Hyper Frame Vision: A Real-Time Vision System for 6-DOF Object Localization

Yasushi Sumi† Yutaka Ishiyama‡ Fumiaki Tomita†
†Intelligent Systems Institute, AIST
Tsukuba, 305-8568 Japan
{y.sumi, f.tomita}@aist.go.jp
‡Stanley Electric Co., Ltd. R&D
Yokohama, 225-0014 Japan
y.ishiyama@yrd.stanley.co.jp

Abstract

A new system for robot vision is proposed that integrates a 3-D object recognition task and a 3-D object tracking task, enabling real-time 6-DOF localization of a known continuously moving object. A computational time-lag between the two tasks is absorbed by a large amount of frame memory. For 3-D sensing, calibrated trinocular stereo cameras are used to employ stereo-vision-based object recognition and tracking methods that allow objects of any shape to be dealt with and to provide robust performance in an environment with partial occlusions and cluttered backgrounds. The effectiveness of the system is demonstrated by experimental results.

1. Introduction

Model-based 3-D object localization is a fundamental issue in robot vision and there have been two approaches to deal with it. One is *3-D object recognition* that identifies a known object in a cluttered scene and determines its 6-DOF position accurately [3, 1]. This approach is generally used for stationary objects, because it requires time-consuming computations for matching an object model and a scene.

The other is *3-D object tracking* that measures 3-D displacements of an object between two frames in time-sequential input images [10, 8, 2, 7, 5, 9, 12]. The displacement must usually be calculated faster than the video-rate (30 fps) to achieve real-time processing. This approach is generally used in a known environment, because the initial position of the object must be given beforehand in order to start the tracking task. For example, the vehicle tracking system by Haag & Nagel [4] predicts the initial positions of vehicles using background knowledge in traffic scenes such as lanes.

Both approaches have been investigated independently, although their objectives are basically the same. It is necessary to integrate them into one practical vision system that can locate a moving object.

We propose a vision system, which we call *Hyper Frame Vision*, that allows the real-time 6-DOF localization of an object in unknown rigid motion. We consider this to be widely applicable to various intelligent robot systems, for example, bin-picking, visual servoing, and self-localization. In our Hyper Frame Vision system, a recognition task and a tracking task are combined through *a spatio-temporal frame buffer* which is a large amount of frame memory to absorb the computational time-lag between the two tasks.

For 3-D sensing, we use calibrated trinocular stereo cameras to employ a stereo-vision-based object recognition method [11] and an object tracking method [6]. Each method is implemented as a vision module that runs on a parallel computing environment.

This paper provides an overview of the vision modules and describes the configurations and the processing steps of Hyper Frame Vision. Experimental results are then presented to demonstrate the effectiveness of the system.

2. Vision Module

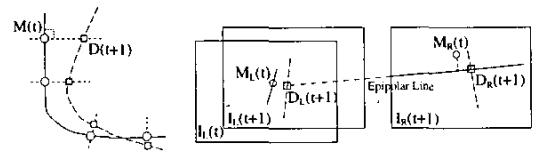
2.1. Recognition Module

The recognition module identifies a known object and determines its 6-DOF position using a stereo-vision-based recognition method [11]. This method recovers 3-D boundaries from a set of stereo images, selects candidates for the object position according to the alignment of local boundary features, and verifies and improves each candidate by iterative closest point registration. The method enables us to deal with objects of any shape and can provide robust performance in environments with partial occlusions and cluttered backgrounds. The computing time of a recognition task took at least one second in our experiments.

2.2. Tracking Module

The tracking module measures the 6-DOF displacement of an object between every two frames in real-time using a

1051-4651/02 $17.00 © 2002 IEEE



(a) Frame-to-frame correspondence. (b) Stereo correspondence

Figure 1. Correspondence Searches.

stereo-vision-based tracking method [6]. This method establishes point correspondences between the two frames to calculate the displacement of the object. First, as shown in Fig. 1(a), frame-to-frame correspondence is established using the model of the object. *A model point* $M_L(t)$, which is a point representing edges on the object, is projected onto one of the stereo images $I_L(t+1)$ and an edge peak $D_L(t+1)$ is traced in the neighborhood of $M_L(t)$. Then, as shown in Fig. 1(b), stereo correspondence is established using epipolar constraints. A model point $M_R(t)$ is projected on the corresponding image $I_R(t+1)$ and an edge peak $D_R(t+1)$ is traced on the epipolar line of $D_L(t+1)$ in the neighborhood of $M_R(t)$. Finally, the 3-D position of a data point D is recovered from $\{D_L, D_R\}$. After all possible data points are recovered, the optimum 3-D displacement of the object is calculated by the least squares method. The above processing steps must be iterated to converge the object's position at time $t+1$.

This tracking module can also deal with objects of any shape and is robust in environments with partial occlusions. The computing time of a tracking task took several milliseconds per frame in our experiments. That is, the computational time-lag between the recognition and the tracking modules is more than 100-times.

3. Hyper Frame Vision

3.1. System Configuration

The Hyper Frame Vision system consists of calibrated trinocular stereo cameras for 3-D sensing and a spatio-temporal frame buffer that absorbs the time-lag between the vision modules. Although the spatio-temporal frame buffer requires hundreds of megabytes of memory to store hundreds of frames of trinocular stereo images, recent inexpensive electronic devices make it easy to build such massive systems.

3.2. Six-DOF Localization of a Moving Object

Figure 2 shows the basic processing of our system. The horizontal axis represents time. The top of the figure denotes images stored to the frame memory, the middle band denotes the actions of the vision modules, and the bottom shows how the modes shift.

After the first image is stored to the 0th frame $F(0)$ at time t_f, the recognition module starts its recognition task to recognize the target object, where t_f is the frame period (typically $t_f = 1/30$). When the recognition task is completed at time $t_f + t_{rec}$, the recognition module sends a recognition result $\boldsymbol{T}(0)$ to the tracking module, where $\boldsymbol{T}(n)$ is a 4×4 transformation matrix that expresses the object position in the nth frame and t_{rec} is the processing time for the recognition task. We call the period until $t_f + t_{rec}$ *the SEARCH mode*. In this mode, the system invokes the recognition module to recognize the object because the object's position is unknown.

Upon receiving the result $\boldsymbol{T}(0)$, the tracking module starts its tracking task to calculate the object's position $\boldsymbol{T}(i \mid i = 1, 2, \ldots)$ in sequence using $\boldsymbol{T}(0)$ as the initial position. At the beginning of the tracking task, the object's previous positions must be calculated in previous frames. We call this *the APPROACH mode*. In the APPROACH mode, the following relationship holds at time t:

$$n_{trac} < n_{in}, \tag{1}$$

$$n_{trac} = \frac{t - (t_f + t_{rec})}{t_{trac}}, \quad n_{in} = \frac{t}{t_f}, \tag{2}$$

where n_{trac} is the number of a previous frame being tracked, n_{in} is the number of the currently stored frame, and t_{trac} is the mean processing time per frame in the tracking task. The right sides of the equations (2) are raised to integral values. If $t_{trac} < t_f$, then n_{trac} gradually approaches n_{in} over time.

At time

$$t_{catchup} = \frac{t_f + t_{rec}}{t_f - t_{trac}} t_f, \tag{3}$$

n_{trac} becomes equal to n_{in}. Thereafter, as shown in Fig. 2, the tracking task is suspended each time the latest stereo images are stored. We call this *the CATCH mode*. In the CATCH mode, the system is tracking the object in real-time.

3.3. Resumption of Tracking Processes

Figure 3 shows the resumption processings when the target object is lost while being tracked. The tracking module can detect that the object is lost using an evaluation value calculated at each frame. The evaluation value of a frame i is defined as

$$S_{trac}(i) = \frac{n_{trac}(i)}{n_{vis}(i)}, \tag{4}$$

where n_{vis} is the number of model points in the correspondence search described in Section 2.2 and n_{trac} is the number of data points corresponding to the model points. In the

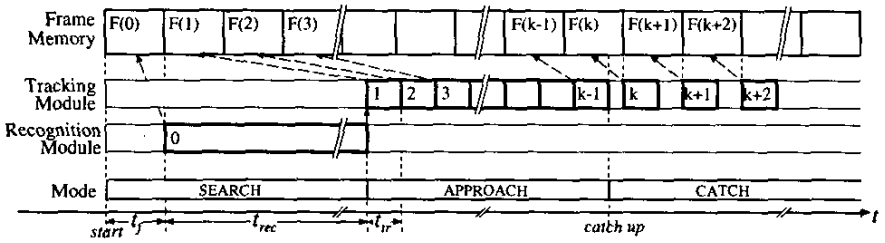


Figure 2. Basic Processing of Hyper Frame Vision.

APPROACH or CATCH modes, if

$$S_{trac}(i) < \theta_S, \tag{5}$$

the system shifts to the SEARCH mode on the supposition that the object may be lost, where θ_S is a threshold.

There are two tracking resumption strategies. One invokes the recognition module whenever the object is lost. As shown in Fig. 3(a), when the system shifts to the SEARCH mode triggered by the evaluation value $S_{trac}(i)$, the recognition module is invoked and starts a new recognition task for that frame $F(i)$. After the recognition task completes, the evaluation values of the recognition and tracking modules are compared. If

$$S_{trac}(j) < wS_{rec}(i) \tag{6}$$

is satisfied, the system then shifts to the APPROACH mode again, where w is a weight coefficient and S_{rec} is an evaluation value that is calculated for the recognition result in the same manner that S_{trac} is calculated. If (6) is not satisfied, as in the case of Fig. 3(a), the recognition module starts a new recognition task for the current frame $F(j)$ because the recognition for $F(i)$ failed.

Another tracking resumption strategy is to keep the recognition tasks running as a background process, as shown in Fig. 3(b). Condition (6) is tested whenever a recognition task is completed. If a good recognition result is obtained, the system is forced to shift to the APPROACH mode. This strategy allows the resumption time to be shorter than strategy (a), but the trade-off is an increase in computational costs.

In a resumption process, the tracking module continues the tracking task persistently, because the tracking task may not have lost the target object, even if condition (6) is not satisfied. This could occur when the evaluation value is temporarily lower because of partial occlusions or other troubles. If the troubles are overcome and the evaluation value S_{trac} exceeds the threshold again, the tracking module will continue its tracking task.

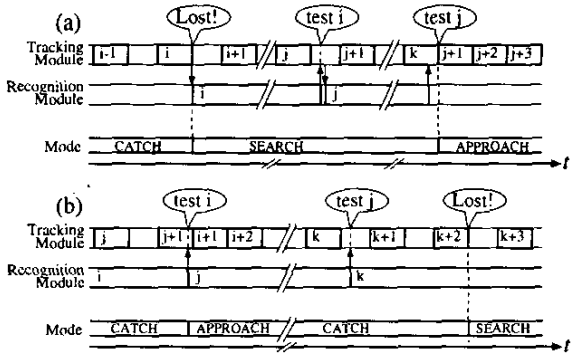


Figure 3. Resumption after Losing Object.

4. Experiments

The Hyper Frame Vision can be implemented with two types of hardware configurations: a PC with large main memory and a DMA frame grabber or a PC with a frame grabber with large onboard frame memory. Our experimental results presented here were obtained using the latter system on a Dual PentiumIII 866MHz PC with a Microtechnica MTPCI-CD frame grabber board which had 786MB of frame memory. Three accurately calibrated CCD cameras with 25 mm lenses were used for acquiring time-sequential trinocular stereo images at 640 × 480 pixel resolution. The Hyper Frame Vision program was developed in C on Linux/SMP using the POSIX thread.

To demonstrate the effectiveness of our Hyper Frame Vision, we carried out experiments locating a moving object using a rotating stage as illustrated in Fig. 4. The two loci in the figure show the experimental results. At each locus, the X-coordinate of a certain point on the object is plotted. The photographs in the figure show the object's positions at the times indicated by the dotted lines in the loci. When the object is being correctly tracked, the locus ought to be a sine curve with a period of approximately 20 seconds be-

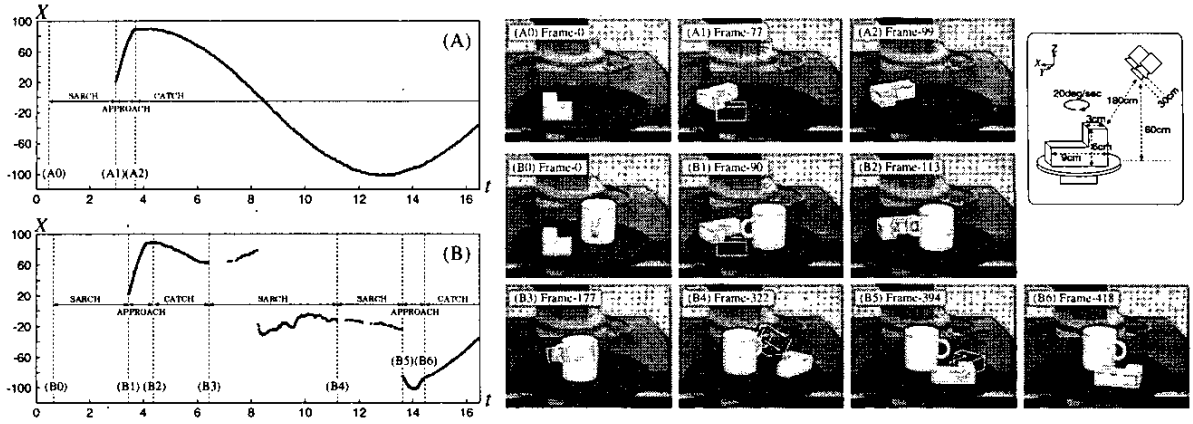


Figure 4. Experimental results: (A) Object without occlusion, (B) Object occluded temporarily.

cause the stage was rotated at 18 deg/sec parallel with the X-Y plane of the world coordinate system.

In experiment (A), the object was not occluded and the result shows that the moving object was recognized at A1 and tracked correctly after A2. In experiment (B), almost the entire object was intentionally occluded to examine the resumption strategy (a) described in Section 3.3. The threshold θ_S in condition (5) was set at 0.5. The experimental result shows that although the object was lost at B3, the tracking process was resumed through the same process of Fig. 3(a).

In our experiments, the mean computational time took approximately 2 to 4 seconds for the recognition task and approximately 7.4 milli-seconds per frame for the tracking task. The localization accuracy was approximately 1 mm.

5. Conclusions

This paper has presented Hyper Frame Vision, a real-time vision system for 6-DOF object localization. Experiments demonstrate that the system can recognize and track a continuously moving object and can resume the tracking process if the object is lost while tracking. Numerous experiments have been also carried out to determine whether the system can deal with objects of various other shapes, such as industrial parts, which were moved arbitrarily by hand and satisfactory results were obtained.

The Hyper Frame Vision is being improved for practical applications. We are now developing a visual servoing system for hand-eye robots and a self-localization system for autonomous mobile robots.

References

[1] F. Arman and J.K. Aggarwal. Model-based object recognition in dense-range images — A review. *ACM Computing Surveys*, 25(1):5–43, 1993.

[2] D. Gennery. Visual tracking of known three-dimensional objects. *Int. J. of Computer Vision*, 7(3):243–270, 1992.

[3] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, 1990.

[4] M. Haag and H.-H. Nagel. Combination of edge element and optical flow estimates for 3D-model-based vehicle tracking in traffic image sequences. *Int. J. of Computer Vision*, 35(3):295–319, 1999.

[5] S. Hiura, A. Yamaguchi, K. Sato, and S. Inokuchi. Real-time object tracking by rotating range sensor. In *Proc. ICPR96*, I, pp. 825–829, 1996.

[6] Y. Ishiyama, Y. Sumi, and F. Tomita. 3-D motion tracking of 3-D objects using stereo vision. *J. of Robotics Society of Japan*, 18(2):213–220, 2000. (in Japanese).

[7] H. Kollnig and H.-H. Nagel. 3D pose estimation by fitting image gradients directly to polyhedral models. In *Proc. ICCV95*, pp. 569–574, 1995.

[8] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Int. J. of Computer Vision*, 8(2):113–122, 1992.

[9] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proc. ICCV99*, I, pp. 262–268, 1999.

[10] R.S. Stephens. Real-time 3D object tracking. *Image and Vision Computing*, 8(1):91–96, 1990.

[11] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita. 3D object recognition in cluttered environments using segment-based stereo vision. *Int. J. of Computer Vision*, 46(1):5–23, 2002.

[12] M. Tonko and H.-H. Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *Int. J. of Computer Vision*, 37(1):99–118, 2000.