

## **ADAPTIVE ESTIMATION ALGORITHM- BASED TARGET TRACKING**

**Mustafa Yağimli**

*Department of Electrical and Electronics Engineering,  
Turkish Naval Academy, Tuzla, Istanbul  
myagimli@dho.edu.tr*

**H. Selçuk Varol**

*Institute for Graduate Studies in Pure and Applied Sciences,  
Marmara University, Göztepe, Istanbul  
hsvarol@marmara.edu.tr*

### **Abstract**

*This study aimed at developing software that can detect moving target's orbits by processing the images provided by a camera. This algorithm was used to estimate their future movements. Systems codes were developed in the Visual Basic programming language.*

## **ADAPTİF TAHMİN ALGORİTMA-TABANLI HEDEF İZLEME**

### **Özetçe**

*Bu çalışmada, kameradan gelen görüntüyü işleyerek, hareketli hedeflerin yörüngesini tespit eden bir yazılım geliştirilmiştir. Bu algoritma, hedeflerin gelecekteki hareketlerinin tahmininde kullanılmıştır. Sistem kodları Visual Basic ortamında geliştirilmiştir.*

**Keywords:** *Target tracking, adaptive algorithm, moving targets.*

**Anahtar Kelimeler:** *Hedef izleme, adaptif algoritma, hareketli hedefler.*

## 1. INTRODUCTION

In today's world, the number of applications of automatic systems is increasing steadily [1].

A literature review showed that target tracking processes are studied using various methods and algorithms: Many different algorithms are used as well [2-5]. Real-time [8], fuzzy logic [9-10] is used, as well as image-based target tracking [6-7]. This fuzzy logic method provides a wider range and adaptive control methods are used in target tracking so as to ensure system continuity and to improve the capacity to adapt to possible changes [11-12].

## 2. SYSTEM STRUCTURE

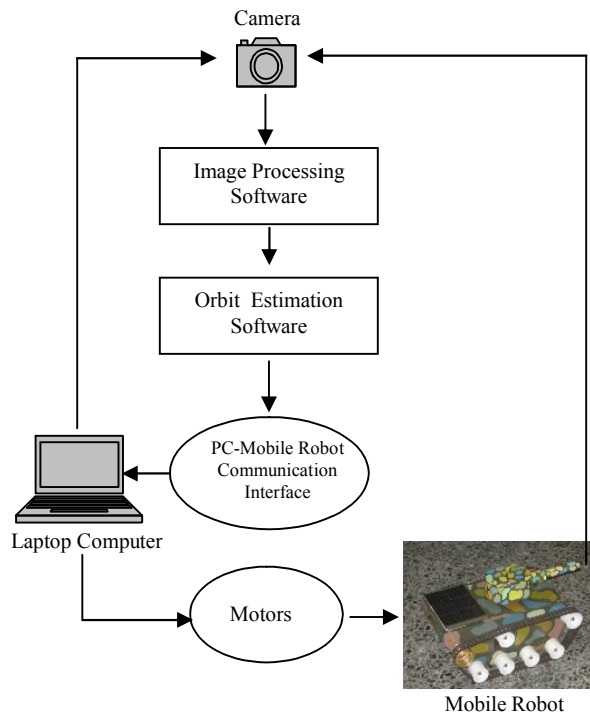


Figure 1: General Model of Systems.

The general model of the system is shown in Figure 1. System design is realized two main steps: image processing software and orbit estimation.

### **2.1. Image Processing Software**

The image processing software, which was used in both target recognition and target tracking units, is the part which takes the image from the camera and transmits information related to the target's location and image capture time to the orbit prediction software. The image is transferred to the computer environment using a wireless camera with a PCMCIA card interface and a 1/3 inch CMOS sensor. The camera and consequently the video stream from the camera is transferred to the program in real time during design and operation. In other words, the CapturePRO component was used in order to be able to add the camera to the form during design. Another component used in the program is Hitime.ocx. This component reduces calculation time, which is a crucial factor in image processing, to the level of microseconds. The program works on the basis of distinguishing the target from the background and then determining the geometrical centre of this identified section.

The software is based on differentiating the target from the background and determining the geometric center.

In the image received from the camera, all the pixels in the form of rows and columns are checked, and pixels at a certain color level are identified and transferred to a series. Thus, the location information of the target is obtained in matrix form [13].

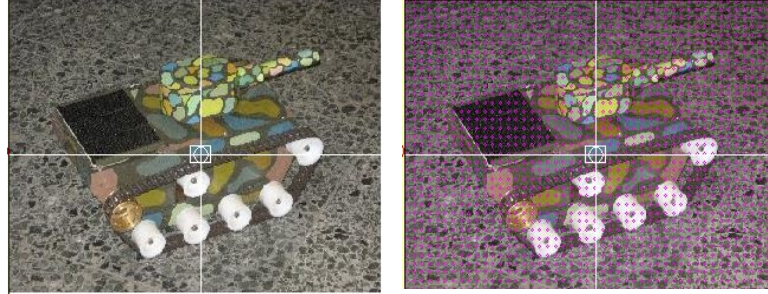


Figure 2: Scanning of the Target Image.

In Figure 2, the target values can be seen with all details. By scanning the rows and columns, pixels which have the same color are differentiated and the colors which can not be detected are accepted as empty space. Thus, the target can be differentiated from the environment.

As can be seen in Figure 2, pixels are comprised of 628x582 sub units, each with an equal area. The middle point of the object can be found using these pixels. The formula which gives the middle point of the object is as follows [14]:

$$x_g = \frac{\int \chi dL}{\int dL}, \quad y_g = \frac{\int y dL}{\int dL} \quad (1)$$



Figure 3: Distances to the Point Taken as Reference for the Calculation of the Middle Point

### *Adaptive Estimation Algorithm-Based Target Tracking*

In Figure 3, the distances to the point taken as reference for the determination of the middle point of the object are shown. When calculating the middle point, the entire image received from the camera is considered as an object, and the locations outside of the object were treated as empty space. As the image is comprised of 628 rows and 582 columns, the formula can be expressed as;

$$x_g = \frac{\int_0^{628} x dl}{\int_0^{628} dl}, \quad y_g = \frac{\int_0^{582} y dl}{\int_0^{582} dl} \quad (2)$$

The coordinates of the middle point are found as an ordered series obtained by calculating the Euclidean distance (E) between the middle point  $(x_1, y_1)$  and each border pixel  $(x_i, y_i)$ . When creating an ordered series, the area of each pixel was considered a unit square. Euclidean distance describes the shortest path between two points. Here, as  $i=1, 2, 3, \dots, N$ , it is

$$E = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2} \quad (3)$$

N indicates the number of corner pixels. The Euclidean distance between pixels P and Q is calculated as follows for  $P=(x_1, y_1)$  and  $Q=(x_2, y_2)$  points [15]:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Starting from the middle point of the object, the pixels which have the specified color values are determined in the images received from the camera. The intersections of the Euclidean distances of each of these identified pixels is found relative to a reference point. The intersection found gives the middle point of the recognized target. For instance, let's assume that an object comprised of 3x4 pixels as in Figure 4 exists:

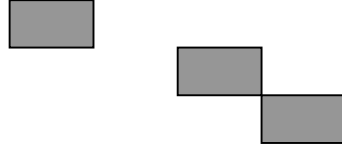


Figure 4: Image Comprised of Pixels Numbered 1-7 and 12.

Considering that the pixels numbered 1, 7 and 12 represent the image and colors of the selected object, the software identifies the middle point relative to a certain reference point by scanning these pixels as follows:

The origin point of the coordinate axis (0, 0) is accepted as the reference point. It is assumed that the image is in the pixels numbered 1, 7 and 12, and that there is no image in the other pixels.

When these pixels are selected, an object as in Figure 5 is obtained.

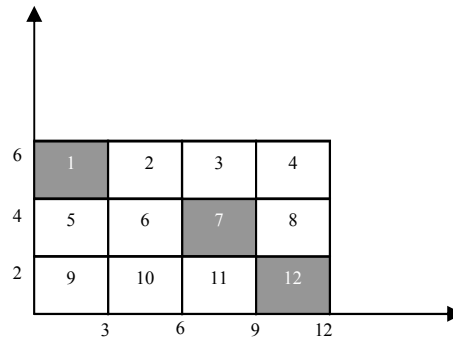


Figure 5: Coordinates of the Pixels (x, y).

Euclidean distance gives the intersection of the middle points of the pixels selected, the middle point of the object. The middle point of the overall shape;

## Adaptive Estimation Algorithm-Based Target Tracking

$$I_{x_1} = \frac{\Sigma Ax}{\Sigma A} = \frac{1.72}{12} = 6$$

$$I_{y_2} = \frac{\Sigma Ay}{\Sigma A} = \frac{1.36}{12} = 3$$

$I_x$  and  $I_y$  give the middle point of the selected pixels. The middle point of this object can also be found without subtracting the empty pixels, by adding the middle points of each part one by one relative to the reference point. Whilst the total area of the object is 3 (unit)<sup>2</sup> each has 1 (unit)<sup>2</sup> area.

### 2.2. Orbit Estimation

Orbit estimation was done via the adaptive algorithm method. Self-controlled, self-replaced and adaptive systems are called *adaptive controlled systems*. Such systems can adapt to new incidents and behavioral changes. In the scope of this study, unknown input and maneuvering instructions were modeled as random processes and were expressed on a real-time basis.

A block diagram of a system with adaptive control is shown in Figure 6. The most important feature of adaptive control is that it can adapt to the change of the system when solving the change in the system, and that it can provide the most suitable control required for the system. A system where feedback is made at a fixed value is not adaptive, and only a control system in which arrangements in the system's software and equipment can be made adaptive.

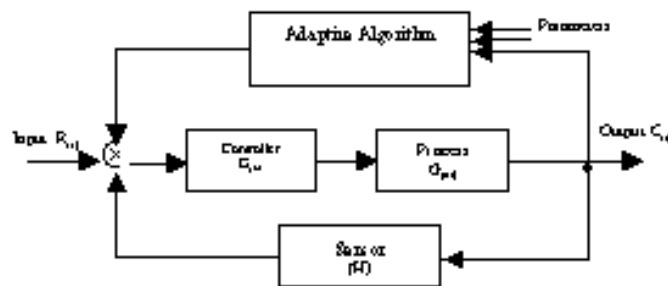


Figure 6: Block Diagram of a System with Adaptive Control [16].

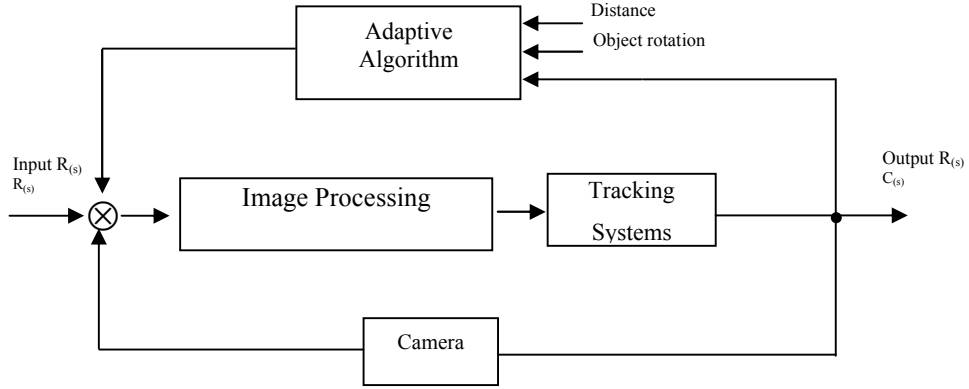


Figure 7: Block Diagram of the Tracking System.

In adaptive prediction, moving targets are represented by the following equation [17]:

$$\chi(k+1) = F(k)\chi(k) + G(k)u(k) + v(k) \quad (4)$$

$F(k)$ ,  $G(k)$  form the predicted situation matrix, and  $u(k)$  forms the data independent of operation. In the scope of this study, unknown input and maneuvering instructions were modeled as random processes and were expressed on a real-time basis. Estimation input;

$$\chi(k+1) = F\chi(k) + Gu(k) + v(k) \quad (5)$$

Estimation ability;

$$z(k) = H\chi(k) + \omega(k) \quad (6)$$

Situation matrix in non-moving targets which don't have an entry;

$$\chi(k+1) = F\chi(k) + v(k) \quad (7)$$

are represented.

$u$  indicates the unknown entry of the target movements and  $v$  represents the operation noise.



### **3. CONCLUSION**

While high resolution cameras provide an advantage in image transfer, low resolution cameras will provide an advantage in terms of system speed. Using high speed or parallel processor computers will also increase the system speed. The quicker the target matrix can be generated, the quicker the system tracking speed will be. The fundamental factor affecting the speed and sensitivity is the speed of the control unit which generates the system control value. As the units forming the system are connected to each other serially because the adaptive prediction algorithm is used as a type of control, a delay in one unit will slow down the whole system.

### **REFERENCES**

- [1] Economic Commission for Europe, *World 2004 Robotics*, Oxford University Press (2004).
- [2] Z. Yanning, Z. Jiangbin, Z. Rongchun, A Hybrid Method For Efficient Target Recognition, *Journal of Electronics*, Vol. 20, No. 1, Jan, (2003).
- [3] G. Edwards, P.T. Jeffrey, Target Recognition and Classification Using Neural Networks, *IEEE*, 0-7803-7625-0/02 (2002).
- [4] M. A. Glover, P.C. Haynes, C. J. Hollier, D.J. Lees, Temporal Sensors for Long Range Target Recognition and Identification, *3<sup>rd</sup> EMRS DTC Technical Conference*, Edinburgh, (2006).
- [5] E. W. Libby, P. S. Maybeck, Application of Sequence Comparison Techniques to Multisensor Data Fusion and Target Recognition, *Proceedings of the 32<sup>nd</sup> Conference on Decision and Control*, San Antonio, Texas, December, (1993).
- [6] Y. Nakabo, I. Ishii, M.Ishikawa, Moment Feature-Based Three-Dimensional Tracking Using Two High Speed Vision Systems, *Advanced Robotics*, Vol. 17, Number 10, pp. 1041-1056, (2003).
- [7] M. Y. Cheng, M. C. Tsai, C. Y. Sun, Dynamic Visual Tracking Based on Multiple Feature Matching and g-h Filter, *Advanced Robotics*, Vol. 20, Number 12, pp. 1401-1423, (2006).
- [8] S. Plermkamon, N. Afzulpurkar, An Intelligent Real-Time Tracking and Grasping System for a Robotic Work Cell, *Advanced Robotics*, Vol. 17, Number 5, pp. 403-425, (2003).

- [9] T. S. .Li, S. Chang, W. Tong, Fuzzy Target Tracking Control of Autonomous Mobile Robots by Using Infrared Sensors, *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 4, 1063-6706/04, August, (2004).
- [10] C. Kia, M. R. Arshad, P. A. Wilson, Supervisory Fuzzy Learning Control for Underwater Target Tracking, *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 6, 1307-6884, June, (2005).
- [11] M. K. Ciliz, Combined Direct and Indirect Adaptive Control of Robot Manipulators Using Multiple Models, *Advanced Robotic*, Vol. 20, Number 4, pp. 483-497 (2006).
- [12] O. Parlaktuna, M. Ozkan, Adaptive Control of Free-Floating Space Robots in Cartesian Coordinates, *Advanced Robotics*, Vol. 18, Number 9, pp. 943-959, (2004).
- [13] M. Yağımlı, H. S. Varol, Low Cost Target Recognising and Tracking Sensory System Mobile Robot, *Journal of Naval Science and Engineering*, Vol. 4, No. 1, pp. 17-26, (2008).
- [14] A. Bedford, W. Fowler, *Engineering Mechanics Statics*, Addison-Wesley Publishing Company, Inc, ISBN: 0-201-58193-0, p. 336, (1994).
- [15] C. Grönwall, F. Gustafsson, M. Millnert, Ground Target Recognition Using Rectangle Estimation, *IEEE Transaction on Image Processing*, Vol. 15, No. 11, November, (2006).
- [16] K.J. Astrom, B. Wittenmark, *Adaptive Control*, Addison-Wesley Publishing Company, Sweden.