelen (1999). Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks*. to appear.

Bruckhoff, C. and P. Dahm (1998, October, 13-17). Neural fields for local path planning. In *Proceedings of the International Conference on Intelligent RObotic Systems (IROS 98)*, pp. 1431–1436.

Hansen, N., A. Ostermeier, and A. Gawelczyk (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. J. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 57–64.

Mallot, H. A., H. H. Bülthoff, J. J. Little, and S. Bohrer (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics 64*, 177–185.

Ostermeier, A., A. Gawelczyk, and N. Hansen (1995). A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation 2*(4), 369–380.

Rechenberg, I. (1994). *Evolutionsstrategie '94*. Werkstatt Bionik und Evolutionstechnik. Stuttgart: Frommann-Holzboog.

Rudolph, G. (1997). *Convergence Properties of Evolutionary Algorithms*. Hamburg: Kovač.

Schwefel, H. (1995). *Evolution and Optimum Seeking*. New York: John Wiley & sons.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off- the-shelf tv cameras and lenses. *IEEE Journal Robotics and Automation*, 323–344.

n	14	dimension of the problem
μ	10	number of parents
λ	150	number of offsprings
$\boldsymbol{x}$		object variable vector, $(x_1, \ldots, x_n)' \in \mathbb{R}^n$
$\boldsymbol{\sigma}$		vector of individual step-sizes, $(\sigma_1, \ldots, \sigma_n)' \in \mathbb{R}^n$
$\boldsymbol{s}$		weighted sum of the realized random vectors $\boldsymbol{z}$, $\boldsymbol{s} \in \mathbb{R}^n$
σ_r		step-size for line mutation
s_r		weighted sum of the realized random numbers z_r with a lower bound of zero
$\boldsymbol{r}$		direction of the line mutation, $\boldsymbol{r} \in \mathbb{R}^n$
$\boldsymbol{z}, z_r$		$\boldsymbol{z} = (z_1, \ldots, z_n)'$, $z_1, \ldots, z_n, z_r$ are independent random numbers drawn according to a normal distribution with zero mean and variance one
N_k		index of the k-th offspring
E_{ζ_k}		index of the ζ_k-th parent, $\zeta_k = 1, \ldots, \mu$ with equal probability
c_u	$\sqrt{(2-c)/c}$	s_r and $\boldsymbol{s}$ are realizations of sums of normally distributed random variables and vectors, respectively, and can therefore be viewed as realizations of single normally distributed random variables (vectors); the factor c_u normalizes the variance of these random variables (vectors) to one
$\hat{\chi}_1$	$\sqrt{2/\pi}$	the expectation of the one-dimensional χ-distribution, i.e. the expectation of the absolute value of a normally distributed random variable with zero mean and variance one
$\hat{\chi}_n$	$\sqrt{n}(1 - \frac{1}{4n} + \frac{1}{1+21n^2})$	approximation of the expectation of the n-dimensional χ-distribution
c	$\sqrt{1/n}$	determines the accumulation time of the individual steps
c_r	$3/n$	determines the accumulation time of the direction adaptation
α	$1/3$	factor that controls the minimum value of σ_r; there has to be a lower bound in order to ensure that the line mutation keeps relevant
β	$2/n$	damping factor of the "global" step-size adaptation
β_{ind}	$1/(4n)$	damping factor of the individual step-size adaptation
β_r	$\sqrt{1/(4n)}$	damping factor of the direction step-size adaptation

Table 1: Symbols and parameters of the derandomized ES, for $\alpha, \beta, \beta_r, \beta_{\text{ind}}, c$ and c_r the default parameters (found by simulations) given by Hansen et al. (1995) are used