IMPROVING THE PERFORMANCE OF MODEL-BASED TARGET TRACKING THROUGH AUTOMATIC SELECTION OF CONTROL POINTS

I. PAVLIDIS M. J. SULLIVAN R. SINGH
N. P. PAPANIKOLOPOULOS

Artificial Intelligence, Robotics, and Vision Laboratory
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455

ABSTRACT

In this paper, we further expand on work first presented elsewhere [9] regarding the automatic selection of control points for model-based target tracking. The shape descriptive qualities of the segmentation algorithm [9] proposed for the tracking task are tested experimentally. Comparative experiments are also presented for a model-based tracking scheme with and without the segmentation algorithm. The experiments highlight the positive features of the algorithm and verify the positive role the algorithm can play in a model-based tracker in terms of speed and quality of tracking.

KEYWORDS: curve segmentation, deformable-model-based trackers, rigid-model-based trackers, corners, key flat points.

1 INTRODUCTION

The need for target tracking arises in a number of different applications in robotics research. Characteristic examples include vision-based control of grasping and manipulation tasks [10, 13], and visual tracking of moving objects [3, 6]. Target tracking is also important in a number of other applications, like automated surveillance and traffic monitoring [12]. A spectrum of techniques has been developed for real-time visual tracking. Model-based tracking is a well-established and popular approach that involves the use of either deformable models [3, 4, 12, 13] or rigid models [5].

A necessary first step in the computation of certain models [3, 4, 5, 12, 13] is to determine a set of control points to approximate the tracked object's contour. Until recently, this was usually done by hand through a user-interface. However, the possibility of using a curve segmentation algorithm was often indicated. Picking control points manually, renders difficult the automation of the entire tracking task. In addition, since the user is picking the points randomly or at best by using some heuristic developed through his/her own experience, he/she tends to pick either too many or too few control points. On the other hand, using some classical curve segmentation algorithms [2, 7, 8] only half-automates the task since the performance of these algorithms depends upon the fine tuning of a number of parameters. Different object shapes may require different parameter settings or otherwise the segmentation algorithm will perform at times either excessive segmentation or sparse segmentation.

In [9], for the first time, a segmentation algorithm was proposed (named P & P), that filled out the existing gap in all the respects. Specifically, the proposed algorithm fully automates the selection of the control points since it does not depend on any parameters and works equally well for most kinds of shapes. Comparative experiments in [9] showed that the P & P algorithm comparatively to other curve segmentation algorithms, manages to select a small number of points that yet deliver a superior description of the original shape.

In this paper, the P & P algorithm is further analyzed and tested. It is also incorporated in a model-based tracker and its beneficial role in tracking in terms of speed and quality is verified experimentally. The organization of the paper is as follows: Section 2 refers to some model-based trackers that may benefit out of the proposed algorithm. Section 3 describes an experimental investigation of the algorithm's descriptive power. In Section 4, the performance of a model-based tracker with and without the algorithm is reported and discussed. Finally, in Section 5, the paper is summarized and conclusions are drawn.

2 MODEL-BASED TRACKERS

There are two major categories of model-based trackers: deformable-model-based trackers and rigid-model-based trackers. Some of them require the selection of control points along the contour of the target and may directly benefit from the P & P algorithm. As far as deformable-model-based trackers are concerned, Curwen *et al.* in [4] use a B-spline approximation to the original target contour. The control points of the B-spline could be appropriately placed by the P & P algorithm. The P & P algorithm is especially suitable for spline approximation of curves because it does not only locate high curvature points on the contour but also key in-between low curvature points. The latter helps in the reduction of the spline's approximating error at a small cost. Yoshimi *et al.* in [13] and Sullivan *et al.* in [11, 12] use a formulation of deformable models that involves an explicit placement of control points along the contour of the tracked object. This, and the fact that the computational cost of their methods is linear in the number of control points makes them ideal candidates for the testing of the proposed algorithm. In fact, Sullivan's implementation in [11] is the method we chose to highlight the potentially beneficial role of the P & P algorithm in model-based tracking (see Section 4). As far as rigid-model-based trackers are concerned, the algorithm can also be proved useful in automatically building a succinct and accurate model of a 2D object from its initial image.

3 EXPERIMENTAL INVESTIGATION OF THE ALGORITHM'S DESCRIPTIVE POWER

The P & P algorithm locates points of high curvature (corners) using a method similar to that in [2]. It also locates key in-between low curvature points (key flat points) by employing a procedure conjugate to that for locating corners. The P & P algorithm is described in detail in [9]. Here, only an interesting experimental investigation of the algorithm's shape approximating power is presented.

In order to get an indication of the goodness of the algorithmic selection of control points in terms of the accuracy of shape description, the following experiment was devised. Let a contour C of an arbitrary shape consist of N points ($C = (\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N)$). Let the P & P algorithm select for the contour C a set S of m control points ($S = (\mathbf{P}_{s1}, \mathbf{P}_{s2}, \ldots, \mathbf{P}_{sm})$). Let also a set $\mathcal{T}$ of m control points ($\mathcal{T} = (\mathbf{P}_{t1}, \mathbf{P}_{t2}, \ldots, \mathbf{P}_{tm})$) to be chosen in a way so that an error norm is driven to minimum (optimal polygonal fit). The norm chosen for the purposes of the particular experiment was the Euclidean distance error of the polygonal fit represented by the point set. The set $\mathcal{T}$ was determined after an exhaustive search of all the $\binom{N}{m}$ combinations for the contour C. It is interesting to compare the set of control points given by the P & P algorithm with the optimal polygonal fit point set for a variety of shapes (see Figs. 1-4).

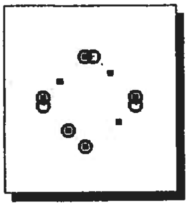

Figure 1: A square contour.

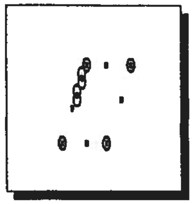

Figure 2: A parallelogram contour.

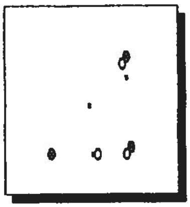

Figure 3: A triangular contour.

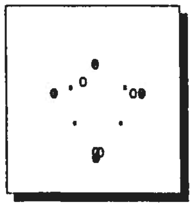

Figure 4: An irregular contour.

The small circles in the above figures represent the points of the optimal polygonal fit set while the points given by the P & P algorithm are represented by small squares. In all the shapes, the prominent corners are included in both the optimal polygonal fit set and the set of the P & P algorithm. Discrepancies arise only for the key flat points of the algorithm. The equivalent points of the optimal polygonal fit are mostly clustered in noisy areas of the shape. In contrast, the key flat points of the algorithm are uniformly distributed between the prominent corner points. This behavior is highly desirable, since the algorithm has not been designed specifically for a polygonal fit but for a more generic fit that may be even a spline fit. In fact, some model-based techniques use the control points for polygonal fits [11, 12, 13] and some others for spline fits [4]. The algorithm loses very little in terms of polygonal fit accuracy by placing the key flat points in a distributed instead of a clustered manner. For example, in the irregular contour case of Fig. 4, the error of the optimal fit is 0.8189 pixels while the error of the P & P fit is 2.1701 pixels. The error of an arbitrary polygonal fit for this shape could run as high as 42.8378 pixels. The small compromise the algorithm concedes in the polygonal fit case pays off in the spline fit case where a clustered distribution like the one favored by the optimal polygonal fit would give very poor results.

4 EXPERIMENTAL TRACKING RESULTS

Preliminary results of experiments incorporating the P & P algorithm for automatic control point selection in a model-based tracking scheme [11] suggest that this approach holds great promise. The P & P algorithm extends the previous system [11] in two important ways. It automates the selection of both the number and location of control points. In the previous implementation, the number of control points was preselected by the operator and their location was manually determined at run-time. By automating these tasks, the P & P algorithm makes the system more general and more independent of its operator. The system has been implemented on the Minnesota Robotic Visual Tracker ([1], see also Fig. 5).

Experiments were conducted in which a target was presented on a 27 inch monitor located one meter from the end-effector mounted camera. The target, a 7.3 cm tall square or triangle, moved around a rectangular path of 100 cm at approximately 8 cm/sec. The position commands sent to the robotic arm were collected and are graphically illustrated in Figs. 6 - 8. Previous results [11] (see Fig. 7) were compared to results using the P & P Algorithm (see Fig. 8).

The previous system used a *predetermined* number of control points irrespective of the target's shape. These points were *manually* placed near the object contour in a highly regular configuration. The generic constraints used by the tracking algorithm created a bias toward equidistant points and equal angles between edges. The new system uses the P & P algorithm to automatically select control points. Because the P & P algorithm does not choose equally spaced points, the constraints used during tracking were modified to reward configurations with angles close to the initial angles and distances close to the initial distances.

The model-based tracking scheme described in [11] worked well only when a small number of control points was selected and the points described the contour well. Since that system encouraged equidistance between control points and equal angles between edges, it performed best when the contour of the object being tracked could be approximated by an equilateral polygon (a highly regular shape) with as many vertices as the model had control points. For less regular shapes or control point configurations, performance degraded. For example, the system in [11] lost track of the square target after just one revolution when an eight-point model was used (see Fig. 7). The old system was not tested with the (non-equilateral) triangular target, since this target is not a highly regular shape.

The system using the P & P algorithm for automatic point selection performed substantially better. Ten trials were measured. In the first five, the arm tracked the moving square. In the second five, the triangular target was tracked. Results from the first trial with each target are presented in Figs. 6 and 8 respectively. The control point selection algorithm invariably selected ten points for the square and six points for the triangle that appropriately described the shapes. The tracker maintained tracking of the objects for several revolutions. In this experiment, the P & P tracker exhibited its ability to maintain tracking at fairly high speeds of different target shapes (square, triangle).

Figure 5: Experimental setup.

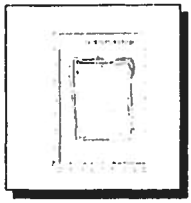

Figure 6: Tracking of a triangular target with the P & P algorithm.

Figure 7: Tracking of a square target without the P & P algorithm. The target was lost after one revolution.

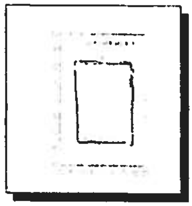

Figure 8: Tracking of a square target with the P & P algorithm.

5 SUMMARY

In this paper, further experimental investigation of the P & P algorithm, first appeared in [9], was reported. The P & P algorithm was designed to automate the selection of control points for certain model-based trackers. The algorithm was designed to perform satisfactorily for polygonal as well as spline fits, since both abound in model-based trackers. In the present work, the algorithm's output was compared with the corresponding point set that gave the optimal polygonal fit for a variety of shapes. The error of the algorithm's polygonal fit was very close to the error of the corresponding optimal fit. In particular, the corner points reported by the algorithm coincided with the corner points of the optimal set for every shape tested. Discrepancies between the algorithm's point set and the optimal polygonal fit set arose for some of the key flat points reported by the algorithm. These discrepancies cost a small approximation error to the polygonal fitness of the algorithm that is anticipated to pay off in the case of spline fits. Similar experiments for spline fits are under way and will be reported in the future.

The algorithm was also incorporated in a model-based tracker [11] and preliminary comparative experiments between the old and new systems highlight the beneficial role the P & P algorithm can play in model-based tracking. Further experiments with a greater variety of shapes and under a greater variety of conditions are under way and will be reported in the future.

ACKNOWLEDGEMENTS

This research was supported by the Department of Energy (Sandia Labs) through Contracts $\#AC-3752D$ and $\#AL-3021$, the National Science Foundation through Grants $\#IRI-9410003$ and $\#IRI-9502245$, the Minnesota McKnight Land-Grant Professorship Program, the Center for Transportation Studies at the University of Minnesota, and the Department of Computer Science at the University of Minnesota. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

References

[1] S. A. Brandt, C. E. Smith, and N. P. Papanikolopoulos. "The Minnesota Robotic Visual Tracker: a Flexible Testbed for Vision-Guided Robotic Research." In *Proceedings the 1994 IEEE International Conference on Systems, Man and, Cybernetics*, pp. 1363-1368, 1994.

[2] J. J. Brault and R. Plamondon. "Segmenting Handwritten Signatures at Their Perceptually Important Points." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, pp. 953-957, 1993.

[3] P. A. Couvignou, N. P. Papanikolopoulos, and P. K. Khosla. "Hand-Eye Robotic Visual Servoing Around Moving Objects Using Active Deformable Models." In *Proceedings of the 1992 IEEE International Conference on Intelligent Robots and Systems*, pp. 1855-1862, 1992.

[4] R. Curwen and A. Blake. "Dynamic Contours: Real-Time Active Splines." In *Active Vision*, pp. 39-57, MIT Press, 1992.

[5] C. Harris. "Tracking with Rigid Models." In *Active Vision*, pp. 59-73, MIT Press, 1992.

[6] N. P. Papanikolopoulos and P. K. Khosla. "Feature-Based Robotic Visual Tracking of 3-D Translational Motion." In *Proceedings of the 1991 IEEE Conference on Decision and Control*, pp. 1877-1882, 1991.

[7] T. Pavlidis. *Structural Pattern Recognition*, pp. 168-184. Springer-Verlag, 1977.

[8] T. Pavlidis and S. T. Horowitz. "Segmentation of Plane Curves." *IEEE Transactions on Computers*, Vol. 23, pp. 860-870, 1974.

[9] I. Pavlidis and N. P. Papanikolopoulos. "Improving the Performance of Model-Based Target Tracking Through Automatic Selection of Control Points." To appear, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 22-28, 1996.

[10] C. E. Smith and N. P. Papanikolopoulos. "Grasping of Static and Moving Objects Using a Vision-Based Control Approach." In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 329-334, 1995.

[11] M. J. Sullivan and N. P. Papanikolopoulos. "Using Active Deformable Models to Track Deformable Objects in Robotic Visual Servoing Experiments." To appear, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 22-28, 1996.

[12] M. J. Sullivan, C. A. Richards, C. E. Smith, O. Masoud, and N. P. Papanikolopoulos. "Pedestrian Tracking from a Stationary Camera Using Active Deformable Models." In *Proceedings of the Intelligent Vehicles '95 Symposium*, pp. 90-95, 1995.

[13] B. H. Yoshimi and P. K. Allen. "Visual Control of Grasping and Manipulation Tasks." In *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 575-582, 1994.

EXPERIMENTAL VALIDATION OF THE EXTERNAL CONTROL STRUCTURE FOR THE HYBRID COOPERATION OF TWO PUMA 560 ROBOTS

V. PERDEREAU - M. DROUIN
PARC Université Paris VI boite 164
4 Place Jussieu - 75252 Paris Cedex 05 - FRANCE

P. DAUCHEZ
LIRMM - UMR 9928 Université Montpellier II/CNRS
161 rue Ada - 34392 Montpellier Cedex 5 - FRANCE

ABSTRACT

For the successful coordination of two arms handling a common object in unstructured or ill-known environments, V. Perdereau and M. Drouin [1] proposed to implement at each arm level an efficient hybrid position/force controller where the force control loop is closed around the position loop. For now, the efficiency of this new hierarchical solution was only proven by simulation results. It was however suggested that real-time applications with industrial robots could be viable. This paper is devoted to reporting the validation of this method we have achieved in collaboration with P. Dauchez at the LIRMM in Montpellier on an experimental setup built around two PUMA 560 robots.

1. INTRODUCTION

When two robots operate in a complex environment and work on a same object interactively to achieve complicated and dexterous tasks, the object motion may be constrained in some directions due to interaction with external environment. It is then necessary to control the constraint force, i.e., the external force, in addition to the motion of the object and to the relative position/orientation of both end-effectors (or the reaction forces, i.e., the internal forces, between the arms). The control objective is therefore to realize the desired position and force profiles in a constrained coordinate frame located at the grasped object; controllers are supposed to explicitly use the forces sensed at the robot end-effectors.

One fundamental advantage of the master/slave approach [2] [3] is that the two arms are controlled independently allowing a distributed computer architecture and an easier implementation. However, both controllers do not share the same force and position errors, the force controller must react fast enough to changes in position to avoid dropping or damaging the object.