This appears in: Proceedings of the International Conference on Robotics and Automation, 2006

A heterogeneous distributed visual servoing system for real-time robotic assembly applications

Youngrock Yoon, Johnny Park and Avinash C. Kak
Robot Vision Lab, Purdue University, West Lafayette, Indiana 47907
{*yoony,jpark,kak*}@*ecn.purdue.edu*

***Abstract*— We describe a multi-loop robotic control architecture that for the fine-motion control required for object tracking uses two different loops that depend on entirely different approaches for visual servoing. One tracking loop uses fast blob analysis of the camera images for servo control. The other tracking loop, which is more robust but slower, uses model-based line matchings for a more robust calculation of instantaneous pose. A control arbitrator gives a higher priority to the faster blob-analysis based control loop. But should any disturbances in the system cause the loss of servo control with respect to the input received from the blob-analysis based control loop, the control arbitrator automatically switches to the other more robust control loop. As a result, the system shows high level of fault tolerance under non-cooperative conditions, such as severe occlusion and lighting condition changes.**

I. Introduction

Visual servoing is being increasingly used for industrial applications such as bin-picking of complicated industrial objects [1] and, more recently, real-time assembly on moving assembly lines [2]. For such applications, especially when assembly must be carried out on moving assembly lines, robustness and reliability are of critical importance [3].

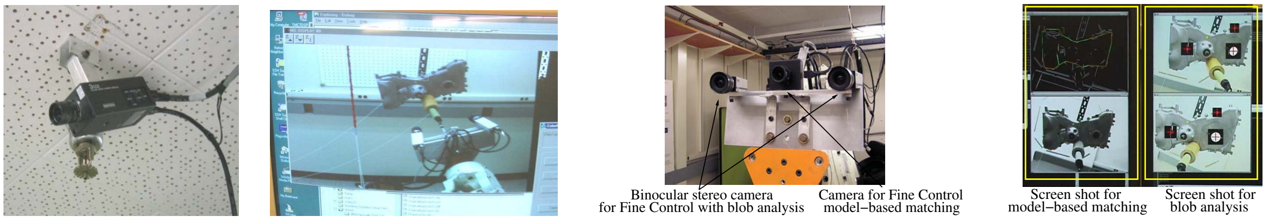
It is now generally recognized that, from the standpoint of reliability and robustness, sensor-based control architectures for robots must use multiple feedback loops [2], [4]–[6]. In particular, we want to draw the attention of the reader to the control architecture described by DeSouza and Kak [2] that uses a distributed and hierarchical control architecture which allows multiple feedback loops to operate in parallel at any level of a control hierarchy. An arbitrator determines as to which specific control input should be used to control the end-effector at any given time instant. This control architecture shows a high level of fault tolerance in the presence of various sources of noise in the visual inputs used by the object tracking modules. When the arbitrator chooses only one of the different available control inputs, another advantage of such a distributed control architecture is that the performance of the overall system is not subject to the sum of delays of all the modules in the system, but it depends on which module is used for controlling the robot at a specific time instant.

To achieve even higher levels of robustness and reliability, we now extend the work of DeSouza and Kak by 1) incorporating another vision-based fine-motion control loop that uses high-level model-based matching for visual servoing; and 2) by using a control arbitration strategy that selects between blob-analysis based fine-motion control and the new model-based fine-motion control. Using more rigorous model-to-scene matching strategy, the new fine-motion control loop is obviously more robust, but also at the same time, somewhat slower than the blob-analysis based fine-motion controller. Therefore, as long as the output of the blob-analysis based controller is within servo range, the control arbitrator uses that output for controlling the end-effector. However, if due to occlusions or other effects (such as instantaneous glare or illumination change) there should be loss of control with respect to the blob-analysis based control loop, the arbitrator automatically switches to the full model-based control loop. The fact that the two modules for fine-motion control are completely different in how they function vis-a-vis the sensory inputs should explain the use of the word 'heterogeneous' in the title of this paper. [1]

The control architecture used in the current work therefore uses two control loops, operating in parallel, for the fine-motion control of the robot end-effector. One of these is the same as in [12]; this one uses blob-analysis in a pair of stereo images for real-time estimation of the pose of the object to be tracked. The new additional control loop for fine-motion control, as already mentioned, uses full-blown model-based matching of an image with the 3D model of the object. The vision processes for model-based matching in the new control loop are the same as reviewed in [13]. The algorithms we use for model-based matching involve assessment of scene measurement and the pose estimation uncertainties, and feature correspondence search to yield pose calculations that possess a higher level of robustness to occlusion and lighting condition.

The overall system described in this paper performs real-time assembly – putting a peg into a hole that is in a visually

[1] There is a parallel to our contribution in the related area of object tracking in cluttered backgrounds in video imagery. Tracking algorithms that employ simple strategies with a priori knowledge of simple features on the target produce outputs at high speed that can be reasonably accurate under certain assumptions [7]–[9]. However, if the assumptions break down because of, say, changes in lighting condition or because the target object becomes occluded, the tracker can easily fail to produce reliable output. On the other hand, tracking algorithms with elaborate model of the target – for example, CAD model of an object – and more complicated pose estimation strategies, such as the ones in [10], [11], should be able to produce reliable tracking results even under non-cooperative conditions. However, these algorithms often involve large amounts of computation, hence they do not show the same level of responsiveness as the ones using simple strategies. With a heterogeneous control architecture, such as the one we present in the work reported here, where algorithms with both strategies run concurrently and independently of each other, a system can adaptively select outputs from modules that give the most reliable and accurate outputs at each instant of time.



(a) Coarse control camera (b) Coarse control screen shot (c) Fine control cameras (d) Fine control screen shot

Fig. 1. Coarse and Fine Control.

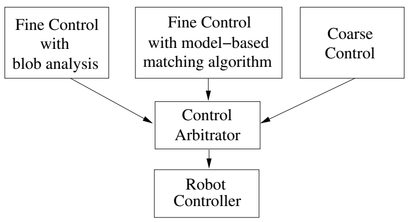


Fig. 2. Control architecture of the visual servoing system.

complex industrial part in motion. The industrial object with the hole hangs from a linear slide. The motion of the industrial object can be easily disturbed by human intervention by pulling at the strings attached to the object.

In the next section, we start by presenting the overall control architecture, followed by a description of each module in the architecture in Section II-B through II-C.

II. Control Architecture

Fig. 2 shows the overall control architecture of the system. There are three control modules in the system: one coarse control and two fine controls. These control modules operate concurrently and independently of one another. Each module reports its control input to be applied to the end-effector to the control arbitrator. The control arbitrator then adaptively selects the most reliable and accurate control input at each instant of time.

A. Coarse Control

The coarse control module, as the name suggests, is responsible for providing an approximate location of the target object that can be used to guide the robot end-effector to the vicinity of the object. In our system, imagery from a camera mounted on the ceiling is used for coarse control. This camera can observe the entire robot workspace. Along the same lines as described in [2], the coarse control module applies a simple threshold to segment out the target object from the input scene captured by the ceiling mounted camera. Fig. 1-(a) shows the ceiling mounted camera used by this module. Fig. 1-(b) shows an example of what the camera sees as the object moves into its field of view.

B. Fine Control

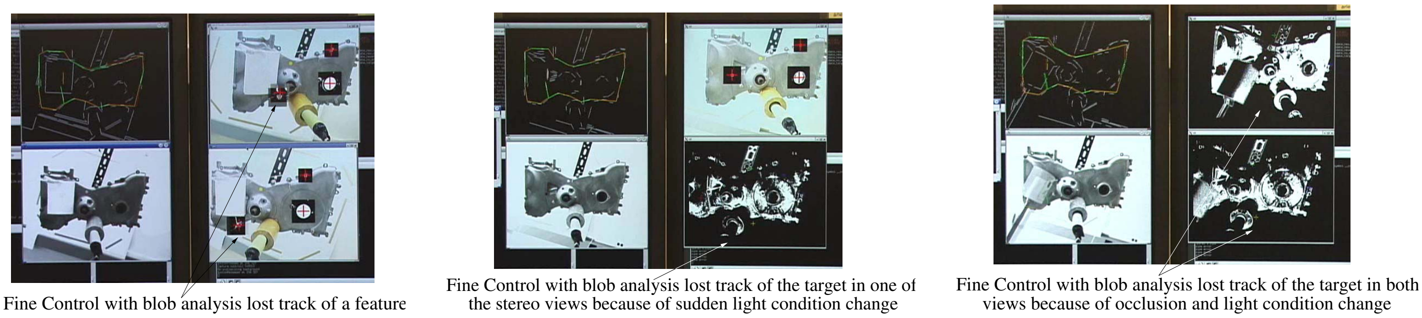
As mentioned in the introduction, we have significantly enhanced the robustness of the control architecture of [2] by incorporating another fine-motion control module that uses model-based matching. As a result, the fine control now consists of two modules working in parallel, one blob-analysis based and the other model-based. We will now briefly describe the two.

1) Fine Control with Blob Analysis of Stereo Images: As shown in the right side of Fig. 1-(d), this module tracks three circular features in two stereo images taken with cameras mounted on the end-effector. The module sets up six search windows (three for each stereo image) on the targeted features, uses simple thresholding to segment out the prominent blobs in each search window, rank-orders the blobs according to certain criteria described in [12], and, finally, selects the blob with the highest scores in each of the six search windows. Stereo triangulation formulas are then used to calculate the 3D coordinates of the centers of the object features that give rise to these blobs in the two images. These 3D coordinates are used to estimate the pose of the target object. This module runs at 60 frames per second – 30 frames for each image of the stereo pair – and estimates the pose of the target with an average error of $0.7mm$ and 0.3° [12].

2) Fine Control with Model-based Matching: This module uses a model-based pose estimation scheme for tracking the target. A wire-frame model of the target that consists of straight-line segments is used for pose estimation (on the left side of Fig. 1-(d)). It first projects the model on to the input scene with respect to the initial pose that is given by the coarse control module. Then, it sequentially matches the straight-line features in the model to the edges in the scene for an updated calculation of the pose of the target. For robust pose estimation, it uses a backtracking scheme for feature correspondence search. This module runs at 7 frames per second with an average error of $8.5mm$ and 1.7°. [13].

C. Control Arbitrator

The control arbitrator adaptively selects the most reliable and accurate control input available at each instant of time. Roughly speaking, the control arbitrator gives the highest priority to the control input given by the fine control module with blob analysis, next the fine control module with model-based matching, and the lowest priority to the the coarse control module. For example, when a control input by the fine control module with blob analysis is available, the control arbitrator always selects this input. The control arbitrator selects the input by the fine control module with model-based matching only when the module with blob analysis is in a fail mode (we will shortly describe what we mean by the fail mode). Finally, the control input by the coarse control module



(a) Occlusion

(b) Light condition change

(c) Occlusion and light condition change

Fig. 3. Sample screen shots of the fine control modules with various disturbances.

is used only when both of the fine control modules are in a fail mode. The rationale for the above rules is that the tracking with blob analysis provides more accurate pose estimation of the object with a higher frame rate than that with model-based matching.

A control loop enters into a fail mode if any of the two following conditions occur. First, when the visual tracking algorithm loses track of the object; in this case the module simply sends a failure message to the control arbitrator. Second, when the module is unable to send messages to the control arbitrator for some unexpected reason; in this case the arbitrator determines that the module is in a fail mode if the elapsed time since the last received message is greater than a predefined threshold.

III. Experimental results

In order to show the enhanced robustness of our system, we manually created disturbances while the system performs real-time assembly – putting a peg into a hole that is in a visually complex industrial part in motion. Our system was tested under three non-cooperative conditions: (1) a part of the target object was occluded; (2) a spot light facing the object was repeatedly turned on and off; and (3) combination of the conditions (1) and (2).[2]

Under all three conditions, the fine control module with blob analysis often lost track of the object while the module with model-based matching continued to track the object. It was evident that the control arbitrator was able to adaptively select the most reliable control input at times of disturbance since the motion of the robot end-effector was smooth during the entire assembly process.

Fig. 3 shows sample screen shots of the two fine control modules in all three experiments.

IV. Conclusion and Future Work

In this paper, we presented a heterogeneous distributed visual servoing system for a real-time robotic assembly. The proposed system achieves a high level of robustness under non-cooperative conditions by using two concurrent and independent fine control loops that use different tracking strategies. Experiments showed that our system is able to successfully perform a real-time assembly on a moving industrial object under severe occlusion and lighting condition changes.

The control architecture of the current system strictly maintains the independence of each control loop. That is, the control arbitrator takes only one control input from a single module. While this approach has advantages in terms of robustness and scalability, it neglects the possibility of improving accuracy by fusing multiple input from other control modules. Thus, our future work includes designing a new control architecture that takes full advantage of the independency of control loops, and, at the same time, is able to fuse multiple available inputs when appropriate in order to improve accuracy.

[2]A movie that shows the robustness of the system under these non-cooperative conditions is posted on the following URL: http://rvl.www.ecn.purdue.edu/RVL/movies/LineTracking/ICRA06.wmv

References

[1] A. Kosaka and A. C. Kak, "Stereo vision for industrial applications," in *Handbook of Industrial Robotics, Second Edition*, 1992, pp. 269–294.

[2] G. N. DeSouza and A. C. Kak, "A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics," *IEEE Trans. SMC, Part B*, vol. 34, no. 5, pp. 1988–2002, 2004.

[3] D. Kragic and H. I. Christensen, "Cue integration for visual servoing," *IEEE Trans. RA*, vol. 17, no. 1, pp. 18–27, Feb. 2001.

[4] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. RA*, vol. 2, no. 1, pp. 14–23, Mar. 1986.

[5] R. C. Arkin, "Motor schema-based mobile robot navigaion," *Intl. J. Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989.

[6] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack, "Experiences with an architecture for intelligent, reactive agents," *JETAI*, vol. 9, pp. 237–256, 1997.

[7] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. RA*, vol. 9, no. 2, pp. 152–165, Apr. 1993.

[8] N. P. Papanikolopoulous and P. K. Khosla, "Adaptive robotic visual tracking:theory and experiments," *IEEE Trans. AC*, vol. 38, pp. 429–445, Mar. 1993.

[9] G. Hager, G. Grunwald, and K. Toyama, "Feature-based visual servoing and its application to telerobotics," in *Intelligent robotic systems*, V. Graefe, Ed. Elsevier, 1995.

[10] H. Kollnig and H. Nagel, "3d pose estimation by directly matching polyhedral models to gray value gradients," *IJCV*, vol. 23, no. 3, pp. 282–302, 1997.

[11] G. Hirzinger, M. Fischer, B. Brunner, R. Koeppe, M. Otter, M. Gerbenstein, and I. Schafer, "Advanced in robotics: The dlr experience," *Intl. J. Robotics Research*, vol. 18, pp. 1064–1087, Nov. 1999.

[12] Y. Yoon, G. N. DeSouza, and A. C. Kak, "Real-time tracking and pose estimation for industrial objects using geometric features," in *Proc. ICRA '03*, Sept. 2003, pp. 3473–3478.

[13] Y. Yoon, J. B. Park, A. Kosaka, and A. C. Kak, "A new approach to the use of edge extremities for model-based object tracking," in *Proc. ICRA '05*, Apr. 2005, pp. 1883–1889.