Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A UAV Vision System for Airborne Surveillance

M. Kontitsis, Kimon P. Valavanis
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida, USA
mkontits@mail.usf.edu, kvalavan@csee.usf.edu

N. Tsourveloudis
Department of Production Management and Engineering
Technical University of Crete
Chania, Greece
nikost@dpem.tuc.gr

***Abstract*—The paper presents a machine vision system for aerial surveillance that can interpret and process data acquired by a UAV on-board infrared camera. System components include noise reduction, feature extraction, classification and decision-making. Decision-making is performed in terms of an alarm signal. The system has been configured for automatic fire-detection applications where the alarm is set off in case of fire identification. Real time tests have been performed and the system has been tested producing sets of real images. Finally, a genetic algorithm was used to automatically define some of the system's parameters.**

Keywords-machine vision; surveillance; UAV; genetic algoritms;

I. INTRODUCTION

Airborne surveillance has been proven to be important and applicable to a wide range of applications, such as search and rescue missions, border security, resource exploration, wildfire and oil spill detection. Until recently, common practice has been to use manned aircraft equipped with special sensors and to assign the actual recognition task (surveillance) to the crew, or record image data and analyze them off-line on the ground [1]. In exceptional cases where Unmanned Aerial Vehicles (UAVs) are used, they are treated as sensor carrying platforms transmitting data to a ground control station (GCS) for analysis, since the involved UAVs lacked the ability of local on-board intelligence for data interpretation [2].

The objective of this research is the design of a (machine) vision system that enables an UAV to acquire and interpret data in real time, followed by decision-making in terms of signaling an alarm, while flying over a specified (targeted) area. Although the discussed system is general enough, the specific application under consideration is the automated fire detection. The UAV may be fully autonomous, semi-autonomous or teleoperated according to application or mission requirements. However, this does not affect the design of the system presented in this paper.

The paper is organized as follows: Section II provides a description of the proposed system. It starts with an overview of each module and concludes with remarks related to the computational complexity of the used algorithms. In Section III, a case study is presented. The system is configured for forest fire detection; results include real images. Section IV refers to the derivation of a genetic algorithm (GA) that helps automating the design process of the system. Section V concludes the paper.

II. SYSTEM DESCRIPTION

A. System Overview

The proposed machine vision system consists of the following components: Noise reduction component, feature extraction component, feature vector classification component, and alarm raising component.

It is considered that images are acquired using an Infrared (IR) or a Near Infrared NIR camera (8-bit grayscale bitmap). Images are subjected to preliminary image processing, namely gauss filtering for noise reduction purposes. Then, each image is segmented to various regions and the *size* and the *mean intensity* of each of these regions are selected as features. This procedure produces a series of feature vectors for each image. The vectors are fed to a fuzzy classifier, which assigns to each region a number indicating the possibility of the corresponding region being the objective. All regions classified as having a high possibility of being the object are kept in a registry. If a certain region is persistently returning a high possibility, the alarm is set off.

The block diagram of the system is shown in Figure 1. It is important to notice that due to the nature of the image sensors (IR/NIR camera) the image is a representation of the energy emitted by the various objects in the electromagnetic spectrum from 1μm to 14μm. These objects, especially those with a strong signature in the 3μm to 14μm band [3], are thermal sources that are likely to be the application objectives.

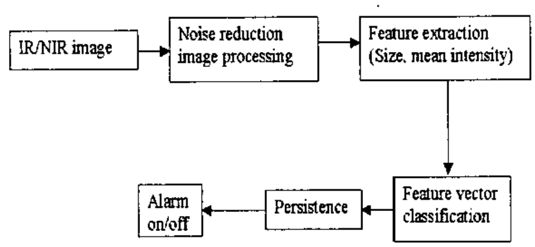


Fig.1: The block diagram of the machine vision module

B. Noise Reduction Component

The image acquired by the IR/NIR camera is, like any other signal generated by any sensor, subject to noise. To mitigate this undesirable effect, a spatial Gaussian filter is used, defined as:

$$g[i,j] = ce^{-\frac{(i^2+j^2)}{2\sigma^2}} \qquad (1)$$

where σ is the deviation and c a normalizing constant. Choosing σ=1 and a size of (5x5) matrix, the following mask is formed [4]:

$$g = \frac{1}{348}\begin{bmatrix} 1 & 5 & 8 & 5 & 1 \\ 5 & 20 & 34 & 20 & 5 \\ 8 & 34 & 56 & 34 & 8 \\ 5 & 20 & 34 & 20 & 5 \\ 1 & 5 & 8 & 5 & 1 \end{bmatrix} \qquad (2)$$

A larger filter, say 7x7, could offer a better representation of the gaussian function at the expense of some extra computational cost. The use of a smaller one could of course reduce the amount of computations needed but it would sacrifice the representation accuracy of the gaussian bell. The 5x5 is a good compromise between the need for adequately representing the function in discussion and the necessity of a low computational cost.

The selection of this kind of filter has been favored by its ability to remove noise without significant information loss.

C. Feature Extraction Component

A very important task in the surveillance process is to define the object the system should look for and to select features describing it in an almost unique manner. Based on the assumption that image regions corresponding to the application's target have different intensity values compared to their surroundings, the region's *mean intensity* is selected as a feature. For example, in an application regarding forest fire detection, targets that are trees on fire, are shown on the image as regions with higher mean intensity [3], which is defined as:

$$\text{Mean intensity} = \frac{\sum_{\forall \text{pixel} \in \text{region}} grayvalue\_of\_pixel}{\text{number of pixels in the region}} \qquad (3)$$

Another feature that may be used to discriminate among regions is the *size*. This helps in the definition of the objective since a specific size range can be favored.

Having chosen the features, all that remains is to extract them from the image. This is done by segmenting the image using a region growing by pixel aggregation algorithm [4][5]. Such an algorithm consists of the following steps:

a) Select a threshold, *Tseed* for the intensity of the pixels that will be chosen as seed points.

b) Select the seed points

c) Grow the regions by appending to each seed point those neighboring pixels that have similar properties.

The threshold *Tseed* is given a high grayscale value (above 160 in a 0 to 255 scale) in accordance with the assumption that targets appear brighter on the image. Recall that the image is an 8-bit grayscale so the intensity of the pixels take 2^8 distinct values ranging from 0 (black) to 255 (white). The local maxima of the image that have an intensity value higher than *Tseed* are selected as seed points. The conditions for appending a pixel *p* with intensity value *g* into a region *S* is:

1. $|g - g(seed\,point)| < Dseed$,

2. *p* is 8-connected with a pixel belonging to the region *S*,

where *g(seed point)* is the intensity value of the corresponding seed point and *Dseed* is a constant. This is repeated until all pixels are assigned to some region. If the *size* of a region falls below a threshold *Tsize* the region is discarded as corresponding to noise rather than to a possible object. When this procedure concludes, a series of feature vectors is created containing information for every region.

D. Feature Vector Classification Component

The basis for the development of the feature vector classifier is fuzzy logic. The choice of a fuzzy logic classifier is preferred for its simplicity as well as for the direct way in which it incorporates the experience into the structure of the classifier. It allows for easy changes on the system's behavior by fine-tuning the rule base or the membership functions that fuzzify the input variables, thus, making the system capable of adjusting to a variety of targets without much difficulty.

The process of classifying feature vectors is shown in Figure 2. At first, the elements of each vector are assigned a membership function. The *mean intensity* feature is fuzzified to *Low*, *Mid* and *High*. Similarly, the size feature is divided to three linguistic variations *Small*, *Medium* and *Large*, respectively. Every one of them is described by a trapezoidal function such as:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, a \le x \le b \\ 1, \quad b \le x \le c \\ \frac{x-d}{c-d}, c \le x \le d \\ 0, \quad x \le a,\ x \ge d \end{cases} \qquad (4)$$

a, *b*, *c*, *d* are constants. Examples of membership functions used are shown in Figures 3 and 4. After completing the fuzzification process, the results are evaluated through a rule based inference engine. Rules determine the output of the system and give an indication regarding the presence or the absence of a target in the area. Rules are stated in the well known IF THEN form, as for example:

IF *Mean Intensity* is *High* AND *Size* is *Medium*
THEN *Target ID possibility* is *High*.

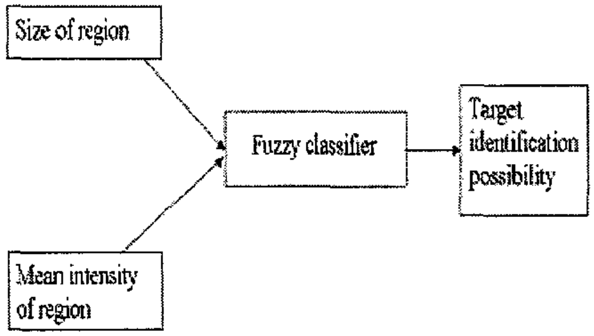


Fig.2: Block diagram of the feature vector classification subsystem.

The output variable is the *target identification possibility* and its membership functions are depicted in Figure 5. After examining the rules and applying the aggregation method (max), the defuzzification process yields a number in [0,1] that classifies each region into one out of three different classes. If the output possibility is lower than a threshold $T1out$ =0.5 then the region, whose feature vector produced that result, has LOW possibility of being the designated target. A MEDIUM possibility classification occurs when the output is between $T1out$ =0.5 and $T2out$ =0.7, while HIGH possibility is assigned to those regions that produce an output higher than $T2out$ =0.7.

E. *Alarm raising component*

The alarm-raising module of the machine vision system is the one responsible for the final decision regarding the presence or the absence of the object in an image. If this decision is left solely to the fuzzy classifier, some random events, such as reflections or noise, can trigger an alarm by inducing into the image bright regions that do not actually correspond to a thermal source. Exploiting the observation that such events are small in duration, this kind of false alarms can be avoided if a sort of duration threshold is introduced.

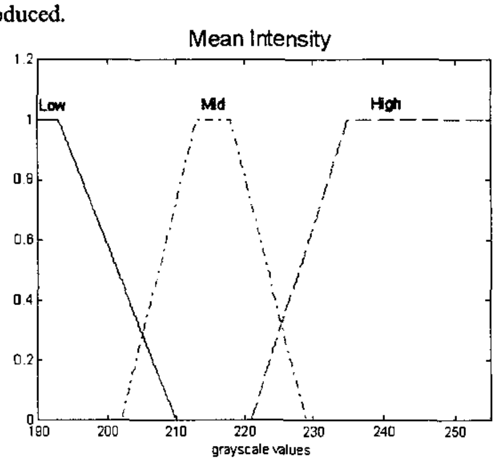


Fig.3: The Mean Intensity membership functions

A region must be classified as of high possibility over several frames before it could be considered as valid indication for the existence of the target.

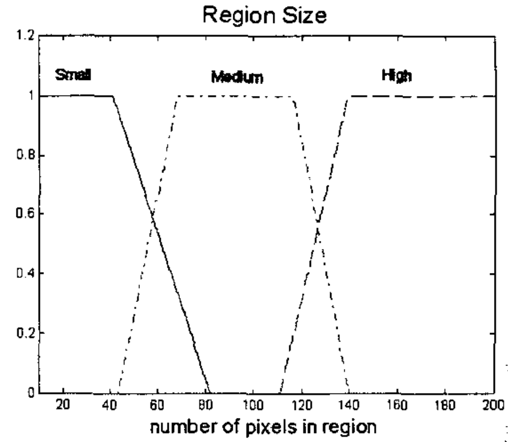


Fig.4: Membership functions of the Size of Region feature.

Then the alarm is raised and it is turned off only if the object is no longer detected. In this way the classifier's initial evaluation of a region is confirmed or rejected by its subsequent ones.

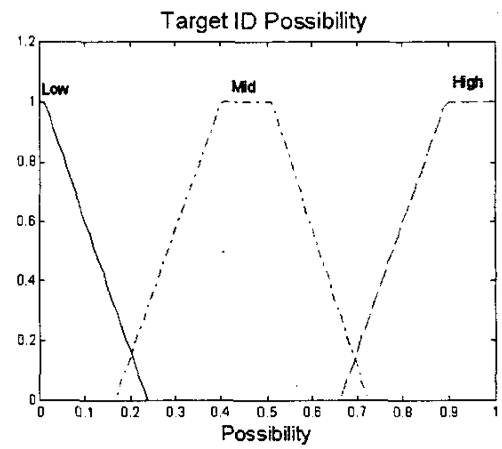


Fig.5: The membership functions for the output variable target identification possibility

The mechanism that implements this behavior uses a registry to keep track of all the regions that are given a high possibility of being the target and therefore are likely to raise the alarm. In this registry the position of every such region is stored as well as a variable indicating the *persistence* with which the classifier assigns a high ID possibility to that region. This variable is increased by a constant, e.g. 2, every time the region is given a high ID possibility and is decreased by 1 in any other case. If the value of *persistence* exceeds a certain threshold *Ton* the alarm is set. On the contrary, should it fall below *Toff* the alarm is turned off.

For the system to be able to accumulate evaluations for a certain region, it must also be able to identify and track it over subsequent frames. Based upon the assumption that a region shifts only by a small amount of pixels over two successive frames, the algorithm searches for it in the same area of the next frame. For a region to be tracked it must not shift more than a distance *Tvar* from its position in the previous frame.

The various thresholds used throughout the system can be viewed as the parameters that define its behavior and performance. For convenience they are summarized in Table 1.

F. Complexity

Dealing with two-dimensional images may be computationally intensive. However, in this case, the algorithms used, namely the noise reduction and region growing ones, did not exceed the order of $O(n^2)$ in complexity, with n being the dimension of a square image (nxn). Also, under the current implementation, the fuzzy classifier is found to have its own complexity depending mainly on the product of the number of features (n) and the number of rules in its inference engine (m). So, in general, it is of order $O(nxm)$.

Although one can argue that $O(n^2)$ algorithms do not scale nicely, the various tests the system has been subjected to, showed that, for images of resolution up to 320x240 pixels, real time or near real time performance can be achieved on average commodity hardware. This resolution is found on the most mid valued IR cameras available today and it is sufficient for most applications like the one described in the next Section.

TABLE I. PARAMETERS OF THE SYSTEM

Parameter	Meaning
Feature extraction component	
Tseed	Intensity threshold for the selection of seed points
Tsize	Region *size* threshold
Dseed	Intensity threshold for the similarity criterion satisfied by the pixels belonging to the same region
Classification component	
a_{ij} b_{ij} c_{ij} d_{ij} for i,j =1,2,3	Parameters that define the form of the trapezoidal membership functions of the fuzzy classifier
T1out, T2out	Thresholds separating the region classes in terms of target ID possibility
Alarm raising component	
Ton,	Threshold for the *persistence* variable above which the alarm is set on
Toff	Threshold for the *persistence* variable above which the alarm is set off
Tvar	Limit of a region's shift between two consecutive frames

III. CASE STUDY

In order to evaluate the system in a real scenario, a test case was set up. A small fire was set in a semi-rural area, having taken of course all the necessary precautions. The system described in the previous section was configured for the purposes of forest fire detection. In particular, the membership functions of the *mean intensity* and *size* features were altered to better describe the inquired object. The resulting functions are shown in Fig.6 and 7. All the other parameters of the fuzzy classifier were retained.

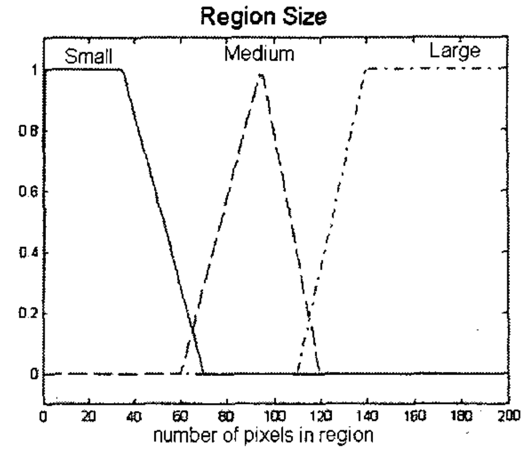


Fig.6: Membership functions of the Size of Region feature for the forest fire detection case.

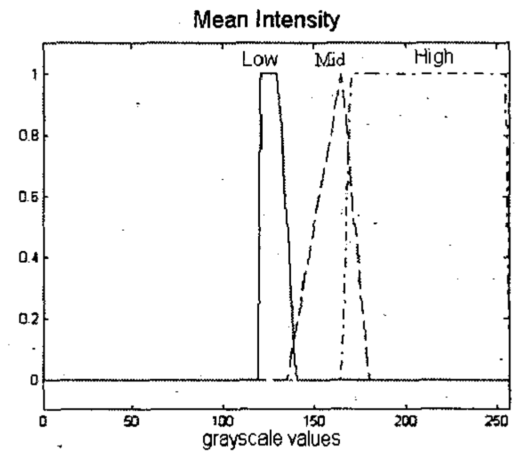


Fig.7: The Mean Intensity membership functions for the forest fire detection case

For visualization purposes the regions that are classified as of low possibility are painted green, while orange and red were chosen to indicate regions of medium and high possibility respectively. Also, when a certain region causes the alarm to be set, it is pinpointed by superimposing a crosshair over it.

The system's performance was evaluated both in the presence and absence of the fire. Results are presented in Fig.8 and Fig. 9. (For more results please visit http://www.dpem.tuc.gr/ISRL/ENPAGES/EN_PERSONNEL/EN_KONTITSIS/IR_RESULTS)

IV. AUTOMATIC PARAMETER SELECTION

In this section the attempt to automate the design process of the system is presented. The aim is to derive a system that functions similarly to the one designed for forest fire detection not by selecting all of its parameters manually. This implies that some of them will be defined using some automatic method. The parameters chosen to undergo automatic selection are those that define the form of the membership functions used to describe the linguistic variations of the *mean intensity* feature (see equation 4). Due to the plethora of possible solutions and the inadequate knowledge of the solution space, a genetic algorithm was employed.

A. Basic Elements of the Genetic Algorithm

The genetic algorithm (GA) implemented in this paper, like any other GA, includes the following [7]:

- Chromosomes consisted of the a_i, b_i, c_i, d_i for i=1,2,3 parameters were used (see equation 4). Recall that there are three membership functions for the *mean intensity* feature. Each one of the chromosomes represents a possible solution to problem of selecting the parameters.

Fig 8: Result of the system's response in the presence of a fire

- The fitness function, through which all chromosomes are evaluated, has three branches. If an image contains a fire and the system raises the alarm, then $fitness(x) = ke^{-0.05dist}$ where k is a constant and *dist* is the Euclidian distance of the region, which the system indicates as a thermal source from its actual position. If there is no fire present and the alarm isn't raised, then *fitness*=1. In any other case *fitness*=0.
- A selection operator that selects individuals for mating. Every individual is chosen as many times as the ratio of its fitness to the total fitness of the population.
- An operator responsible for the crossover of two mating chromosomes. It randomly chooses a locus and with a probability p_c=0.7 exchanges the gene subsequences of the two chromosomes.
- Finally a mutation operator is used to add to every gene of the offspring a random number in [-5, 5]. Mutation probability was p_m=0.001.

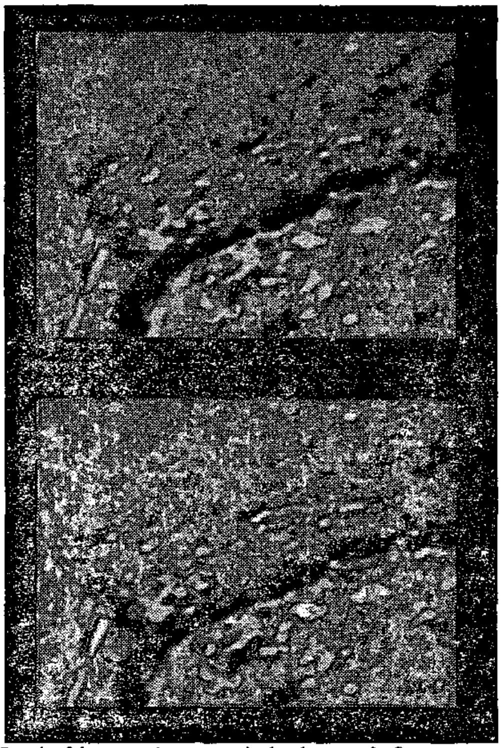


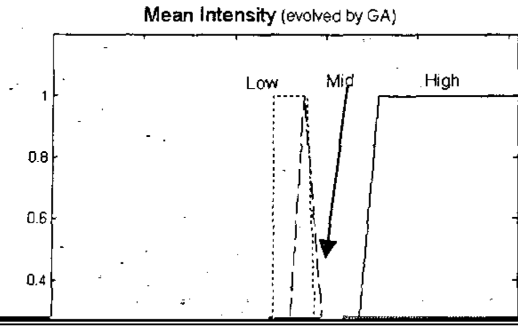
Fig 9: Result of the system's response in the absence of a fire

The GA was used to evolve a population of 100 chromosomes for 50 generations. After several runs the fittest individual yielded the membership functions depicted in Fig. 10.

It can be observed that the *Low* and *Mid* membership functions almost share the same support set. This is due to the GA's tendency to favor those individuals that can identify a region corresponding to a fire, without paying any attention to the characterization of the rest of the regions.

B. Results

A training set of about 200 images was used during the runs of the GA. Among them there were some that had a fire present as well as some without one. The system designed by the fittest individual managed to respond correctly to about 90% of the cases. Afterwards the system was prompted to evaluate a different sequence of images with more or less the same success, although it didn't entirely avoid some false alarms. Fig. 11 and 12 show exactly that. In comparison with the system that had manually selected parameters, it shows about the same performance being worse only in terms of false alarm occurrence.



disturbances such as random reflections, without sacrificing its detection sensitivity, as it would be expected.

Despite all these, the system has no way to adapt to new data. Although the automatic selection of some parameters presented in Section IV, may qualify as some sort of adaptation, the fact that it is carried out offline deprives the system the ability to constantly modify its structure. Online adaptation as well as the automatic selection of a wider range of parameters should be a vital part of any future developments that could be considered.

[5] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison Wesley, 1992.

[6] S. Pal, S. Mitra, *Neuro-fuzzy pattern recognition: methods in soft computing*, Wiley, New York, 1999.

[7] Mitchel Melanie, "An introduction to genetic algorithms", MIT Press, 1996.