Proceedings of the 2004 IEEE
International Conference on Control Applications
Taipei, Taiwan, September 2-4, 2004

An Air Combat Simulator in the Virtual Reality with the Visual Tracking System and Force-feedback Components

Cheng-Ming Huang[1], Su-Chiun Wang[1], Chih-Fu Chang[1], Chin-I Huang[1], Yu-Shan Cheng[1], Li-Chen Fu[1,2]
Department of Electrical Engineering [1]
Department of Computer Science and Information Engineering [2]
National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail: lichen@ccms.ntu.edu.tw

Abstract - *This paper presents a new multi-plane air combat simulator based on the visual servo system. The scenario is an air combat of two fighters in virtual reality. The escaping one is simulated by a Stewart platform with a force-feedback joystick, overall controlled by a human operator. The chaser is realized as a visual servo system, as if the simulated air fighter were equipped with a military light-of-sight (LOS) to lock on to the mentioned escaping air fighter. Specifically, any arbitrary-shaped object that appears on a screen reflecting the virtual space of the escaping fighter can be automatically detected and located by the visual system in real-time so that the 2-DOF camera simulating LOS can be controlled due to visual servo to keep the target object centered in the camera image. By sensing the motion of the camera platform, the autopilot of the chasing fighter tries to follow the escaping one as closely as possible with an aim of shooting it down. The overall system is experimented in a virtual reality environment to validate the underlying research work. A promising application of this setup is to evolve into an entertainment facility or a military training platform.*

Keywords: visual tracking, visual servo, virtual reality.

1 Introduction

Visual tracking has been an important topic in computer vision and robotics fields. For practical visual tracking systems, there are some basic functionalities required: real-time, automatic, and robustness in non-ideal situations, such as occlusions and cluttered environment. Surveys show that individual problems mentioned above have been solved. For example, CONDENSATION algorithm [5] is highly regarded since it solved both the real-time and the robustness problems, but not the automatic one. Up to now, to the best of our knowledge, there is no literature addressing all the problems. Therefore, an integrated algorithm is presented in this paper to overcome the overall problems mentioned above by combining template matching [10-12] and Snake-based counter matching [1-3, 13-14]. The proposed algorithm also improves the disadvantages of classical Snake [1-3]. The traditional Snake algorithm has two major disadvantages. One is about that the poor initial contour [15] will fail to converge to the desired global contour feature. The other one is about the low convergent speed, so it is seldom applied in visual servo problem.

On the other hand, virtual reality (VR) is a kind of advanced technology which facilitates scientists and engineers to simulate and to verify their works in an interactive application environment. Apparently, unmanned air vehicle (UAV) experiments pose really challenging environments for various related conduct researches [9]. Owing to this condition, before performing that real experiment, this paper proposes a multi-plane simulator which integrates the visual tracking algorithm and the VR presentation technique so that a UAV equipped with a visual tracker can be tested in lab.

This paper is organized as follows. Section 2 describes the functions of the subcomponents of the visual tracking system for tracking arbitrary-shaped moving object in the low-contrast environment, even there are some shelters. Section 3 presents the visual servo application of the UAV air combat in the virtual reality. Section 4 shows some experiment results, and Section 5 gives some conclusions.

2 Visual Tracking System

The architecture of our visual tracking system can be briefly shown by the block diagram in Fig.1. It consists of four subsystems, including motion detection, Snake-based outline extraction, hybrid tracking algorithm, and visual probability data association (VPDA) filter. The on-line image sequences are grabbed by a camera, which is mounted on a pan-tilt servo platform. In the beginning, the camera platform is stationary. When a moving target appears in the detecting screen image, the motion detector will starts the system. The Snake-based outline extraction is used to find out the target position and size in the image.

0-7803-8633-7/04/$20.00 ©2004 IEEE

We always keep the target in the center of the screen image by moving the 2-DOF camera platform. The visual servo system will correct the errors from the image information feedback of the visual tracking system. The hybrid-based tracking algorithm does the image processing to find out the target in each captured image frame. Since the target may keep on moving during the processing time between the image frames, we use the VPDA (visual probabilistic data association) filter to estimate the future position of the target. The VPDA filter also can increase the robustness and reliability of our tracking system by considering several similar image processing results of the hybrid-based tracking algorithm. Finally, the estimated target position will be transformed to adjust the pan and tilt rotation angles of the camera platform motors. If the system is lost tracking, it has homing ability to restart the system. Otherwise, it will keep on tracking in the right loop in Fig. 1.

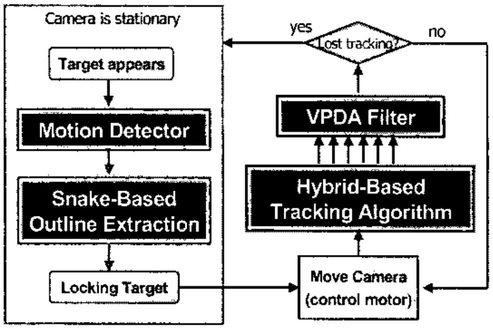


Fig.1. The architecture of visual tracking system.

2.1 Motion Detection

There are several advantages of using motion detection before tracking an object. First, it provides important clues to an automatically tracking system. If a moving object enter the camera view field, the motion detector will detect some differences between two consecutive frames and start the system. Second, we can determine a smaller searching range to reduce the computation time of finding out the position of the moving object in the image, obtained by motion detection with a stationary camera. Third, the segmentation of the moving object can be the basis for automatic initialization of Snake.

First, two consecutive image frames $I(k)$ and $I(k-1)$ are subtracted pixel by pixel, and the absolute values of the results are then binarized. If the total number of white pixels is less than the threshold, the motion detection unit continues to examine the next two consecutive frames. On the contrary, if the number exceeds the threshold, it means that there is a target to be detected. The concept of "moving edge" is included by doing a logic AND operation between the subtracted image and the edge image of the current frame. We can yield a more accurate image position of the target outline from the moving edge image. Finally, since we have known the approximate size and position of the moving target from the moving edge image, the moving edge image is submitted to generate a proper-sized initialization of Snake for outline extraction, which is detailed in next section.

2.2 Snake-Based Outline Extraction

The objective of the section is to extract the outline of the target by active contour models (also called Snakes) [1]. A modified Snake is presented here on the basis of the Greedy algorithm [2]. The proposed Snake defines a force constraint to speed up the convergence to the desired feature of the object, and provides an easy way to avoid the wrong convergence on a local noisy point. If $v_i = (x_i, y_i)$ for $i = 0,1,\dots,N$ represents the N-length discrete contour, the modified Snake energy is

$$\begin{aligned} E^*_{snake} &= \sum_{i=1}^{N} E_{snake}(v_i) \\ &= \sum_{i=1}^{N} \{\alpha_i E_{cont}(v_i) + \beta_i E_{curv}(v_i) + \gamma_i E_I(v_i) + \eta_i E_{dis}\} \end{aligned} \tag{1}$$

where E_{cont} is the continuity energy, E_{curv} is curvature energy, E_I is the image energy and E_{dis} is the constraint energy, called distance energy. The parameters α_i, β_i, γ_i and η_i are used to adjust the relative influence of the four terms. The four energy terms are defined as

$$\begin{aligned} E_{cont} &= \left| \frac{\bar{d} - |v_i - v_{i-1}|}{MAX\left\{\bar{d} - |v_i(j) - v_{i-1}|\right\}} \right|, \\ E_{curv} &= \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{MAX\left\{|v_{i-1} - 2v_i(j) + v_{i+1}|^2\right\}}, \\ E_I &= \frac{\nabla I_{\max} - \nabla I(v_i)}{\nabla I_{\max} - \nabla I_{\min}}, \\ E_{dis} &= \frac{D_{\max} - D(v_i)}{D_{\max} - D_{\min}}, \end{aligned} \tag{2}$$

where $\bar{d}$ is the average distance between contour points v_i, and $v_i(j)$ represents the eight neighbors of a point v_i for $j = 1,\dots,8$. Moreover, $\nabla I(v_i)$ denotes the image intensity of the edge image at current position, $D(v_i)$ denotes the absolute value of the distance between the current position and the center of the object, and $\nabla I_{\max}$ ($D_{\max}$) and $\nabla I_{\min}$ ($D_{\min}$) denote the maximum and minimum image intensity (distance from the center of the object) in the neighborhood, respectively. The moving direction of the Snake can be decided by adding or subtracting the distance energy E_{dis}.

The concept of Greedy algorithm [2] is to take minimized E^*_{snake} as the sum of the minimized values $E_{snake}(v_i)$ for each individual point in the contour, that is

$$\min E^*_{snake} \approx \sum_{i=1}^{N} \min E_{snake}(v_i)\,. \tag{3}$$

Figure 3(a) demonstrates how the iterative Greedy algorithm works. The energy function is computed for the current position v_i and each of its neighbors. Then, the location having the smallest energy value is chosen as the new position v_i'. As a result of repeating the aforementioned process point by point, all points of the contour keep moving forward to their corresponding new positions to form a new contour. Throughout the processes, the Snake completes an iteration of the deformation loop. The Snake will repeat the deformation loop iteratively until it converges to the desired feature of the object, as shown in Fig. 3(b).

From the result of motion detector, a moving object is detected. Assume the whole body of the object is completely inside a 140×140 area, which is the searching area. We choose a proper-sized ellipse, which can enclose the whole moving object inside, as an initialization of Snake. The center of ellipse is located at the center of the bounding box that contains the object inside. As shown in Fig. 4, we also use the center of the minimized rectangle as that of the searching area of the hybrid tracking algorithm.

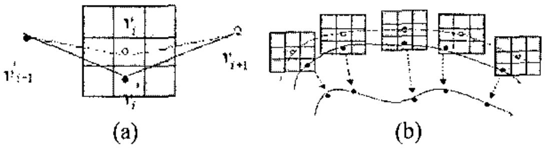


Fig. 3. (a) The demonstration of Greedy algorithm. (b) The deformable of Snake.

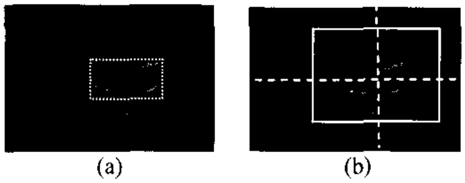


Fig. 4. In moving edge image, (a) the bounding box of the object, and (b) is the searching area and its center.

2.3 Hybrid Tracking Algorithm

Different from the traditional Snake-based tracking algorithms [3-10], Snake is only used to extract the outline of the target rather than to perform tracking in our system. Since Snake-based tracking algorithms should be under the assumption of slowly moving object, it is failed in tracking the object with large movement. This paper presents a contour matching method which makes use of the extracted contour model for tracking an arbitrary-shaped object. But the method also has a restriction against low-contrast environment, just like what the Snake-based tracking algorithms encountered. Hence, we integrate the most commonly used method in visual tracking, called template matching, with contour matching to overcome the foregoing drawback.

We use the SAD (sum of absolute difference) criterion for template matching:

$$\phi_{SAD}(s) = \sum_{m=0}^{B-1}\sum_{n=0}^{B-1}\left|I(x+m,y+n) - I_T(m,n)\right|, \tag{5}$$

where $s=(x,y)$ represents the position of the center of template model, $I(x+m,y+n)$ is the current searching image values, and $I_T(m,n)$ is the referenced template image values with size $B \times B$.The details of contour matching are described as follows. To highlight the contour of an object in an image, the edge image is used for object tracking (see Fig.5). Similar to template matching, we need to sum the total gradient values pixel by pixel in the edge image along a pre-extracted Snake contour model. After summing over a contour model, we will shift the center of contour model to the next pixel and compute the summation along the perimeter of the contour model again. After going through all the searching area, we can get the largest summation corresponding to the distribution of edge pixels which resembles the object's contour. The numerical process in each searching loop mentioned above can be summarized by the following normalized sum equation

$$\phi_g(s) = \frac{1}{N_\sigma}\sum_{i=1}^{N_\sigma}\left|g_i\right|, \tag{4}$$

where $s=(x,y)$ represents the position of the center of contour model, N_σ is the total number of all pixels along the perimeter of contour model, and g_i denotes the gradient value of each pixel along the perimeter of contour model. Hence, the best object position is considered to combine eq. (4) and eq. (5) as

$$s^* = \arg\max_{s_i \in Sr}\left\{\bar{\phi}_g(s_i) + \bar{\phi}_{SAD}(s_i)\right\}, \tag{6}$$

where

$$\overline{\phi}_g(s_i) = \frac{\phi_g(s_i) - \min\limits_{s_i\in Sr}\phi_g(s_i)}{\max\limits_{s_i\in Sr}\phi_g(s_i) - \min\limits_{s_i\in Sr}\phi_g(s_i)} \quad \text{and}$$

$$\overline{\phi}_{SAD}(s_i) = \frac{\max\limits_{s_i\in Sr}\phi_{SAD}(s_i) - \phi_{SAD}(s_i)}{\max\limits_{s_i\in Sr}\phi_{SAD}(s_i) - \min\limits_{s_i\in Sr}\phi_{SAD}(s_i)}$$

are the normalized forms, in which s_i is the center position of the contour model as well as the template, and Sr is the searching area.

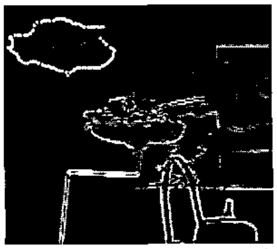


Fig. 5. Search the object by contour matching in the edge image.

2.4 VPDA Filter and Trajectory Design

Highly cluttered environment may contain many background contents that are quite similar to the target and then result in false alarms, which may cause further error accumulation in the subsequent tracking. Not only a bad template or contour will be involved, but also the trajectory detection will be misled. Furthermore, it will seriously affect the position of the searching window. Thus, it gives rise to a high possibility of lost of tracking once the best match is a false match. For this reason, VPDA is adopted to overcome the challenge in the cluttered environment. It can provide a more reliable approach to predict the next position of the target and enhance the robustness of the tracking system, even in the cluttered environment. Actually, the VPDA filter is the original probabilistic data. association (PDA) filter [7] integrated with the visual information, introduced in [8].

The concept of the PDA filter is to take all possible targets into account instead of only the best matched one that may be correspond to a part of the cluttered environment. Finally, it produces a weighted-average output from all possible candidates. The method applied for computing the weights is probabilistic. It computes the posterior association probabilities for all current possible candidate measurements and uses them to form a weighted sum of innovations for updating the target's state in a suitably modified version of the Kalman Filter.

3 Air Combat Simulator

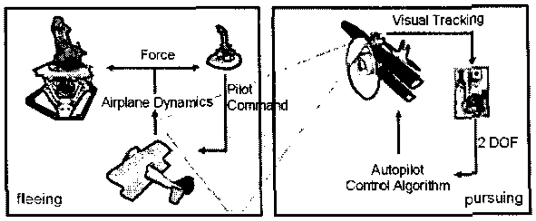


Fig. 6. The configuration of the air combat simulator.

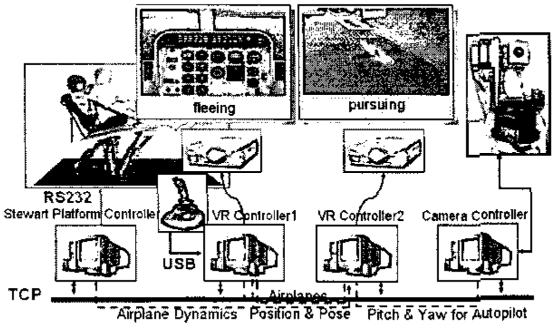


Fig. 7. The architecture of the overall system.

As has been mentioned in introduction, the hereby proposed visual servo algorithm is taken to build a virtual reality system involving UAV which allows simulation and evaluation of the control performance of some UAV equipped with a visual tracker. As shown in Fig. 6 and Fig. 7, there are two air fighters in the virtual reality environment. The challenge is that the tracking path of the UAV is dynamically changing, not a regular cruising or off-lined designed one. On the other hand, the UAV is experimented in the VR, however, our visual tracking system and the force-feedback system is constructed in the real world. There must be complex coordinate transformations and relationships between each subsystem.

The fleeing one is operated through force-feedback joystick by the pilot who sits on a steward platform. he force-feedback system utilizes the motional Stewart platform and the force-feedback joystick to express the real force feeling of the pilot with respect to the real airplane dynamics. The force-feedback joystick is 3 DOF with 2 DC motors and throttle. Our Stewart platform has the features of the high accuracy servo valve with higher control bandwidth to offer rapid response and high power to payload weight ratios. With the highly nonlinearity actuators combining with dynamic platform model, the backstepping adaptive control law.

The pursuing one is mounted with the visual tracking system as mentioned above. We want to emphasize that the visual tracking system here is to play the role as a military light-of-sight to lock on the target. Also, we can use the pan and tilt information of the camera to aid the autopilot of the pursuing air fighter. In other words, the visual tracker of the pursuer in the virtual world will be replaced by the camera mounted on a motion platform, and the VR image will be presented from the viewpoint of the pursuer so that the visual servo system can successfully fulfill its endowed function. Each subsystem communicates with one another through TCP/IP protocol.

4 Experimental Results

The performance of the Snake-based outline extraction is quite appealing and very fast even if the object appears in highly cluttered environment, as shown in Fig. 8. In Fig. 8, the dashed contour curve describes the contour matching, the small rectangle shows the template matching, and the large rectangle is the search range. Figure 9 and 10 demonstrate tracking of a moving tank model and a VR target, respectively. In the VR autopilot case, we can see that the fleeing one is chased by the pursuer, as shown in Fig. 11 (a) and (b), from the viewpoint of the pursuer. Experimental results show that our tracking system is highly robust against occlusion, low-contrast environment, and rapid motion. Figure 11(c) shows the fleeing airplane and the force-feedback components.

5 Conclusions

The contribution of this work is to establish a multi-plane simulator based on an integrated visual servo system. The overall system integrates the visual tracking, virtual reality, force feedback, network communication, and robotics. The major component is the visual servo system, which can track an arbitrary-shaped object in a highly noisy environment, even with some occlusions on the target. The visual tracking system can perform the tracking processes with 320×240 image size in nearly real-time constraint (less than 34 ms) and center the target up to 15-pixel square range tolerance. We also improve the Snake model, the visual tracking method with a hybrid tracking method, and the VPDA filter. The simulator can further be developed as an entertainment or military training platform.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: active contour models," *Int. J. Comput. Vis.*, Vol. 1, pp. 321-331, 1987.

[2] D. J. Williams and M. Shah. "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, Vol. 55, No. 1, pp. 14-26, Jan. 1992.

[3] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow", *IEEE Trans. Image Processing*, Vol. 7, No. 3, pp. 359-369, 1998.

[4] J. Denzeler and H. Niemann, "A two stage real-time object tracking system," *In Pavesi'c et al.*

[5] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for visual tracking," *Int. J. Computer Vision*, pp. 1-36, 1998.

[6] J. Denzeler and H. Niemann, "A new energy term combining Kalman filter and active contour models for object tracking," *Machine Graphics Vision*, Vol. 5(1/2), pp. 157-165, 1996.

[7] Y. B. Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, Vol. 11, pp. 451-460, 1975.

[8] D. Liu, "Real-time visual tracking in cluttered environment with a pan-tilt camera," *Master Thesis*, Dept. of E.E., National Taiwan University.

[9] A. H. Göktoğan, E. Nettleton, M. Ridley, and S. Sukkarieh, "Real time multi-UAV simulator," *Proc. IEEE ICRA*, pp. 2720-2727, 2003.

[10] R. Brunelli and T. Poggio, "Template matching: matched spatial filters and beyond," *MIT AI Memo 1549*, July 1995.

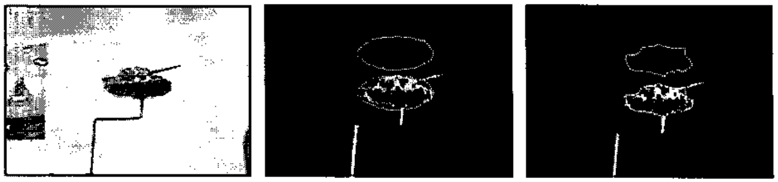
[11] M. S. Lew, N. Sube, and T. S. Huang, "Improving visual matching," *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 58-65, 2000.

[12] Y. S. Chen, Y. P. Hung, and C. S. Fuh, "A fast block matching algorithm based on the Winner-update strategy," *Proc. of the 4th Asian Conf. on Computer Vision*, Vol. 2, pp. 977-982, 2000.

[13] S. Birchfield, "An elliptical head tracker," *31st Asilomar Conference on Signal, Systems, and Computers*, November 1997.

[14] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, California, July 1998.

[15] C. Han, S. Hatsukami, J. N. Hwang, and C. Yuan, "A fast minimal path active contour model," *IEEE Trans. Image Processing*, Vol. 10, No. 6, June 2001.



(a) current image (b) initial contour (c) extracted outline

Fig. 8. It takes only 9 iterations between (b) and (c) to extract the tank's outline with 84 pixel-length of initial contour.

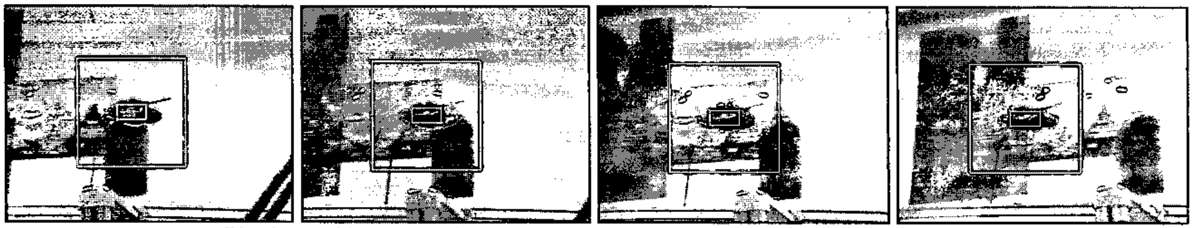


Fig. 9. Tracking real tank model in the cluttered environment with occlusions.

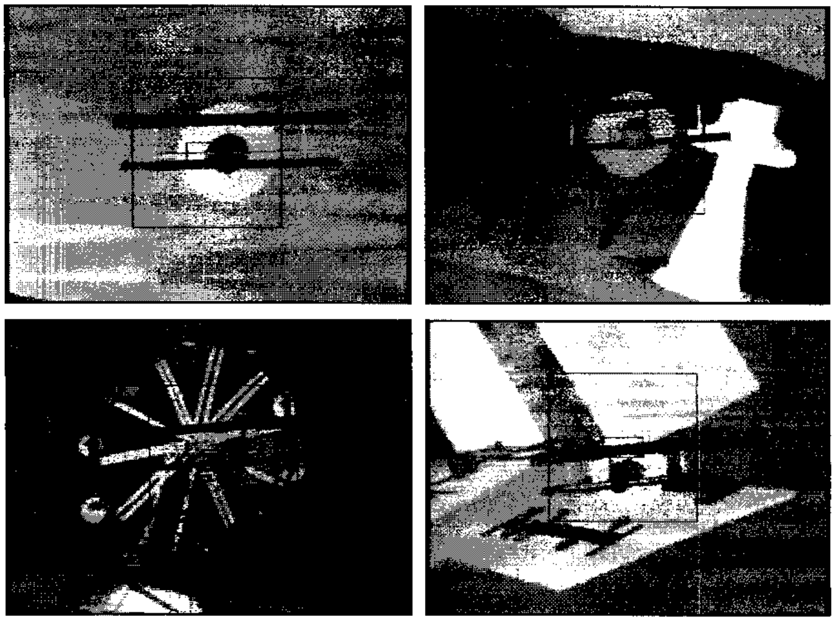
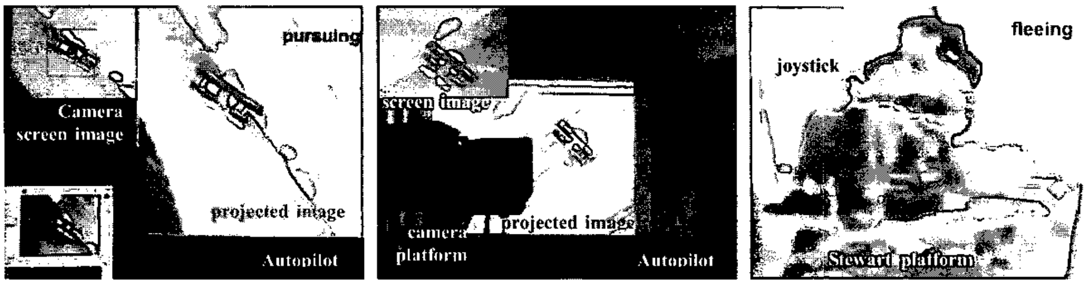


Fig. 10. Tracking the air fighter in the various VR environment.



(a) (b) (c)

Fig. 11. (a) and (b) The autopilot ability shown from the viewpoint of the pursuer.
(c) The pilot and the force-feedback components of the fleeing air fighter.