

Active Tracking Using Intelligent Fuzzy Controller And Kernel-Based Algorithm

Moteaal Asadi Shirzi

Assistant Supervisor and Administrative Manager of
Control Laboratory, University of Tehran
Tehran, Iran
Moteaalasadi@gmail.com

M.R. Hairi-Yazdi

Associate Professor and Supervisor of Control
Laboratory, University of Tehran
Tehran, Iran
Myazdi@ut.ac.ir

Abstract—This paper introduces a practical system by combining an Intelligent Fuzzy Controller and vision processing algorithm to obtain an efficient active tracker. Target tracking performance is heavily dependent on a good blend of vision algorithm and control. Because of performance and computational complexity, many visual tracking algorithms cannot be linked with control systems in the real-time tracking. Robustness and speed are the bottlenecks of current visual tracking algorithms. Here, the target's visual model is used along with a kernel-based searching algorithm and a motion detection module to predict the target position. A model update is also incorporated to recognize when the target's appearance is changing due to variations in its position. An Intelligent Fuzzy Controller (IFC) is also synthesized to reach control performance objectives. The IFC used in this tracking system not only helps to track the target accurately in different situations, but also tries to find the target's presence after track loss. Additionally, the parallel processing technique has also been deployed to provide the desired accuracy and speed in real-time tracking. The idea has been implemented in an active camera system to track moving targets.

Keywords- *Intelligent Fuzzy Control, Kernel-Based Object Tracking, Active Tracking, Vision*

I. INTRODUCTION

With the rapid enhancement of computational power, object tracking has been an active research area in the computer vision in recent years. It plays an important role in robot tracking, monitoring and surveillance systems, virtual sport, human-computer interfaces [1], vision-based control, smart rooms, military and driver assistance, astronomy and medical science. There are three common approaches that are widely used in object tracking. One is based on Kalman filter or extended the Kalman filter [2]. The Kalman filter assumes that the probability distributions of the states are Gaussian. Another is mean shift [3], [4] which is a robust non-parametric method for climbing density gradients to find the peak of probability model. Third, particle filter as a widely used approach in object tracking is well suited to estimate state of non-linear, non-Gaussian dynamic systems [5], [6]. Tracking under the constraint of low computational complexity presents a challenge in the real time tracking applications.

Recent years have witnessed rapidly growing use of fuzzy control systems in engineering applications. The numerous successful applications of fuzzy control have sparked a flurry of activities in the analysis and design of fuzzy control systems [7]. Stability and robustness are the two major points where the active tracking system presented can benefit from fuzzy controller [8]. Use of different fuzzy rules, depending on situations such as velocity and dimension of target, results in a more efficient controller to track the target. In addition, the tracker can detect the target's presence after track loss by running suitable fuzzy rules. Selecting the appropriate fuzzy rules is a complicated process. Thus, to implement these rules, an intelligent module has been used that runs special fuzzy rules with regard to situations. Therefore, the fuzzy controller is adaptive and the rules change in fuzzy engine intelligently.

In this paper, the target's visual model is used along with a kernel-based searching algorithm and a motion detection module to predict the target position. A model update is also incorporated to recognize when the target's appearance is changing due to variations in its position. An Intelligent Fuzzy Controller (IFC) is also synthesized to reach control performance objectives. In case of target track loss, a search algorithm sweeps the space in the vicinity of last target position to recover the lost target. Additionally, the parallel processing technique has also been deployed to provide the desired accuracy and speed in real-time tracking. The idea has been applied to an active camera system.

II. PROBLEM DEFINITION

The main goal is to track a dynamic target using a camera with two degrees of freedom in pan and tilt directions. This research centers on the tracking of two types of objects. The first type includes objects that are close to the camera such as faces of people. The active camera tracks people's faces regardless of whether they are approaching the camera or moving away from it or turning their heads. This example is expandable to a variety of objects that move around the camera. Sudden and rapid changing of targets aspect and their surroundings is one of the most important properties of them. The environment of target may be cluttered with other objects. Therefore, it is very important that the active camera be able to determine changes and differentiate between objects. The second type includes targets at long distances from the active

camera, such as an aircraft. These objects have small aspect in the camera with a small rate of changes. A small unsuitable movement of the active camera causes large tracking errors for distanced objects. Accuracy is therefore crucial.

III. GENERAL OPERATION

Figure 1 shows the schematic of operation of the active camera. Each of pan or tilt angles uses the same operation that is mentioned in figure 1. There are four major modules: The Intelligent Fuzzy Controller (IFC) module makes the appropriate control signal for plant due to errors. The Active Camera module is the sensors, digital camera and motors of the active camera transform the control signal to physical movements in pan and tilt movements. The vision box recognizes the selected dynamic target that must be tracked. In fact, the main goal is to keep the object in the center of the camera. The distance between object and center of the camera is determined as the error, figure 2. The filter box checks the output data coming from the vision box. Filter box uses some rational rules to reject idle data latest In case of target loss, the suitable fuzzy rules runs and an intelligent local search algorithm sweeps the scene in the vicinity of last target position.

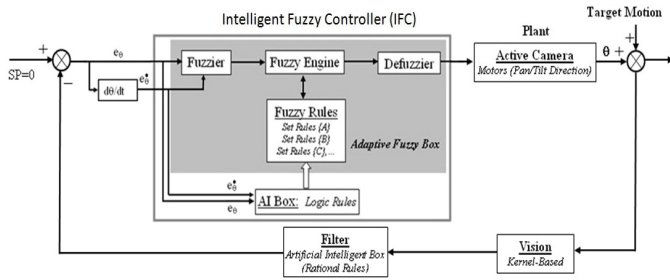


Figure1. The schematic of general operation

IV. VISION BOX

As mentioned, vision box recognizes the object and calculates the distance between object and the center of screen camera. This distance transforms to the two angle errors, $\Delta\theta_{tilt}$ and $\Delta\theta_{pan}$.

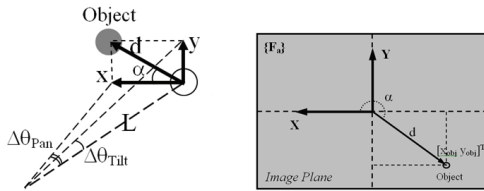


Figure2. $\Delta\theta_{tilt}$ and $\Delta\theta_{pan}$ are calculated from the distance of the object and the center of screen camera

$\Delta\theta_{tilt}$ and $\Delta\theta_{pan}$ are calculated from equation (1). In this formula, L is a constant value to normalize $\Delta\theta_{tilt}$ and $\Delta\theta_{pan}$.

$$\begin{aligned}\Delta\theta_{pan} &= d \sin \alpha / L \\ \Delta\theta_{tilt} &= d \cos \alpha / L\end{aligned}\quad (1)$$

Vision box uses various techniques to recognize and track objects. Following, briefly introduces these techniques.

V. MEAN SHIFT

Recently, the mean shift algorithm was employed for object tracking, measuring the similarity between consecutive frames based on Bhattacharyya coefficient [9]. A major limitation of the mean shift tracker is that it uses one object model to track the target during the entire tracking process. Hence, it fails to track the target as it undergoes severe deformations and rotations. To achieve robust tracking, model updates should be considered. The adaptive Kalman filter can be used to filter the object kernel histogram and obtain the optimal estimate of the object mode [10]. Another problem is also experienced with mean shift tracker. It cannot track objects apart for more than the size of the target model between two consecutive frames. This means that if the camera is unable to capture the target movement between these two frames, it virtually fails to track it. An extension of mean shift algorithm employs spatiograms (instead of histograms) which improves the performance of the mean shift tracker under wild rotations of the target [9]. However, it does not solve the second problem mentioned previously. Thorough global template search for the target is not a practical solution as it burdens the computer in real-time. More importantly, it is not suitable for real time tracking because of the controller stability issues that can result from erratic sampling rates. However, use of spatiogram has one evident advantage over histograms due to the integration of distance information that helps distinguish between different objects of similar color distribution.

Target Representation

To characterize the target, first a feature space is chosen. The reference target model is represented by its pdf (probability density function), q , in the feature space. For instance, the reference model can be chosen to be the color pdf of the target. In the subsequent frame, a target candidate is defined at location y , characterized by the pdf $p(y)$. To satisfy low computation cost imposed by real time processing, discrete densities or m -bin histograms should be used. Thus:

$$\begin{aligned}\hat{q} &= \{\hat{q}_u\}_{u=1\dots m} \quad \sum_{u=1}^m \hat{q}_u = 1 \\ \hat{p}(y) &= \{\hat{p}_u\}_{u=1\dots m} \quad \sum_{u=1}^m \hat{p}_u = 1\end{aligned}\quad (2)$$

Target model

A target is represented by an ellipsoidal region on the image. To eliminate the influence of different target dimensions, all targets are first normalized to unit circle. Let $\{x_i\}_{i=1\dots n_h}$ be the normalized pixel locations of the target region in the current frame. The probability of feature u in the target model is calculated from:

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i\|^2) \delta[b(x_i) - u] \quad (3)$$

Where $b(x_i)$ is the index of the bin at the pixel location X_i and δ is the *Kronecker delta function* and k is an isotropic kernel profile. The presence of such a kernel function assigns more weight to central pixels and reduces the effect of

peripheral pixels. The constant, C , is computed by using the fact that the sum of all probabilities over the *histogram* is one. Figure 3 shows an example of *kernel histogram* construction.

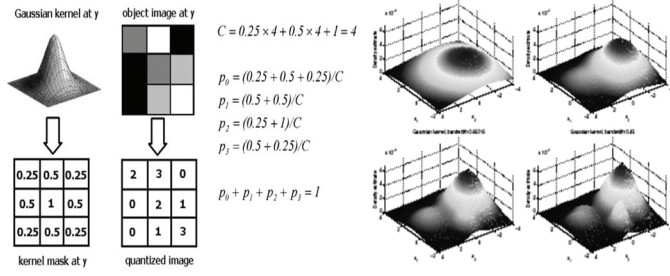


Figure3. Example of kernel histogram construction, Multi-bandwidth density estimate on 2D artificial Gaussian mixture data.

Target Candidates

Let $\{x_i\}_{i=1 \dots n_h}$ be the normalized pixel locations of target candidate, centered at y in the current frame. The probability of feature u in the target candidate is given by:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i)-u] \quad (4)$$

Where h is the bandwidth (number of pixels in the region) of target candidate and C_h can be calculated the same as C_h . The similarity function ρ is defined by the *Bhattacharyya coefficient* as:

$$\rho(y) = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \cdot \hat{q}_u} \quad (5)$$

Target localization

Localization maximizes the similarity function between the two probability density functions. The search for new target location starts at the location y_0 of the target in the previous frame. The Taylor expansion of $\rho(y)$ around $\rho(y_0)$ yields:

$$\rho(y) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \cdot \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(\hat{y}) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (6)$$

Taking the partial derivative of $\rho(y)$ with respect to y and setting it to zero results in the mean shift equation. In this procedure, the kernel is recursively moved from the current location y_0 to the new location y_1 , therefore:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i \cdot w_i \cdot g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i \cdot g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} \quad (7)$$

Where,

$$g(x) = -k'(x)$$

Kernels of *Epanechnikov profile* are typically used, i.e.:

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Where C_d is the volume of unit d -dimensional sphere and d is the dimension of the state space (2 in this case) [5]. In this case, derivative of the kernel profile, $g(x)$, is constant and equation (7) reduces to:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i \cdot w_i}{\sum_{i=1}^{n_h} w_i} \quad (9)$$

‘Hence, the mean shift algorithm calculates the similarity function for the target model and target candidate in subsequent frames and then, estimates the location of the target candidate using (9). This process is repeated until the similarity function goes up and the estimated location of the target changes less than a desirable threshold.

Target model update

Updating the target model during the tracking process is important if this process is to sustain for a long period of time. The generic target model update procedure can take on the following form:

$$q_{t+1} = c \cdot (q_t + f(\rho(q_t, p_t), \alpha, \dots) \cdot p_t) \quad (10)$$

Where c is an arbitrary constant that is usually one; f is a function of the similarity, ρ , between the current target model and candidate model; $\alpha, \dots$ are other visually extractable elements such as the perimeter of object in pixel. In (11), f is taken as an exponential function as follows:

$$f = e^{-a[1-\rho(q_t, p_t)]} \quad (11)$$

Where a is the update rate and was taken as 10. In this paper an update function also is used in a similar fashion. The function has the advantage of having the least effect on target model when the similarity function is obtained too low. Hence, even if the target suddenly is buried in scene clutter the updating procedure will have no significant effect on the previous target model and it still holds the visual characteristics of last-seen target.

VI. FILTER

The vision module produces the error value. It is possible that this module produces idle error value, especially when the active camera loses the object or when the environment is very complex and vision module makes mistake in recognizing the object. Some simple logic laws are used in the filter module to check and eliminate the idle error value.

Any target in the real world such as airplane, missile, the human face or anything that the active tracker wants to track, has a continuous movement. It means that targets could not be vanished and appeared in other place immediately. So the big change in error value could not be acceptable and the filter assumes it as an idle error value and eliminates it from the data. In addition, velocity and acceleration of the object are also continuous. The filter part the big error value is replaced with the value that calculates from this formula:

$$\begin{aligned} \text{if } (\Delta\theta_{pan}^{t_i} > \alpha_{pan}) &\Rightarrow \Delta\theta_{pan}^{t_i} = 0.5(\Delta\theta_{pan}^{t_{i-1}} + \frac{\Delta\theta_{pan}^{t_{i-2}} + \Delta\theta_{pan}^{t_{i-3}}}{2}) \\ \text{if } (\Delta\theta_{tilt}^{t_i} > \alpha_{tilt}) &\Rightarrow \Delta\theta_{tilt}^{t_i} = 0.5(\Delta\theta_{tilt}^{t_{i-1}} + \frac{\Delta\theta_{tilt}^{t_{i-2}} + \Delta\theta_{tilt}^{t_{i-3}}}{2}) \end{aligned} \quad (12)$$

Where $\alpha_{pan}=100$ and $\alpha_{tilt}=70$. If the active camera couldn't find the target, the search algorithm is operated. In other case, if the vision box loses the target, it produces a null value. This situation happens when the target hides behind other objects. For example, when an airplane goes behind clouds, the vision module can't recognize the position of the target. Thus it produces null value. The logic law that is used here is that the target continues its motion according to its last velocity and acceleration. In this case, filter module estimates the position of the target before it disappeared and produces a proximate error value.

$$if(\Delta\theta_{pan}^i > null) \Rightarrow \Delta\theta_{pan}^i = \Delta\theta_{pan}^{i-1}, if(\Delta\theta_{tilt}^i > null) \Rightarrow \Delta\theta_{tilt}^i = \Delta\theta_{tilt}^{i-1} \quad (13)$$

Other task of the filter module is to find the lost target. By following human behavior the suitable logic law can be extracted for the active tracker when the tracker loses the target. Actually, when the active tracker loses the target, first it tracks the estimated position proportional to the velocity and acceleration of the target. But if it fails to find the target, the search algorithm is activated. For finding the lost target, first the active camera searches around the target's lost position and it gradually increases the radius of the search area with increasing the rotation pan and tilt angles.

VII. INTELLIGENT FUZZY CONTROLLER

Fuzzy logic is a powerful albeit somehow controversial technique. It permits control without extensive knowledge of the equations of the process and it represents in a very effective way the human reasoning methods [11]. A fuzzy control designer is inevitably faced with two questions: How does one determine the shape of the fuzzy sets, and how many sets are necessary and sufficient? [12] There are some different ways to determine the shape of the fuzzy sets in a controller, based on expert experience and control engineering knowledge, based on the operator's control actions, based on a fuzzy model of the process and based on learning [13].

While non-adaptive fuzzy control has proven its value in some applications, it is sometimes difficult to specify the rule-base for some plants. Additionally, the need could arise to tune the rule-base parameters if the plant changes. This provides the motivation for intelligent fuzzy control, where the focus is on the automatic on-line synthesis and tuning of fuzzy controller parameters [32]. There are several techniques for determining and tuning the fuzzy rules that have some strength and weaknesses which make them suitable for a specific situation. The *Intelligent Fuzzy Controller (IFC)* uses collections of fuzzy rules that each of them is efficient for a specific situation. Fuzzy laws are selected by using intelligent laws in accordance to the situation of the target. Some of these intelligent rules are as follows:

- Distanced targets need high accuracy to be tracked. Every change of the pan and tilt angles at the active camera causes large changes in vision area of the active camera at faraway. Therefore a small unsuitable movement of the active camera causes large tracking errors for faraway targets.

- Because the image of close targets in the camera move quickly and randomly, these targets could be tracked with less accuracy but with higher tracking velocity. High accuracy tracking may result in losing the target.

Three collections of different fuzzy set rules are used in the IAFC:

1. For targets with very slow velocity that are near the camera or faraway targets with slow velocity: if $\Delta\theta_{pan} < 4.5 \text{ deg/sec}$ and if $\Delta\theta_{tilt} < 2.5 \text{ deg/sec}$ the IAFC uses fuzzy set rules that are mentioned at figure 4.
2. For faraway targets with high velocity or near targets with medium velocity, if $4.5 < \Delta\theta_{pan} < 8 \text{ deg/sec}$ and if $2.5 < \Delta\theta_{tilt} < 4.5 \text{ deg/sec}$, the IAFC uses figure 5 fuzzy set laws.
3. For near targets with fast velocity: if $\Delta\theta_{pan} > 8 \text{ deg/sec}$ and if $\Delta\theta_{tilt} > 4.5 \text{ deg/sec}$, the IAFC uses figure 6 fuzzy set laws.

Actually the tracker has been run frequently with variety of fuzzy rules. The suitable values and fuzzy laws are selected by considering the results of system operation. Changes in Pan Direction are more than tilt direction because objects generally move horizontally, so the signal controls are stronger for pan direction. For example there is a special collection laws for a target that is near the active camera and has high velocity. For this, accuracy is not very important. The main objective of the active tracker is to avoid losing the target. High accuracy in tracking the target causes losing the target, especially when the direction of target changes instantaneously. But if the velocity of the target is low, active camera can be more accurate in tracking by using other collection of fuzzy laws.

Angle Pan $\alpha = \Delta\theta_{pan}$ Tilt $\beta = \Delta\theta_{tilt}$	Angular Velocity $\dot{\alpha}$ Velocity $\dot{\beta}$	Controller Signal for Pan Direction	Controller Signal for Tilt Direction
Low (-)	Low (-)	High	Medium High
	Zero (·)	Medium High	Medium High
	High (+)	Zero	Zero
Medium Low	Low	High	Medium High
	Zero	Medium High	Medium High
	High	Zero	Zero
Zero (·)	Low	Medium High	Medium High
	Zero	Zero	Zero
	High	Medium Low	Medium Low
Medium High	Low	Zero	Zero
	Zero	Medium Low	Medium Low
	High	Low	Medium Low
High (+)	Low	Zero	Zero
	Zero	Medium Low	Medium Low
	High	Low	Medium Low

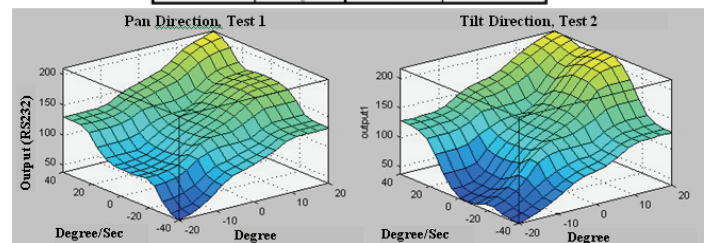


Figure 4. Selected fuzzy rules for targets with very slow velocity that are near the camera or faraway targets with slow velocity. In this case camera has fast motion, but accuracy is not much important. So, the main purpose of fuzzy rules is that the camera doesn't lose the target. Fuzzy rules produce strong commands to reduce big errors.

Angle Pan $\alpha = \Delta\theta_{pm}$ Tilt $\beta = \Delta\theta_{dp}$	Angular Velocity $\dot{\alpha}$ Velocity $\dot{\beta}$	Controller Signal for Pan Direction	Controller Signal for Tilt Direction
Low (-)	Low (-)	Very High	High
	Zero (-)	High	Medium High
	High (+)	Zero	Zero
Medium Low	Low	High	Medium High
	Zero	Medium High	Medium High
	High	Zero	Zero
Zero (-)	Low	High	Medium High
	Zero	Zero	Zero
	High	Low	Medium Low
Medium High	Low	Zero	Zero
	Zero	Medium Low	Medium Low
	High	Low	Medium Low
High (+)	Low	Zero	Zero
	Zero	Low	Medium Low
	High	Very Low	Low

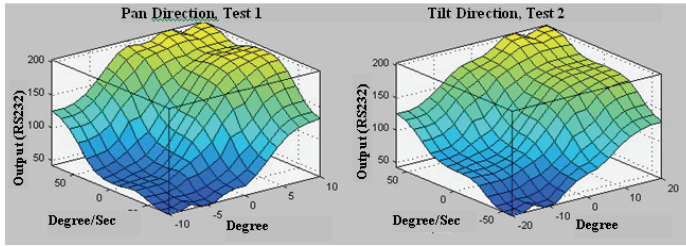


Figure 5. Selected fuzzy rules for faraway targets with fast velocity or near targets with medium velocity. In this case fuzzy rules prepare commands that cause camera tracks targets with moderate accuracy and velocity.

Angle Pan $\alpha = \Delta\theta_{pm}$ Tilt $\beta = \Delta\theta_{dp}$	Angular Velocity $\dot{\alpha}$ Velocity $\dot{\beta}$	Controller Signal for Pan Direction	Controller Signal for Tilt Direction
Low (-)	Very Low	Very very High	Very High
	Low (-)	Very High	Very High
	Zero	High	High
Medium Low	High (+)	Zero	Zero
	Very High	Low	Low
	Very Low	Very High	Very High
Zero (-)	Low (-)	High	High
	Zero	Medium High	Medium High
	High (+)	Zero	Zero
Medium High	Very High	Low	Low
	Very Low	Very High	Very High
	Low (-)	High	High
High (+)	Zero	Zero	Zero
	High (+)	Low	Low
	Very Low	Very High	Very High
Medium High	Low (-)	High	High
	Zero	Medium Low	Medium Low
	High (+)	Low	Low
High (+)	Very High	Very Low	Very Low
	Very Low	High	High
	Low (-)	Zero	Zero
Medium High	Zero	Low	Low
	High (+)	Very Low	Very Low
	Very High	Very Very Low	Very Low

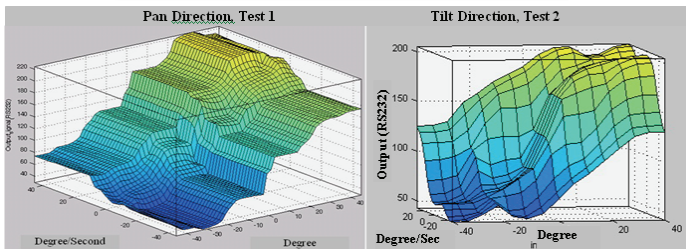


Figure 6. Selected fuzzy rules for near targets with fast velocity. In this case tracking does not need high velocity pan and tilt directions. But it needs to track targets accurately. So, Fuzzy rules produce commands that cause high accurate movement.

Owing to the interconnection of vision and control system, the optimization problem of active tracking is rather nontrivial. For instance, if robustness of tracking algorithm implies more latency, the control system would become unstable. Whereas, high overshoot of the closed loop system

will cause the vision processing program to lose sight of the target.

VIII. IMPLEMENTATION AND EXPERIMENTAL RESULTS

To achieve an automatic target tracker, motion detection, target tracking and IFC must be combined. To link these two components, computers have been parallelized by the UDP protocol for communications. The vision processor extracts the target's position from the image streams delivered by the camera and packages the data to be sent over the predefined UDP sockets. On the other end, the Matlab [®] sits in wait for the UDP packets, calculates the control signals by the fuzzy controller and relays commands to this active camera. The camera could track different objects such as human face or body excellently in both tilt and pan directions. Because of some unknown hardware flaws, sometimes, delays occurred in the tracker. This problem usually is solved because of the robustness of fuzzy controller. Details of active tracker and connections are shown in figure-7.

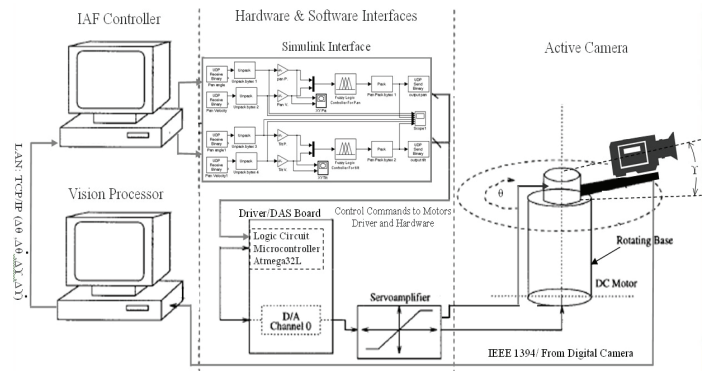


Figure7. The schematic of hardware and software of active camera

To assemble the active tracker, vision processor and fuzzy controller are each implemented on two separate computers. This way allows each portion to usurp full computation power of its own processor.

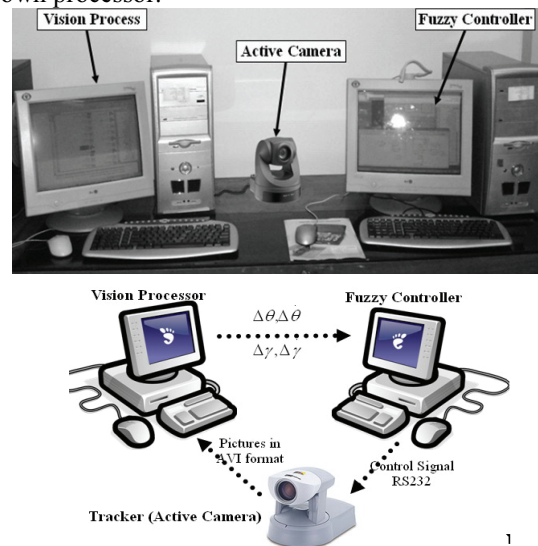


Figure 8. Using the parallel processing, the computational load is divided between two computers and delay times can decrease

Two different experiments are considered to test the fuzzy controller and vision algorithm. In the first experiment, the active camera tracks a black spot on a rotating circular plate which paces back and forth between two limits. The real position of the black spot is:

$$\theta = \tan^{-1}\left(\frac{r}{l}\right)\sin(\omega t) \quad , \quad \alpha = \tan^{-1}\left(\frac{r}{l}\right)\cos(\omega t) \quad (14)$$

Where θ is the pan angle and α is the tilt angle; r refers to the radius of rotation of the black spot and l is distance between the active camera and the circular plate; ω represents the angular velocity of plate and finally t denotes time.

In addition, the normal mean-shift algorithm and a conventional PID controller are developed for comparing by the operation of the IFC and considering of effectiveness of the proposed method precisely. Figure 9, 10 and 11 shows the results in terms of angle of camera and outputs from vision and controller that the active camera has tracked. Figure 9, 10 displays the operation of the normal mean-shift algorithm and a conventional PID controller beyond the IFC algorithm.

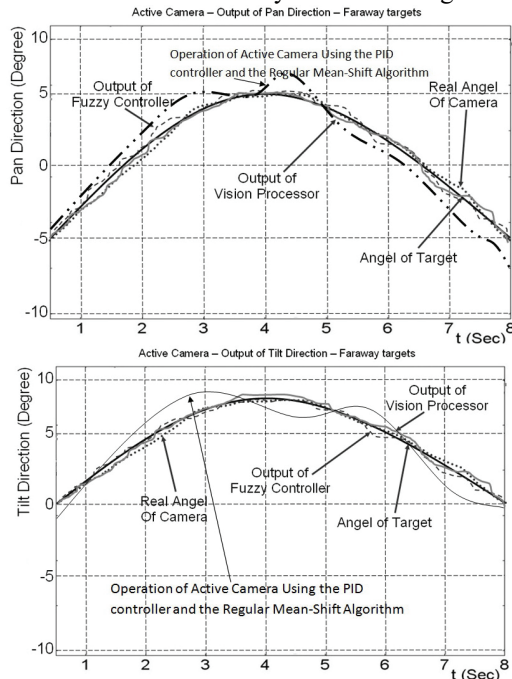


Figure9. Tracking the target with circular path($w=0.4\text{rad/sec}$). These figures show that the operation of the IFC is better than the operation of the normal mean-shift algorithm and a conventional PID controller

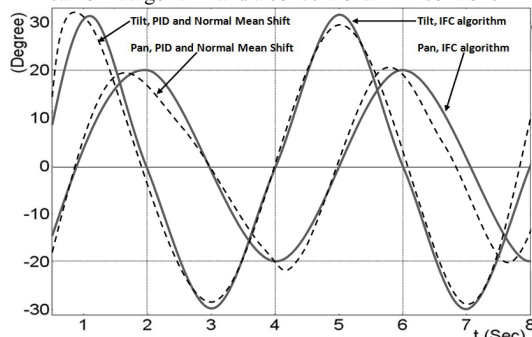


Figure10. Comparison the operation of the IFC and the operation of the normal mean-shift algorithm and a PID controller. The IFC algorithm decreases errors in the active tracker.

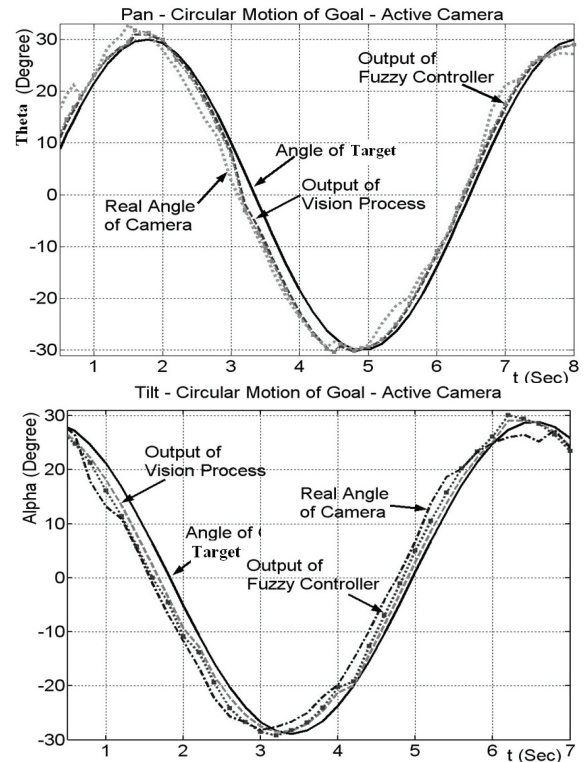


Figure11. Tracking the target with circular path ($w=1\text{rad/sec}$).

In the second experiment, a person with a constant velocity, moves around the camera and camera track his face. Figure 12, 13 shows the results of this experiment in the same way.

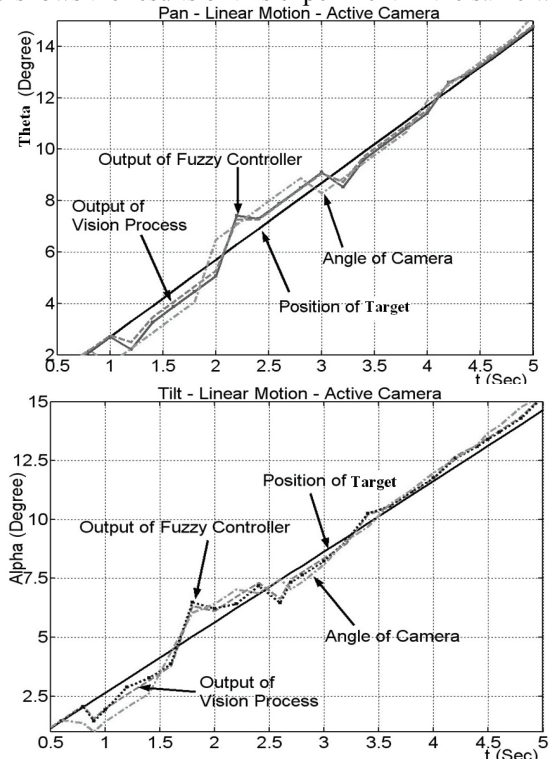


Figure 12. Tracking a person with a constant velocity



Figure13. The pictures show the operation of the vision processor box too

Figure 14 Shows the result of the tracking the faraway targets with high velocity

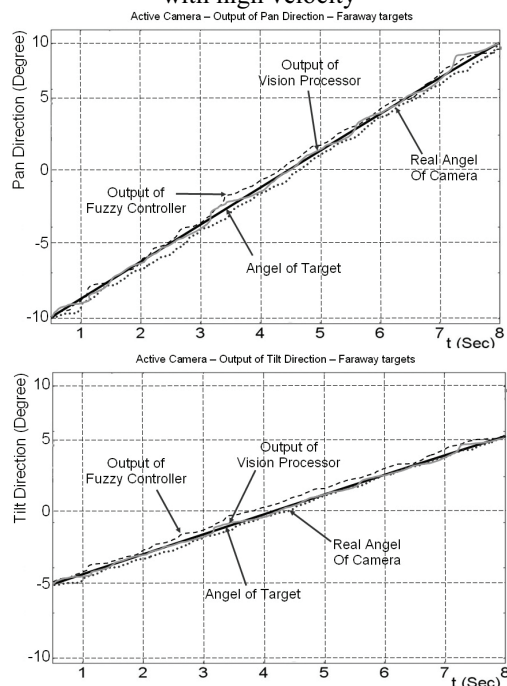


Figure 14. Tracking the faraway targets with high velocity

I. CONCLUSION

A new combination of vision and control have been presented with a new algorithm allowing a robust and accurate detection of moving objects for a small cost in memory consumption and computational complexity that allow us to use ordinary computers and cameras instead of expensive and high performance computers and vision facilities. Although the

set faces to delays especially in vision process box and connections (LAN), the IFC usually solves delay problems. In addition, considering the operation of the normal mean-shift algorithm and a conventional PID controller shows that the IFC has better performance. In fact, using the IFC helps the active tracker to miss targets less than the active cameras which use the normal mean-shift algorithm and a conventional PID controller. Undoubtedly, the set can improve by using of more fuzzy rules sets for more different situations and objects.

REFERENCES

- [1] F. Xu, J. Cheng, C. Wang, "Real Time Face Tracking Using Particle Filtering and Mean Shift", IEEE, International Conference on Automation and Logistics, ISBN: 978-1-4244-2502-0, pages: 2252 - 2255, 2008
- [2] J. Gao, A. Kosaka, A. Kak, "A multi-Kalman filtering approach for video tracking of human-delineated objects in cluttered environments", Robot Vision Lab, Purdue University, Computer Vision and Image Understanding, Volume 102, Issue 3, Page 259-264, 2006
- [3] A. Birchfield, S.T. Rangarajan, "Spatiograms versus histograms for region-based tracking", IEEE conference on computer vision and pattern recognition (CVPR), pages: 1158 -1163, vol. 2, 2005
- [4] D. Comaniciu, V. Ramesh, "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 25, NO. 5, Pages: 564-577, 2003
- [5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P.J. Nordlund, "Particle Filters for Positioning Navigation and Tracking", IEEE Transactions On Signal Processing, pages: 494 - 499, 2002
- [6] Z. Wang, X. Yang, S. Yu, "CamShift guided particle filter for visual tracking", Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Pattern Recognition Letters Journal, Elsevier, Volume 30, Issue 4, 2008
- [7] K. Tanaka, H. Wang, "Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach", John Wiley & Sons, 2001
- [8] A. Mustafa, M. Asadi Shirzi, M.R. Hairi Yazdi, "Active Tracking Using Kernel-Based Vision Processor and Robust Fuzzy Control", Advances in Intelligent Systems Research, Vol. 3, Page 273-280, JCIS 2006
- [9] N. Song Peng, J. Yang, Z. Liu, "Mean shift blob tracking with kernel histogram filtering and hypothesis testing", Pattern Recognition Letters, Elsevier, Volume 26, Issue 5, Pages 605-614, 2005
- [10] H. YungCheng, J.N. Hwang, "Adaptive particle sampling and adaptive appearance for multiple video object tracking", Volume 89 Issue 9, Signal Processing Journal, Elsevier, 2009
- [11] J.E. Naranjo, C. González, R. García, T. de Pedro, "Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping", IEEE, intelligent transaction systems, Volume 4, No. 3, Pages 132-142, 2003
- [12] Jan Jantzen, "A Robustness Study of Fuzzy Control Rules", Technical University of Denmark, Department of Automation, pages 31-36, 1998
- [13] Massa, Ruffin, Tison, Lallemand, "A complete method for efficient fuzzy modal analysis", Journal of Sound and Vibration, Elsevier, Volume 309, Issues 1-2, Pages 63-85, 2008
- [14] T. Hayakawaa, H. Ishii b, K. Tsumurac, "Adaptive quantized control for nonlinear uncertain systems", Systems & Control Letters, Elsevier, Volume 58, Pages 625-632, 2009
- [15] K. Choudhali, M. Jedra, N. Zahid, "Kernel relevance weighted discriminant analysis for face recognition", Pattern Analysis & Applications, 2010