Model-less Visual Servoing Using Modified Simplex Optimization

K. Miura, K. Hashimoto, H. Inooka
Graduate School of Information Sciences
Tohoku University
Aramaki aza aoba01,
Aoba-ku, Sendai, 980-8579, Japan

J. A. Gangloff, M. F. de Mathelin
LSIIT UMR CNRS 7005
Louis Pasteur Strasbourg University
Bd. S. Blant,
67412 Illkirch Cedex, France.

Abstract

In this paper, we present a robot positioning task with respect to a static target by visual servoing. The vision system is uncalibrated and the kinematic model of the robot may be totally unknown. The displacements of the robot in joint level are generated in real time in order to minimize an objective function. The objective function includes the quadratic error between the current and the desired target images. A *simplex* method is used to minimize the objective function, and a Newton-like method is also used near the convergence. We successfully validate this method with simulations under the graphic library OpenGL.

1 introduction

Most of the previous works on visual servoing assume that the kinematic model of the robot and the camera intrinsic parameters are known. Most of these methods could work with weak calibration, but they would fail if the robot and the vision system were fully unknown.

Uncalibrated and model-less visual servoing has been addressed by Hosoda *et al.* [1], and Jägersand [2]. They use an on-line estimation of the Jacobian between the joint velocities of the robot and the feature velocities in the image plane. Their approach assumes the on-line identification of a large number of parameters. In the presence of noise, this type of approach may lead to a badly estimated Jacobian matrix if the motions of the robot do not guarantee identifiability of the parameters.

In previous research, we proposed a novel approach for uncalibrated and model-less visual servoing using a modified *simplex* iterative search method is proposed. We successfully demonstrated the algorithm with a industrial manipulator[4]. However, it usually requires a lot of iteration to complete the positioning task accurately.

In this paper, a Newton-like optimization algorithm is adopted for quicker convergence. The simulations are carried out in the case where the kinematic model of the robot is unknown, assuming that the joint limits are known. The intrinsic parameters of the camera are also unknown. The improvement of the convergence speed is discussed with the simulation results.

2 Simplex method and its modification for visual servoing

2.1 Conventional simplex method by Nelder and Mead

Simplex method is an unconstrained optimization technique. Note that it is different from the linear programming technique method also called "simplex". This method was originally proposed by Spendley, Hext, and Himsworth [7] and was developed later by Nelder and Mead [5].

This section deals with the methods for solving the minimization problem:

$$\text{Find } \boldsymbol{X} = \{x_1, x_2, \ldots, x_n\} \text{ which minimize } F(\boldsymbol{X}) \quad (1)$$

The geometric figure whose vertices are defined by a set of $n+1$ points in an n-dimensional space is called a simplex. For example, in two dimensions, the simplex is a triangle, and in three dimensions, it is a tetrahedron.

The basic idea in the simplex method is to compare the value of the objective function at the $n+1$ vertices of a simplex and move the simplex gradually toward the optimum point during the iterative process, known as reflection, contraction, and expansion.

If $\boldsymbol{X}_h$ is the vertex corresponding to the highest value of the objective function among the vertices of a simplex, we can expect the point $\boldsymbol{X}_r$ obtained by reflecting the point $\boldsymbol{X}_h$ in the opposite face to have a smaller value. Mathematically, the reflected point $\boldsymbol{X}_r$ is given by;

$$\boldsymbol{X}_r = \alpha(\boldsymbol{X}_o - \boldsymbol{X}_h) + \boldsymbol{X}_o \tag{2}$$

where $\boldsymbol{X}_o$ is the centroid of all the points $\boldsymbol{X}_i$ except $i = h$, and α is the reflection coefficient ($\alpha > 0$).

In the case $f(\boldsymbol{X}_r)$ gives the smallest cost between all the vertices, one can generally expect to see the function value decrease further by expanding $\boldsymbol{X}_r$ to $\boldsymbol{X}_e$;

$$\boldsymbol{X}_e = \gamma(\boldsymbol{X}_r - \boldsymbol{X}_o) + \boldsymbol{X}_o \tag{3}$$

where γ is called the expansion coefficient ($\gamma > 1$).

If the reflection process gives a worst value, $\boldsymbol{X}_r$ is contracted to $\boldsymbol{X}_c$;

$$\boldsymbol{X}_c = \beta(\boldsymbol{X}_h - \boldsymbol{X}_o) + \boldsymbol{X}_o \tag{4}$$

where β is called the contraction coefficient ($0 \leq \beta \leq 1$).

If the contracted point still has worst value, the contraction process will be a failure, and in this case we apply a reduction process, *i.e.* all $\boldsymbol{X}_i$ are replaced by;

$$\text{new}\boldsymbol{X}_i = \frac{1}{2}(\boldsymbol{X}_i + \boldsymbol{X}_l) \quad i = 1, 2, \ldots, n+1 \tag{5}$$

with $\boldsymbol{X}_l$, the vertex where the objective function is minimum. Then, we restart the reflection process.

The method is assumed to reach convergence whenever some stopping criteria have been met, e.g.:

$$\sqrt{\sum_{i=1}^{n+1} \frac{\{F(\boldsymbol{X}_i) - \bar{F}\}^2}{n+1}} \leq \varepsilon_1, \ \textit{with} \ \ \bar{F} = \sum_{i=1}^{n+1} \frac{F(\boldsymbol{X}_i)}{n+1} \tag{6}$$

$$\text{or} \qquad F(\boldsymbol{X}_l) \leq \varepsilon_2 \tag{7}$$

2.2 Modified simplex method for real-time visual servoing

We slightly modify the Nelder and Mead simplex method for the particular case of visual servoing with a robot. The main idea is to use a simplex like optimization algorithm to move the robot from an initial position to a goal position, so that the robot is performing the optimization. Since the vision system acquires images continuously and assuming that joint angles are also measured, the cost function can be computed along the trajectory of the robot while it is moving from a vertex to its reflection point. Therefore, the optimum could be selected along that vertex. In real-time, in other words, this is equivalent to modifying the Nelder and Mead simplex method by adding a line search procedure during the reflection, expansion, or reduction process. With this technique, the contraction process is omitted and faster convergence is expected.

3 Newton-like optimization method

Assuming that m features are detected in the image, let define $f(\boldsymbol{X})$ as the m-dimensional vector of the feature errors, where $\boldsymbol{X}$ is the vector of size n defining the position of the robot (e.g., $\boldsymbol{X}$ are the joint angles). Hence, the feature error vector $f(\boldsymbol{X}) = 0$ if $\boldsymbol{X} = \boldsymbol{X}^*$, the desired position of the robot. Therefore, the positioning task using visual servoing is achieved by minimizing the following objective function:

$$F(\boldsymbol{X}) = \frac{1}{2} f(\boldsymbol{X})^\top f(\boldsymbol{X}) \tag{8}$$

$$\text{with} \quad f(\boldsymbol{X}) = \begin{pmatrix} f_1(\boldsymbol{X}) \\ \vdots \\ f_m(\boldsymbol{X}) \end{pmatrix} \tag{9}$$

Indeed, the feature error value $f(\boldsymbol{X}^*) = 0$ minimizes $F(\boldsymbol{X})$.

A Newton-like algorithm is given by:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \alpha_k (J_k J_k^\top)^{-1} J_k f(\boldsymbol{X}_k) \tag{10}$$

where, $0 < \alpha_k$ is the step size in the descent direction, and $J(\boldsymbol{X})$ is the $n \times m$ Jacobian matrix of the feature errors:

$$J_{ij}(\boldsymbol{X}) = \frac{\partial f_j(\boldsymbol{X})}{\partial x_i} \tag{11}$$

and $J_k = J(\boldsymbol{X}_k)$.

If the Jacobian matrix is unknown, it must be estimated on-line. Piepmeier *et al.* [6] presented a moving target tracking task based on the quasi-Newton optimization method. The Jacobian of the objective function is estimated on-line with a Broyden's update formula (equivalent to a LMS algorithm). This approach is adaptive, but cannot guarantee the stability of the visual servoing scheme in presence of large errors in the image or if the motions of the robot do not guarantee identifiability of the parameters.

It should be pointed out that the Newton optimization algorithm has local properties of convergence, i.e., it will converge toward the nearest local minimum of $F(\boldsymbol{X})$

4 Simulation

4.1 Experimental procedure

The model of robot manipulator is a 6DOF industrial manipulator, and the target object is a cylindrical white cup in Fig. 1. The task consists in bringing the end-effector fitted with a camera (eye-in-hand configuration) at the vertical of the object. To compare several optimization runs, we use the same starting positions.

Three procedures are compared with respect to the number of iterations and the accuracy of the convergence:

Scheme1: A positioning task is carried out only with the simplex optimization.

Scheme2: During simplex iterative search, on-line Jacobian estimation using a Broyden's update formula is carried out using the information at the simplex vertices. Once the simplex optimization arrives near the minimum, the process switches to the Newton-like optimization with Eq. 10.

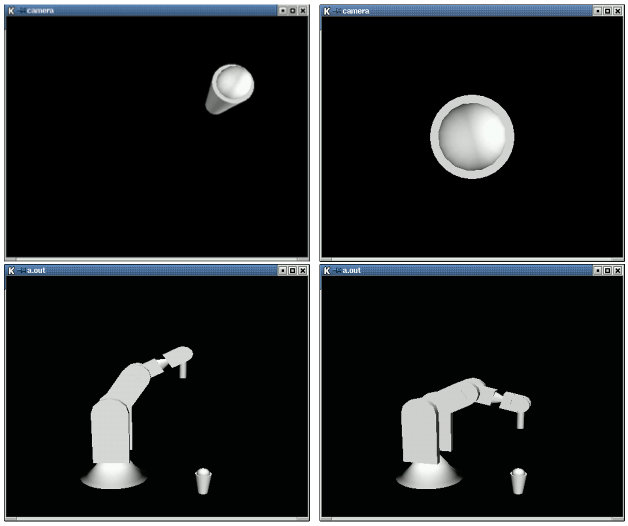
Scheme3: The process switches to the Newton-like optimization without on-line Jacobian estimation during the simplex optimization phase. The Jacobian matrix is estimated with small movement around the current position.

Furthermore, to accelerate the convergence, a hybrid scheme combining an optimization algorithm with an image-based partial visual servoing loop is proposed. The idea is to bring the target object to the center of the image, during the optimization process. Thus the target will not be lost during the iterative search, which is a well known problem in visual servoing (cf. E. Malis [3]). The manipulator has six joints, and the image-based centering scheme is assigned to two joints near the end-effector. Three joints are controlled by the simplex and Newton-like optimization, and a joint is fixed.

4.2 Objective function

The objective functions are usually selected as sum of square of feature errors. A possible objective function could have the following features:

- The distance between the end-effector and the object is given by its size in the image.
- The angle between the object axis (cylinder axis) and the optical axis of the camera is given by the form factor of its image (form factor = 1 when this angle is zero). This angle can be decomposed in two elementary angles (*e.g.* roll and pitch).



(a)start position (b)final position

Figure 1: Start and goal position of the robot and their images taken at the end-effector

We also use the sum-of-square-difference (SSD) of each pixel intensity between the current and the reference image of the object as an error. Note that SSD does not need feature extraction.

For this task, we propose the following objective/cost function:

$$F(\boldsymbol{X}) = \frac{1}{2} f(\boldsymbol{X})^\top f(\boldsymbol{X}) \qquad (12)$$

where,

$$\begin{aligned} f_1(\boldsymbol{X}) &= W_1 \left(\frac{s}{S} - 1 \right) \\ f_2(\boldsymbol{X}) &= W_2 \left(\frac{l_{long}}{l_{short}} - 1 \right) \\ f_3(\boldsymbol{X}) &= W_3 \sum_{i,j} \left(P_{cur}(i,j) - P_{ref}(i,j) \right)^2 \end{aligned}$$

where s and S are the actual and the desirable size of the object in the image, l_{long} is the longest distance from the center of the object to its contour, and l_{short} is the shortest one. Therefore $\frac{l_{long}}{l_{short}}$ is the form factor of the object image. P_{cur} and P_{ref} are pixel intensities of the current and the reference image, respectively. Finally, W_1, W_2, and W_3 are the respective weights of feature errors in the cost function.

5 Results and Discussion

Simulations were carried out comparing several schemes. All schemes were executed with modified simplex optimization until rough convergence. Termination tolerance in Eq.6 and Eq.7 were decided experimentally.

Scheme1 converged, but it took more iteration near the minimum. The modified simplex roughly converged with 16 iterations, and the process took 27 iterations near the minimum. The vertices of the simplex often fell into a small local minimum, and the simplex restarted the process again.

Scheme 2 did not converged. The estimated Jacobian was not close enough to the true Jacobian. Though this type of estimation method guarantees the accuracy in a small area, the estimation used the image data at the vertices of the simplex in a large area. A solution is to have a large forgetting factor λ, however, it may lead to a badly estimated Jacobian matrix in the presence of noise.

Scheme 3 converges quicker than simplex near the minimum with 36 total iterations. The trajectory was also very simple, as shown in Fig. 2. However, this scheme was not always robust when the objective function had large noise at the initial position of the Newton-like optimization process.

All these experiments make it clear that the optimization with only the simplex method takes many iterations near the minimum. Newton-like method with on-line Jacobian estimation converged faster, since the Newton optimization algorithm has better local properties of convergence. However, it has a risk of a badly estimated Jacobian matrix in the presence of noise.

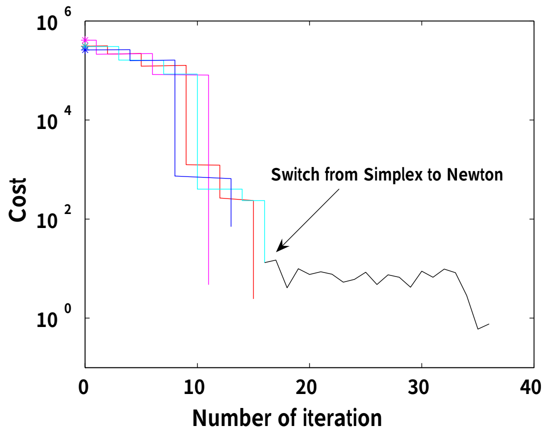


Figure 2: Trajectory of the optimization process

6 Conclusion

In this paper, we used the modified simplex optimization techniques for a positioning task by visual servoing. This method does not need a model of the robot and does not require the estimation of Jacobian matrices. Thus, a robot never goes in the wrong direction due to the bad estimation. Moreover, the objective function does not need to be differentiable.

Since the simplex method took more iterations near the minimum, the Newton-like method with on-line Jacobian matrix estimation was also executed in order to have quicker convergence. We successfully demonstrated the proposed scheme with simulations.

Improvements are needed for stable convergence. Jacobian matrix estimation was not always correct because of the large motions of the robot between the vertices of the simplex that do not guarantee sufficient excitation for the identification of the parameters.

For the future works, it is important to find the objective function without noise, which may cause a badly estimated Jacobian matrix.

References

[1] K. Hosoda, and M. Asada, Versatile Visual Servoing without Knowledge of True Jacobian, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Munchen, September 1994.

[2] M. Jägersand, O. Fuentes, and R. Nelson, Experimental Evaluation of Uncalibrated Visual Servoing for Precision Manipulation, Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation, New Mexico, April 1997.

[3] E. Malis, F. Chaumette, and S. Boudet, *2 1/2 D Visual Servoing*, IEEE Transactions on Robotics and Automation, Vol. 15, No. 2, pp. 238–250, 1999.

[4] K. Miura, J. A. Gangloff, and M. F. de Mathelin, *Robust and Uncalibrated Visual Servoing without Jacobian using a Simplex Method*, Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, September 2002.

[5] J. A. Nelder, and R. Mead, *A simplex method for function minimization*, Computer Journal, Vol. 7, No. 4, pp. 308–313, 1965.

[6] J. A. Piepmeier, G. V. MacMurray, and H. Lipkin, Uncalibrated Dynamic Visual Servoing, IEEE Transactions on Robotics and Automation, Vol. 20, No. 1, pp.143–147, 2004.

[7] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimization and evolutionary operation*, Technometrics, Vol. 4, pp. 441–461, 1962.