

Infrared Target-Flare Discrimination using a ZISC Hardware Neural Network

G. Labonté , W.C. Deck

Department of Mathematics and Computer Science, Royal Military College of Canada

Kingston, Ontario, Canada K7K 7B4,

Email: labonte-g@rmc.ca, deck.wc@forces.gc.ca

Telephone of G.L.: 613-384-3052

ABSTRACT

The last generation of infrared imaging aircraft seekers and trackers uses pattern recognition algorithms to find and keep a lock on an aircraft in the presence of decoy flares. These algorithms identify targets, based on the features of the various objects in the missile's field of view. Because modern both aircrafts and missiles fly faster than sound, speed of operation of the target identifier is critical. In this article, we propose a target recognition system that respects this time constraint. It is based on an artificial neural network implemented in hardware, as a set of parallel processors on a commercially available silicon chip called a ZISC, for Zero Instruction Set Computer. This chip would be integrated in the infrared missile seeker and tracker. We describe the characteristics of the images that the image processing module of this seeker and tracker extracts from the infrared video frames and show how to construct from these translation and rotation invariant features that can be used as input to the neural network. We determine the individual discriminating power of these features by constructing their histograms, which allows us to eliminate some as not being useful for our purpose. Finally, by testing our system on real data, we show that it has a 90 % success rate in aircraft-flare identification, and a processing time that during this time, the aircrafts and missiles will have traveled only a few millimetres. Most of the images on which the neural network makes its mistakes are seen to be hard to recognize even by a human expert.

Keywords: *Infrared tracker, target identification, artificial neural network, ZISC.*

List of abbreviations:

ZISC: Zero Instruction Set Computer;

DRDC: Defence Research and Development Canada;

IIR: Imaging Infrared;

DSP: Digital Signal Processing;

RCE: Reduced Coulomb Energy (neural network);

ZISC036: ZISC with 36 neurons;

ZISC78: ZISC with 78 neurons;

FPGA: Field Programmable Gate Array;

PCI: Peripheral Component Interconnect;

RBF: Radial Basis Function (neural network);

1. INTRODUCTION

Automatic Target Recognition is a major field of research within the larger domain of Pattern Recognition. It is itself divided in sub-fields that correspond to the type of sensors used, as sonar, radar, infrared, video imagers, and to the types of targets to identify. Unfortunately, as pointed out in Roger, Colombi, Martin, Gainey, Fielding, Burns, Ruck, Kabrisky and Oxley [58], a general-purpose automatic target recognition system does not exist. This fact is actually true also for the whole domain of Pattern Recognition in which there exist still a wider range of different techniques. Indeed, Duda, Hart and Stork [19] (in Chapter 9) also explain that there is no answer to the question as to which classifier is the "best". This fact is often stated as a "No Free Lunch Theorem" that says that there is no context-independent classification method that should always be favoured over the others. In principle, any algorithm can be considered a candidate while in practice, some methods are better at some specific types of problems than others. We can't really know in advance how good a particular method will be until it is tested on typical data that correspond to the practical application considered.

We note that when studying a particular classifier, it is the normal practice to compare its performance to that of other ones used in the same domain. However,

when it comes to military systems that are presently used or under development, strict secrecy surrounds the methods that are most successful, with which such a comparison could be made, because these are precisely those used in deployed systems. Therefore, in studies such as the present one, we are left with demonstrating the potential of a technique that we suspect to be very promising. As said in Roger et al. [58], studies as this one can only be considered as a proof of concept; its true validation can only be done when the actual system is fielded and proven "under-fire". In this article, we will only report the results of tests that we have conducted with a neural network implemented in hardware in order to determine whether it could compute fast enough and with enough precision to be a possible candidate for incorporation in an infrared seeker and tracker for an anti-aircraft missile.

1.1 On Infrared Missile Seekers

According to Air force Link [2], a web site of the Office of the US Secretary of Air Force, the first infrared missiles was deployed in 1953; it is the AIM-9A that evolved into the still used Sidewinder missile. Initial infrared guidance systems simply made the missiles home in on aircraft engine exhausts. They were only effective at close range, could not engage targets close to the ground, and did not have night-time or head-on attack capability. The heat-seeking missile provided a major advantage, called "fire-and-forget", in that the pilot could launch it, then leave the area or take evasive action while the missile guided itself to the target. IR-guided missiles are difficult to detect because they do not emit any signal as would radar homing missiles for examples.

The advent of the IR missile set off a search for techniques to defeat the “optical” seeker system in these weapons. As recounted in Titterton [69], many of the active countermeasure projects were initiated in the 1960s. Their aim was to decoy the relatively simple, but effective, IR homing technology. On-board systems, known as jammers, as well as various pyrotechnic devices known as flares, were developed to confuse the target-tracking system in the missile’s seeker. Other measures were also taken such as extending the jet pipes in fighter aircraft to mitigate the effects of an interception by a heat-seeking missile. This was the beginning of a classic “cat and mouse” activity between the measures and the countermeasure, which continues to this day.

Goddard [24] and Koch [40] present the state of the art in present day countermeasures, the extent of their information being of course limited to what is available from unclassified sources. Goddard points out that initially flares were almost 100% effective against the first generation of missiles. Upon deploying flares, the aircraft would pull away at a sharp angle from the flare, toward which the missile would be attracted, and reduce its engine power in an attempt to cool its thermal signature. The missile's seeker was then confused by this change in temperature and flurry of new signatures, and thus made the missile follow the flare instead of the aircraft.

In response to the introduction of flares, the missile manufacturers had to imagine some counter counter-measures (CCMs), which were improvements in the infrared seekers. One of them is a rise rate trigger that senses the rapid rise of radiated energy of the flare and compares it with an acceptable rise rate for an engine change of power. If it exceeds that rate, the missile ignores the scene for

some time, hopefully until the flare is outside the field of view of its seeker while the aircraft is still there. Another technique takes advantage of the fact that flares generally deploy to the rear and move downwards with respect to the aircraft, due to the drag that decreases their velocity and the force of gravity. Thus, a reject sector can be created in the field of view of the missile that reduces the interest of IR sources in this region about the aircraft. The seeker can also take into account the rate of separation between the flare and the target aircraft. Finally, the seeker can use a two-colour detector to spectrally discriminate between the decoy and the aircraft exhaust plume, by comparing their luminosity at two different wave lengths simultaneously. A typical flare burns at 2000 C^0 while an aircraft engine is in the $600\text{-}800\text{ C}^0$ range; this allows the seeker to recognize its aircraft target. According to Titterton [69], the effectiveness of IR missiles was remarkable; they had been responsible for the majority of aircraft losses since their introduction into service during the 1960s, until the 1990s. Some statistics suggest that heat-seeking missiles have been responsible for more than 80% of all combat aircraft losses over the last 40 years according to Herskovitz [29].

In reaction to these developments, the flare manufacturers took measures to improve the decoy's success against the missile. They devised a technique to adjust the burn profile of the flare energy over the time period it is burning. Thus, for example, the intensity of the flare could be made to rise very fast with a short burn period when it was destined to a fast jet, because its flares separate quickly from the aircraft due to its speed. Aerodynamic flares have also been developed to counter the trajectory sensing ability and separation rate trigger of some missiles. Fast aircrafts, such as the F-18, can use flares that are powered by thrusters that make them move forward, out of the way of the aircraft. Dynamic

flares can also be made to move with a speed that mimics that of the aircraft. Towed flares exist that can be effective against rise rate discrimination. Because their motion is the same as that of the aircraft, kinematical discrimination will not help identify them. Koch [40] and Titterton [69] mention IR missile defeat mechanisms other than flares that are already used and some that could eventually be developed.

This sophistication of flares makes it clear that it has also become more and more challenging to produce seekers that can effectively discriminate between the aircrafts and the flares they deploy, that is to devise IR counter-countermeasures (IRCCM). Titterton [69] describes how the first optical seekers used a single infrared detector to sense the position of its target. These had a spinning reticle or mechanical modulator, in the focal plane of the optical system. This reticle consisted of a series of opaque and transparent segments that created the modulation, a simple one of which is shown in Figure 1. The reticle created a series of time-referenced pulses that were translated into the position of the designated target. It also provided the important function of spatial filtering that eliminates extended targets, such as clouds. Han, Hong, Jahng, Seo and Choi [28], Goldberg [25] and Hong, Jahng, Doo and Choi [31] respectively describe the principle of operation of the fixed reticle scan seeker, the conical scan seeker and the concentric annular ring scanner. That technology was followed by the more sophisticated one of the rosette scanning seeker, which is described in details in Jahng, Hong, Han, Choi [35], Dinaki, Shokouhi, Soltanizadeh [17]. The latter two references mention that in 2008, the rosette scanning seekers were still actually used in many missiles; the popular Stinger by Raytheon [68] is one of them. This system also uses a single detector that is provided information about

the scene in front of the missile through a small window that moves along the path of a rosette pattern, as shown in Figure 1. Many studies have been conducted and are still being conducted on algorithms that would effectively identify aircrafts and flares with that system, some of which are Jahng, Hong, Han, Choi [35], Jahng, Hong [33], Jahng, Hong, Seo and Choi [36], Doo, Oh, Jahng, Hong, Choi and Seo [18], Dinaki, Shokouhi, Soltanizadeh [17], Soltanizadeh and Shokouhi [63].

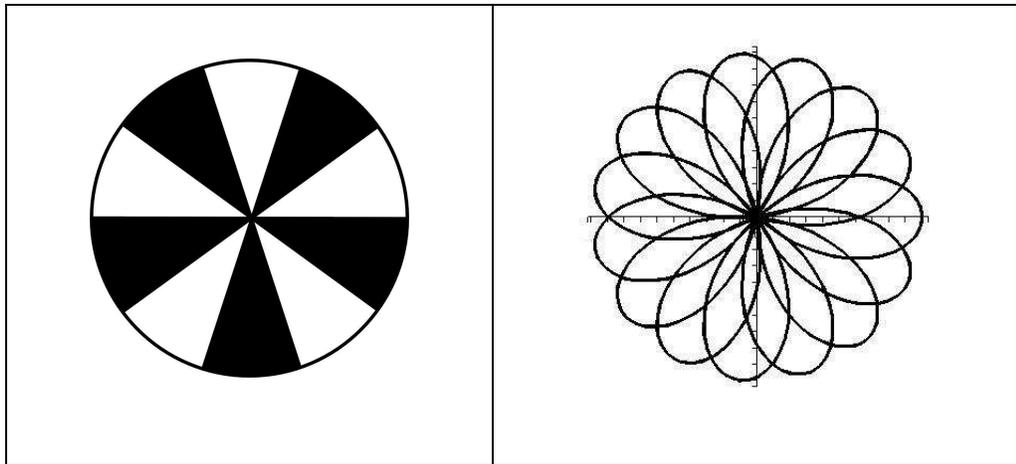


Figure 1: On the left-hand-side: reticule mask and on the right-hand-side: rosette scan used initially in infrared seekers.

Those seekers that used a single IR detector are inherently limited in speed by the fact that they require a mechanical device to produce the scan. Thus, for example, Soltanizadeh and Shokouhi [63] mention that 10 ms are required for the complete rosette to be scanned. Present day progresses in infrared, electro-optic and computing technologies make it possible to use target seekers and trackers with a complete focal plane array of IR detectors that produces instantaneously an image of the whole field of view. The signals produced by these detectors are interpreted as grey-scale values for corresponding pixels in an image. This complete image of the field of view made it is possible to use more complex

features of the objects seen, in more sophisticated algorithms, and thus have a higher probability of success in differentiating between the countermeasures and the platform they protect. Many missiles exist today with that technology, such as the AIM-9X that is the successor of the Sidewinder missile [2] and that presently arms the F-16 and F-18 fighters. Sagem Défense Sécurité, Safran Group [51] sells the MICA infrared seeker for air-to-air missile that, they claim, offers a high protection against infra-red countermeasures through the use of dual band infra-red imaging, highly sophisticated image and signal algorithms. Table 1, taken from Koch [40], gives an overview of the evolution of IR seeker technology.

Evolution of IR seeker technology				
Gen.	Signal Processing	Detector Type	Missile Example	Ref.
1	Reticle, Spin Scan, Amplitude Modulation	Single Color	SA-7, SA-9, SA-13, AIM-9B	[26]
2	Reticle, Conical Scan, Frequency Modulation.	Single Color	SA-14, SA-16, AA-8, AIM-9L/M	[26]
3	Rosette Scan, Conical Scan	UV / IR	RIM 92B/C, SA-18	[26]
	Cross Array, Conical Scan	Single & Dual Color	AA-11	[26, 34]
	Concentric Annular Ring, Conical Scan	Single Color	AA-10	[26, 31]
4	Focal Plane Array	Single Color	IRIS-T	[50]
5	Focal Plane Array	Hyperspectral	None yet	

Table 1: Evolution of IR seeker technology. The first column is the generation number, the second column is the method of detection of the signal, the third column is the type of IR detector used, and the fourth column gives examples of missiles using the corresponding technology. The last column gives a reference to that technology.

The availability of a whole image of the scene meant that all the usual techniques of artificial vision could be adapted to this problem of target recognition. Many studies existed already in the domain of infrared target recognition for land vehicles in background clutter. Roger, Colombi, Martin, Gainey, Fielding, Burns, Ruck, Kabrisky and Oxley [58] reviewed concepts associated with the processing of military data to find and recognize targets. They recognized that the target identification process starts with pre-processing or filtering to suppress noise and enhance spatial discontinuities or edges. This is followed by segmentation, a process that isolates the blobs that correspond to the objects present in the scene. Once this has been done, features are extracted from the blobs, care being taken to use the smallest possible number of features, which have to be selected for their strong discriminating power. A classifier is then trained with some data and then tested on new data it had never seen before. Singstock [62] examined different methods of generating features to use for IR target identification: standard features as intensity, shape and Fourier coefficients, Karhunen-Loève transform and discrete cosine transform features. He mentions, as particularly useful intensity and spatial features such as the ratio of object length to its width, the standard deviation of pixel values, the maximum intensity of the object pixels, the complexity, which is the ratio of the border pixels to the total number of pixels, and various other intensity and shape features. Many of these features are actually the same as those we use in the present study. Singstock [62] compared the efficiency of different sets of features used as input vectors for a three-layer back-

propagation neural network. He found that the recognition success rates were comparable with the different types used when the individual features making up the input vectors were carefully selected for their discriminating power. Hung, Webb, Elliott and Chandler [32] point out that combining consecutive frames will improve the quality of the images and also provide dynamic characteristics of the objects in the field of view, which can have a strong discriminating power. In these studies, the features were extracted from blobs in the image that had been segmented out as regions of interest. Other approaches also simply use features extracted from image patches of a given size, such as in Chan, Der and Nasrabadi [9], Khan and Alam [37], Bhuiyana, Alam and Alkanhal [3], Khan and Alam [38], Singh, Pettersson, Karlholm, Berndt and Brunnström [67].

There has been much consideration given to the problem of enhancing the image of the target and filtering out the clutter. Khan and Alam [37] mention statistical modeling, Fuzzy logic, Fourier analysis, Gabor filtering, and wavelet transform of image patches as possible approaches. As pointed out in Feng, Shang-qian, Dabao and Wei [21], an advantage of using wavelet moments as features is that such features are invariant under translation, scaling and orientation transformations. Khan and Alam [37] used the wavelet transform of patches of an image to provide the feature vectors. They train a probabilistic neural network [66] with patches data extracted from the first few images of a video sequence and showed that this neural network can thereafter track the target in the rest of the sequence. Bhuiyana, Alam and Alkanhal [3] considered the same tracking problem and solved it with the help of a special correlation filter that points out the designed target in subsequent video frames. Khan and Alam [38] combined the two methods described above to solve the same target tracking problem. Singh,

Pettersson, Karlholm, Berndt and Brunnström [67] used image correlation and Der and Nasrabadi [9] used a multilayer perceptron neural network to discriminate between the target and the background clutter. Their feature vectors were extracted from patches of the image and their dimensionality was reduced by Principal Component Analysis.

Different features than those mentioned above have also been considered to identify the targets. For example, Yu and Azimi-Sadjadi [73] proposed to use the temporal tracks of moving objects to recognize the targets. These tracks are constructed by computing a correlation between successive frames of a video sequence. This method allows for the detection of targets even in very cluttered environments because it is based on recognizing the motion of the target instead of its intrinsic attributes. They used a neural network to classify the ground vehicles seen, based on their acceleration and the curvature changes of their trajectory. Nair and Aggarwal [54], Zhao, Shah, Choi, Nair and Aggarwal [75], Nair and Aggarwal [55] proposed an object recognition method based on the object's individual parts. They devised a hierarchical recognition strategy that uses salient object parts as cues for classification and recognition. They point out the advantage of this approach in situations where the objects are partially occluded. Zhang, Zhong, Yan and Wang [74] considered the last phase of an anti-aircraft missile homing on its target, during which the size of the target image increases rapidly, which makes it harder to follow with a standard correlation approach. They suggested using the corners of the target image as characterizing features in this situation.

In most of the above mentioned studies, the target identification methods were tested on real video sequences taken from the database of Army Missile Command (AMCOM), the objects represented being ground vehicles such as tanks and trucks. Furthermore, they were not really concerned with the real-time aspect of the problem that becomes so critical with fast moving objects as aircrafts and missiles, as we consider in our study.

Image correlation and artificial neural networks clearly stand out, in the published literature, as the two main methods used for target identification and tracking. However, because the focal plane array imaging seekers represent the latest developments in IR missile seeker technology, no information is publicly available about the particular image processing and pattern recognition algorithms they use.

The present study can be seen as the continuation of that in Cayouette's MSc thesis [7] and Cayouette, Labonté and Morin [8] that considered the same aircraft-flare discrimination problem we deal with here. This former study used a probabilistic neural network [66] to classify the patterns, seen in a single infrared video frame, as corresponding to aircrafts or flares. They reported success rates in the 90–95% range. Labonté and Morin [42] then used temporal features of the objects, extracted from a few successive video frames, to discriminate between the aircrafts and the flares. They report a success rate of 92–100%. However, these studies used a probabilistic neural network that was realized in software on a sequential computer. Thus, although they established the discriminating power of the neural network, they did not yield a real time system. In the present study we show that a neural network realized in hardware can solve the same problem in

real time. This neural network is a Reduced Coulomb Energy (RCE) neural network, which is a particular type of Radial Basis Functions (RBF) neural network, that was devised by Leon Cooper et al. [56] for pattern identification. As can be seen in Bishop [4] and Duda, Hart and Stork [19], this neural network stands out among the various artificial neural network models that serve for pattern recognition by its relative simplicity. It is this feature that lead to its being very early realized in hardware as the ZISC, which stands for "Zero Instruction Set Computer" [61].

1.2 IR Seeker Test Bed

Morin and Lessard [52] describe the simulator of a focal plane array infrared imaging seeker and tracker that has been developed at the Defense Research and Development Canada establishment at Valcartier. Its purpose is to test and improve the individual components of operational seekers and trackers. It is a good example of the different modules that constitute such systems and their mutual interactions. This seeker and tracker has an infrared video camera that operates in the 3 to 5 μm waveband. It is mounted on motorized gimbals that are controlled in real time by a series of fast processors. This camera produces images on a 256×256 matrix of detectors in its focal plane, each one of which corresponds to a pixel in the digital image. Each of these video frames is then sent to a series of high speed computing modules to be processed in real time in order to identify the targets in the seeker and tracker's field of view. When a target of interest is recognized and selected, tracking commands are sent to the control system that moves the gimbaled platform so as to keep the camera locked on that particular target. Figure 2 illustrates its structure.

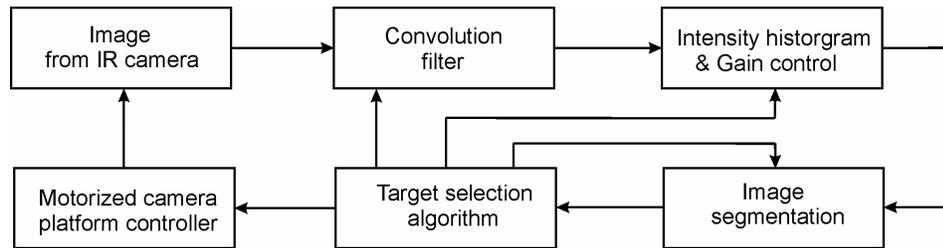


Figure 2: Basic configuration of the infrared seeker and tracker.

1) The block in the upper left-hand corner indicates that the image from the camera is sent to a convolution filter that aims at correcting for the non-uniformity of infrared detectors' sensitivity. This filtering serves to facilitate the image segmentation and subsequent target detection by reducing the noise and attenuating the background clutter. The relative efficiency of various filters has been examined in Morin [53].

2) The intensity histogram of the filtered image is then computed and used to adjust automatically the gain for the intensity level of the images.

3) The image is then sent to an image processing module that segments the objects present in the image into blobs. A simple algorithm that works well here and that we have used for the present study involves simply defining them as the continuous domains in the image in which the intensities of the pixels exceed a certain threshold. Intensity and spatial characteristics of the blobs, which we describe below, are then extracted.

4) A target selection algorithm uses these features to identify each one of the blobs and determine which one is the aircraft to track.

5) The coordinates of the centroid of the blob to track are fed to the tracker controller that commands the platform servomechanism. This turns the platform so that the camera points toward this particular blob.

The arrows in Figure 2 indicate the flow of information in the seeker tracker. As can be seen, there are many feedbacks between the different modules. One of their function is to re-adjust various thresholds and filters so as to optimize the quality of the target features extracted from the images. Figure 3 illustrates the image pre-processing stage and segmentation into blobs.

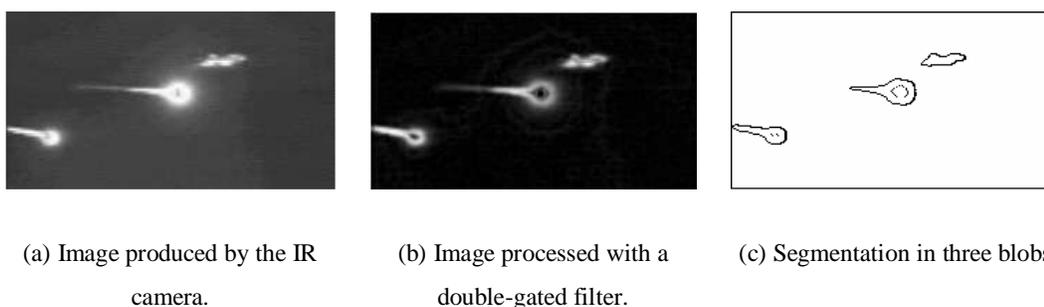


Figure 3: Pre-processing of the infrared image and its segmentation into separate blobs.

1.3 Problem Statement and Proposed Solution

The most difficult and critical process in the imaging seeker and tracker described above is the fourth one, in which the nature of each one of the blobs has to be recognized from a set of their features. This is the process that we propose to realize with the ZISC hardware artificial neural network.

In general, the infrared images seen by the camera can contain part of the ground, some clouds and the sun, besides aircrafts, flares and the bare sky. Although we did not test it explicitly, the tests we did with aircrafts and flares showed that the trained neural network would easily dismiss the sun as not being an aircraft. Indeed, none of the flares that deployed in spherical patterns were ever mistaken for an aircraft, mainly because the blob aspect ratio is used as a discriminating feature. Because the present study is a proof of concept, we used only images in which the clutter had been eliminated and concentrated on the problem of discriminating between the aircrafts and the flares. We considered that demonstrating the efficiency and speed of the ZISC neural network at this task constituted a first step. If successful, it would then be worthwhile to endeavor to generalize our system to include other objects. It has to be noted also that, in most situations, when the missile is getting close to its aircraft target, there will only be the aircraft and its flares in its rather small field of view. In that sense, the aircraft and flare discrimination problem is also the most crucial one. In the present study, we also used only static features of the images, even though dynamic features also have strong discriminating power as shown, as was demonstrated in Labonté and Morin [42]. We shall consider dynamic characteristics in a subsequent study.

Of course, when an aircraft deploys some flares, there is more than one blob in the field of view of the missile: the one for the aircraft and one or more for the flares. Thus, there will be many blobs to analyze in one video frame, while our system analyses only one of them at the time. This situation can easily be dealt with by having many identically trained ZISC neural networks mounted in parallel in the target seeker and tracker, which would simultaneously each identify one of these

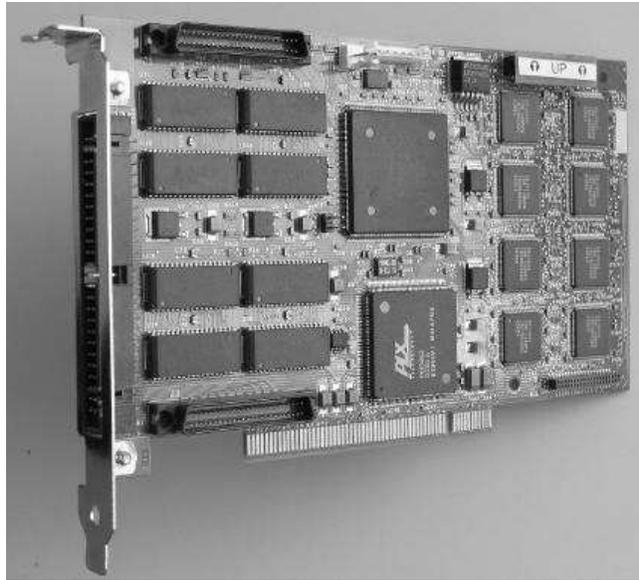
blobs. At the present stage of development, we are not concerned with the particulars of the physical connections and inter-module communication protocols. These will be considered only once the merit of our ZISC target recognition module has been clearly demonstrated by itself.

The data we used for this study were extracted from video sequences provided by the Defence Research and Development Canada (DRDC) establishment at Valcartier, QC. These video sequences were taken with an IR video camera that was on the ground and showed various aircrafts deploying different types of flares. The aircrafts moved in different directions with respect to the observer, and are seen at different distances. In our tests, we used 1480 blobs that corresponded to an equal number of single aircrafts and single flares. (The feature vectors for these blobs can be obtained by email from the first author of this article.) Most of the images are of the quality of those shown in Figures 2 and 13. Although, in some of these images, as in Figure 3, it is relatively easy for human observers to recognize the aircraft, in some others it is not, as can be seen in Figure 14. The data we had at our disposal were therefore rather challenging for an automatic target identification system.

1.4 The ZISC and the Cognimem

According to the ZISC Technology Reference Guide [61] (page 2), the ZISC is a data processing chip that has been developed jointly by Guy Paillet, founder of Silicon Recognition in California and NorLiTech of Japan, and IBM. Its development took place at the IBM Essonnes laboratory near Paris, France, and was first introduced to the public market in 1994. This chip is a complete

realization in hardware of the Reduced Coulomb Energy neural network [56]. In Section 4, we shall describe the version of the algorithm that is implemented in the chip. The first ZISC chip, the ZISC036, contained 36 neurons and the second version, the ZISC78, 78 neurons. One of the founding companies, Silicon Recognition, has mounted the chips on PCI boards for personal computers. Our research used one such board, the EZB 624 PCI, which has eight ZISC78 chips, giving a total capacity of 624 neurons. This board also has its own memory and a Field Programmable Gate Array (FPGA). It is shown in Figure 4a. The portion of the Silicon Recognition company that manufactured and supported the EZB 624 PCI has been sold to an Italian company called EOS. It was from EOS that we purchased the board for this research. In 2007, Guy Paillet presented a new chip with 1024 neurons called a Cognimem. This chip is presently produced by Recognetics Ltd in Suzhou, China. It is shown in Figure 4b and its properties are described in the Cognimem information sheet [11]. The ZISC and the Cognimem chips are designed to be cascaded to create neural networks with a virtually limitless number of neurons.



3a



3b

Figure 4a: EZB 624 PCI board. The eight ZISC78 chips that make the 624 neurons neural network are visible to the right of the board. **Figure 4b:** The Cognimem chip with 1024 neurons.

Eide et al [20] (page 3) mention that the ZISC “has been designed for cost-effective recognition and classification in real-time.” Its effectiveness comes from its being a computing device dedicated to a single particular task and its ability to process data in parallel. It has already been used successfully in many practical applications, such as the following ones.

- The application for which IBM developed the ZISC initially was the automatic visual inspection of the VLSI they produced in their Essones plant. The goal of this application was to inspect vias, which are standard or test dedicated Input-Output pads on the VLSIs, for probe damage during wafer tests. Each via was analyzed and classified as having good impact, bad impact or absence of impact. This application provided the first demonstration of the efficiency of the ZISC in a manufacturing environment. It is described in de Trémiolles, Tannhof, Plougouven,

Demarighn and Madani [15]. The efficiency of the ZISC in real-time vision or pattern recognition systems was thereafter recognized and many other practical applications followed. .

- Lindblad, Lindsey, Minerskjöld, Skhniaidze, Székely and Eide [43] used the ZISC to look for Higg's boson events amongst the very large number of traces made by the elementary particles created in high energy particle accelerators. Lindsey, Lindblad, Sekhniaidze, Székely [45], and Minerskjöld [45] also report tests of the ZISC used in high energy physics tasks.
- Madani, Mercier, Chebira and Duchesne [49] used the ZISC to devise a new approach to control that implements a parallel real time intelligent adaptive controller. They presented experimental results that validate their concept.
- Chebira, Madani and Mercier [10] devised a data driven method, they called DTS (Divide to Simplify), that builds dynamically a Multi-Neural Network Architecture of ZISCs. The Multi-Neural Network architecture they propose solves a complex problem by splitting it into several easier problems. Tests they reported show that, in the resulting neural network, learning is performed in few milliseconds and a very good rate of classification is obtained.
- Lindsey, Lindblad and Eide [44] devised a ZISC based star-tracker to identify star constellations that are used to determine with a very high

precision the attitude of spacecrafts. The ZISC then compares feature vectors derived from histograms of distances to multiple stars around the unknown star. This method is seen to be robust with respect to position noise and requires a smaller database to train than conventional methods, especially for small fields of view.

- David, Williams, de Tremiolles and Tannhof [12] presented a ZISC based solution to the problem of noise reduction and image enhancement and demonstrate its efficiency. The goal of their application was the restoration of old movies (noise reduction, focus correction, etc.), the improvement of digital television images, and the treatment of images which require adaptive processing (medical images, spatial images, special effects, etc.).
- Madani, de Tremiolles, Williams and Tannhof [48] used the ZISC to solve the difficult problem of prediction and modelling of complex non-linear industrial systems. In particular, they deal with the production yield prediction in VLSI manufacturing.
- Gliever and Slaughter [23] described an application based on the ZISC that discriminates between images of cotton crop and weeds, in order to determine where herbicides should be sprayed by an automatic sprayer. They reported better than 99% correct identification.
- Yang and Paindavoine [72] developed a real time vision system based on the ZISC that localizes faces in video sequences and verifies their identity.

They demonstrated the robustness of their system by testing it on real video sequences and comparing its performance to that of other systems.

- Gaughran [22] presented a novel approach employing the ZISC to implement binary neighbourhood operations in image processing such as dilation, erosion, opening and closing. Despite increases in serial processor speed, such operations have remained computationally intensive, but parallel devices as the ZISC have significantly reduced the computation overheads.
- Multispectral and imaging systems on a spacecraft can produce more data than can be analyzed by humans on Earth. Cai, Hu, Siegel, Gollapalli, Venugopal and Bardak [6] considered the problem of reducing this data, in particular lidar profile data, by devising a feature indexing system to perform pattern recognition and data compression onboard. They implemented a prototype of the onboard computer with ZISC chips and FPGA (Field Programmable Gate Array) so that it takes advantage of intrinsic parallel computing and reconfigurability. They reached a high data compression rate of 99.17% with reasonable error range. They showed that their method significantly outperformed the wavelet compression technique.
- Zhang, Ghobakhlou and Nikola Kasabov [74] described a person identification system based on statistical methods and the ZISC that recognizes features extracted from faces. They demonstrated the

efficiency of their system and showed that it is able to evolve and improve its performances.

- Holton [30] reported that the ZISC, and its newer version, the Cognimem, have been incorporated in intelligent cameras that perform automatic image classification. Practical applications of these have been made to inspect automotive cartridge filters at Norcon Systems (Lombard, IL) and to inspect the disk platters in hard disk drives at IBM to determine whether they are missing or ill inserted.
- Budnyk, Chebira and Madani [5] used the ZISC in a new approach to estimate task complexity that involves building a self-organizing neural tree structure.
- Kim and Slaughter [39] described a precise displacement measurement system that uses a non-contact image-based optical sensor and a ZISC to control the application of material in precision agriculture. Field tests, with the system mounted on a tractor-drawn toolbar demonstrated that a much better precision was attained with this system than with the usual ground-wheel driven encoder.
- Deck [13] and Deck and Labonté [14] used the ZISC as a parallel computer to calculate correlation coefficients between an input pattern and patterns stored in its neurons. They explored the possibility of using the ZISC in a target tracking system by devising algorithms that take advantage of the ZISC's parallelism and testing them on real video

sequences. Their experiments indicated that the ZISC does improve appreciably the computing time compared to a sequential version of the algorithm.

- Finally, the ZISC can be seen in the context of a review of commercially available neural network hardware in Dias, Antunesa and Motab [16], in Hammerstrom [27], in Madani [46] and in Smith [65].

Based on considerable experience accumulated working with the ZISC, Madani, Detremiolles and Tannhof [47] provide precious suggestions concerning using it efficiently. They present an analysis of the main parameters that influence its image processing power. They discuss more particularly the learning and the working of the ZISC as a massively parallel device.

The consideration of the wide range of domains covered by these applications provided a definite motivation for us to test its ability to perform the task of IR aircraft and flare discrimination.

2. TARGET CHARACTERISTICS

The characteristics of the blobs identified by the DSP board of our target seeker and tracker can be grouped in intensity and in spatial features.

2.1 Intensity features

Let $Z_i : i = 1$ to n , represent the intensities associated with the n pixels of the blob considered. Since the image is gray-scale, each Z_i is an integer in the range $[0,$

255]. The intensity characteristics are as shown in Table 2. The moments of the intensity distribution measure the apparent texture of the target.

Intensity characteristics produced by the DSP module	
Formula	Designation
$Z_{\max} = \text{Max}_{i=1..n} Z_i$	The maximum intensity.
$\bar{Z} = \frac{1}{n} \sum_{i=1}^n Z_i$	The average intensity.
$\mu_Z^2 = \frac{1}{n} \sum_{i=1}^n (Z_i - \bar{Z})^2$	The variance of the intensity distribution.
$\mu_Z^3 = \frac{1}{n} \sum_{i=1}^n (Z_i - \bar{Z})^3$	The third moment of the intensity distribution.

2.2 Shape features

In order to relieve the computation load, the image is then binarized such that the pixels inside and outside the blobs have respective intensity one and zero. We let (x_i, y_i) for $i = 1$ to n be the coordinates of the pixels inside a given blob, and (x_i^b, y_i^b) for $i = 1$ to m be the coordinates of the pixels on the boundary of this same blob. The shape characteristics are then defined as follows.

Shape characteristics produced by the DSP module	
Formula	Designation
$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$	The coordinates of the centroid.
$I_{\max} = \frac{1}{2}(I_{xx} + I_{yy}) + \frac{1}{2}\sqrt{(I_{xx} - I_{yy})^2 + 4I_{xy}^2}$ with $I_{xx} = \sum_{i=1}^n (y_i - \bar{y})^2$	The large principal moment of inertia.

$I_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$ $I_{yy} = \sum_{i=1}^n (x_i - \bar{x})^2$	
$I_{\min} = \frac{1}{2}(I_{xx} + I_{yy}) - \frac{1}{2}\sqrt{(I_{xx} - I_{yy})^2 + 4I_{xy}^2}$	The small principal moment of inertia.
The angle θ , such that if the blob is rotated by $-\theta$ around its centroid, its moment of inertia I_{xx} will have the smallest possible value.	The angle of orientation θ .
$e = \sqrt{1 - \left(\frac{b}{a}\right)^2} = \sqrt{1 - \frac{I_{\min}}{I_{\max}}}$	The eccentricity.
$P = m$, the number of pixels on the perimeter.	The perimeter.
$A = n$, the number of pixels in the blob.	The area.
$R = \frac{P^2}{4\pi A}$	The roundness.
$D_{\max} = \text{Max}_{i=1..n} D_i$ <p>where $D_i = \sqrt{(x_i^b - \bar{x})^2 + (y_i^b - \bar{y})^2}$</p>	The maximum radial distance.
$D_{\min} = \text{Max}_{i=1..n} D_i$	The minimum radial distance.
$\bar{D} = \frac{1}{m} \sum_{i=1}^m D_i$	The average radial distance.
$\mu_D^2 = \frac{1}{m} \sum_{i=1}^m (D_i - \bar{D})^2$	The variance of the distance distribution.

We have defined D_i as the Euclidean distance from the centroid to the point (x_i^b, y_i^b) on the boundary of the blob, as illustrated in Figure 5.

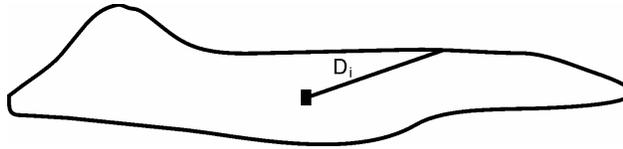


Figure 5: Radial distance, from the centroid to the perimeter of the blob.

Obviously, the set of these radial distances $\{D_i; i=1 \text{ to } m\}$ provides a complete description of the shape of the blob. The DSP module produces the characteristics listed in the above table.

2.3. Invariant characteristics

The intensity of the light received from an object, at a certain distance, depends on the transparency of the medium between the object and the observer. In order to cancel this effect and obtain characteristics that are proper to the observed objects themselves, we shall consider preferably features that are ratios of the light intensities. Furthermore, they should be invariant under translations and rotations in a plane perpendicular to the line of sight.

We note that all the characteristics listed in Section 2.2 possess this invariance except of course for the centroid coordinates and the angle of orientation of the target. However, many of these characteristics depend on the distance "d" between the object and the observer, so that they do not correspond to intrinsic properties of the objects. This is the case for the characteristics that are calculated with the intensities of the target pixels, because intensities vary as $1/d^2$. Similarly, the observed linear dimensions of an object, such as its radial distances and its perimeter, vary as $1/d$. The dependence on d of the other characteristics can easily be computed from these two facts. Thus, for example, the observed area will vary

as $1/d^2$, the observed moments of inertia as $1/d^4$, etc. Based on these considerations, we define the following variables that are independent of the line of sight distance "d". Table 4 shows the invariant characteristics that we have defined. The numbers that appear in parenthesis represent the identification number we will use hereafter when referring to these characteristics.

Invariant characteristics of a blob	
Formula	Designation
(1) Z_{\max}/\bar{Z}	The normalized maximum intensity.
(2) \bar{Z}/A	The normalized average intensity.
(3) $\frac{255 \mu_Z^2}{\bar{Z}^2}$, (4) $\frac{255^2 \mu_Z^3}{\bar{Z}^3}$	The normalized second (variance) and third moments of the intensity distribution.
(5) $\sqrt{I_{\max}}/A$, (6) $\sqrt{I_{\min}}/A$	The normalized square root of the maximum and the minimum moments of inertia.
(7) e	The eccentricity.
(8) R	The roundness.
(9) D_{\max}/\sqrt{A} , (10) D_{\min}/\sqrt{A} , (11) \bar{D}/\sqrt{A}	The normalized maximum, minimum and average radial distances.
(12) μ_D^2/A	The normalized variance of the radial distance distribution.

Table 4: The numbers in parenthesis is an identification number we use for that characteristic.

We note that the eccentricity e and the roundness R are used as such because these variables are already invariant under translations along the line of sight. The angle of the principal axis of minimum inertia was not used because it is not a characteristic of an object's shape.

There exist, of course, many other combinations of the initial variables that correspond to distance and rotation independent features. The particular choice we made here was motivated by the fact that dividing the variables by the average intensity \bar{Z} and the area A should not degrade too much their precision. Indeed, there should not be large measurements errors in \bar{Z} and A themselves, because their calculation involves computing a sum over all the pixels of the blob, and it is expected that the normally distributed random measurement errors, made at each pixel, cancel out if their number is large enough. On the other hand, normalizing with variables that are obtained in a single measurement, such as I_{\max} or D_{\max} could result in an appreciable loss of precision.

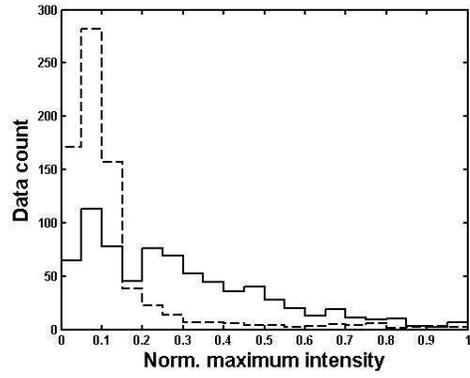
3. INDIVIDUAL CHARACTERISTICS DISCRIMINATING POWER

When designing artificial neural networks, a phenomenon, known as "the curse of dimensionality", makes it desirable to try and minimize the number of components of their inputs. This comes from the fact that neural networks essentially realize a representation of the probability density function of the data used to train them. Thus, the number of samples that are required to yield a reasonable approximation of this function grows exponentially with the number of dimensions of the input vectors (see Section 4.3 of Duda, Hart and Stork [19]).

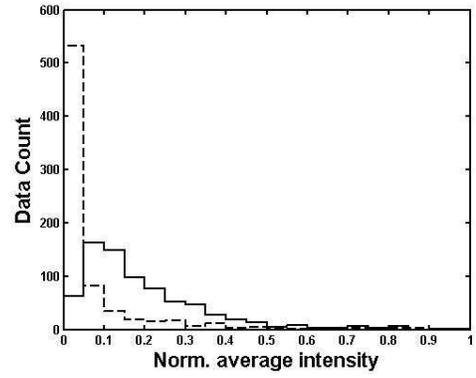
We therefore try and minimize the number of features that we will use to characterize the blobs while keeping those that have the largest discriminating power. In order to see which ones these are, we construct the histograms that correspond to the number of blobs as a function of the values of an invariant feature. The features with the largest discriminating power will be those for

which there are two separate regions of the feature values in which the majority of the aircraft and of the flare blobs lie.

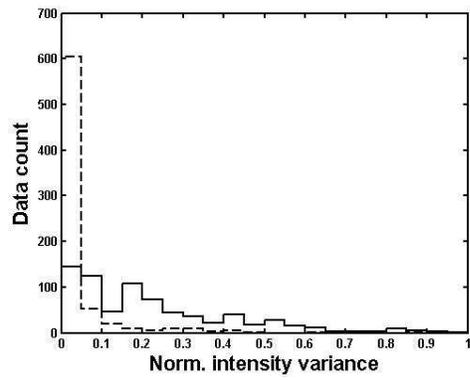
Firstly, in order to make easier the comparison of these histograms, we normalize the values of all the features so that they lie in the interval $[0, 1]$, as follows. Let C_{\max} and C_{\min} be respectively the largest and the smallest value of a particular feature, over all the aircraft and flare images, then if a blob has the value C for this feature, this value is replaced by $C_{\text{norm}} = (C - C_{\min}) / (C_{\max} - C_{\min})$, which is then dimensionless. We then divide the interval $[0, 1]$ in 20 equal subintervals and, for each of these sub-interval, we count the number of aircraft and of flare blobs for which the value of this feature lies in this sub-interval. We then plot the staircase curve where the ordinate is the number of objects counted and the abscissa is the interval $[0, 1]$, divided in the 20 sub-intervals. Figure 6 shows the histograms we obtained for the two classes of objects on the same graph; those for the aircrafts are shown with solid lines and those for the flares with dashed lines. The data sets we used for the present study consist in the features of 740 blobs of each aircrafts and flares; all of them were used in constructing the histograms.



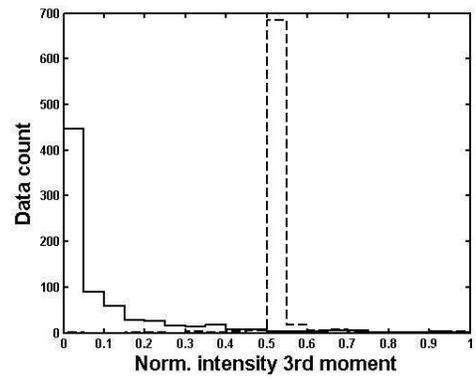
(a)



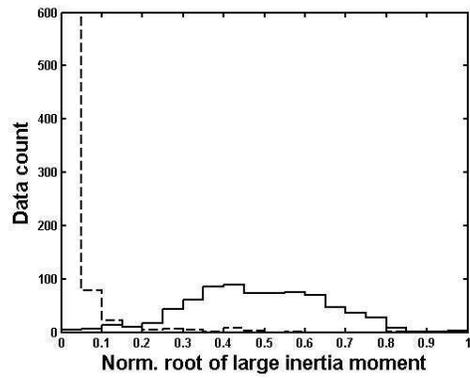
(b)



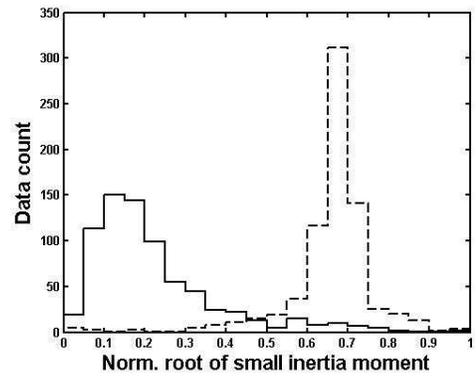
(c)



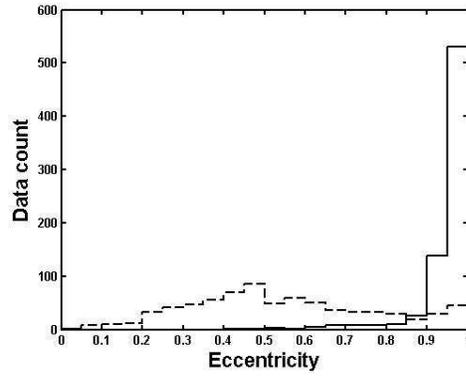
(d)



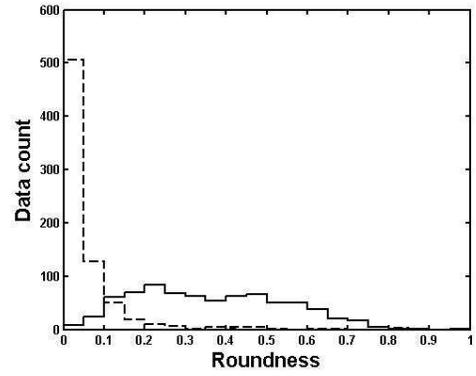
(e)



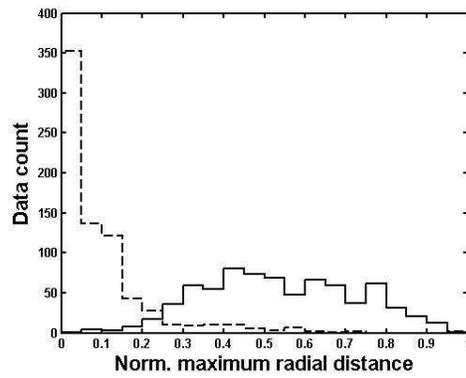
(f)



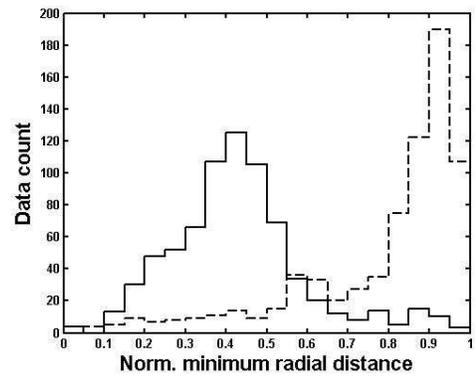
(g)



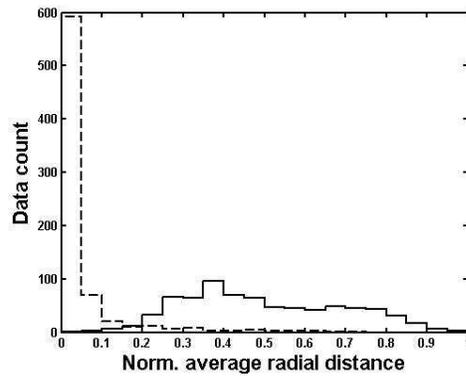
(h)



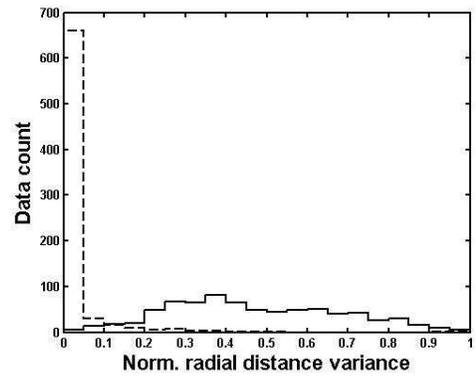
(i)



(j)



(k)



(l)

Figure 6: Histograms of the invariant features of the blobs that correspond to aircrafts and flares. The aircraft and flare histograms are respectively represented by solid and dashed lines. In these histograms, the feature values are separated in 20 bins, and the ordinate is the number of blobs that have the value of this feature within the boundaries of the corresponding bin.

As these graphs make conspicuous, there are certain features for which there are definitely much more aircrafts than flares in a certain domain of values of that feature and vice versa. Such features should possess a good discriminating power. This is the case for the third moment of the intensities, the moments of inertia, the eccentricity, the roundness and the minimum radial distance. Some others have little discriminating power, such as the maximum intensity, the average intensity, the intensity variance because the majority of the aircraft and flare blobs have essentially similar values for this feature. It is interesting to note that the histograms for the eccentricity and the minimum radial distance seem to indicate the presence of two distinct populations of flares in that they have two separate discernable peaks. When one looks at which flares have properties that lie within these peaks, one indeed finds that there is one family of flares that have the appearance of fireballs and another family that are less symmetrical and often have a long tail. The latter family is responsible for the smaller peak that lies within the large aircraft peak in the histograms. This property entails that these flares will be more difficult to discern from the aircrafts. Based on these graphs, we decided to use only the last nine characteristics, i.e. those corresponding to the sub-graphs (d) to (l), as components of the input vectors, considering that the first three have very low discriminating power.

4. THE REDUCED COULOMB ENERGY NEURAL NETWORK

The Reduced Coulomb Energy neural network is a particular type of Radial Basis Functions (RBF) neural network that was devised by Leon Cooper et al. [56] for pattern identification. Its name suggests the analogy that exists between its

neurons and electric charges. The adjective "Restricted" is used to refer to the fact that the influence fields of its neurons, or charges, have a finite range and are bounded at the origin, whereas the Coulomb field about an electric charge varies as the inverse of the distance to that charge, i.e. as $1/r$. It should also be noted that in the RCE neural network, the neurons /electric charges can exist in a space of any dimension. Many different types of influence fields can be considered and will work as well as the truncated $1/r$ field. The simplest form, which is the one that is used for the RCE implemented in the ZISC, is a field that is constant within a finite sphere about the generating charge, and zero outside of that sphere.

4.1 The structure of the RCE network

There are two layers and two kinds of neurons in the RCE neural network. One kind of neurons, the RCE neurons per se, are assembled in the first layer, while the second layer is made up of neurons that act as logical OR functions. The structure of the RCE network and the way information flows through it are shown in Figure 7. In this drawing \mathbf{X} represents an input vector that is fed to each of the neuron in the first layer of the neural network. Each of these neurons reacts by producing an output $y = \phi(\mathbf{X})$, which differs from neuron to neuron, according to their internal state. These outputs are then sent to one of the logical OR units, which then produce their own output z . We will describe these processes in details below.

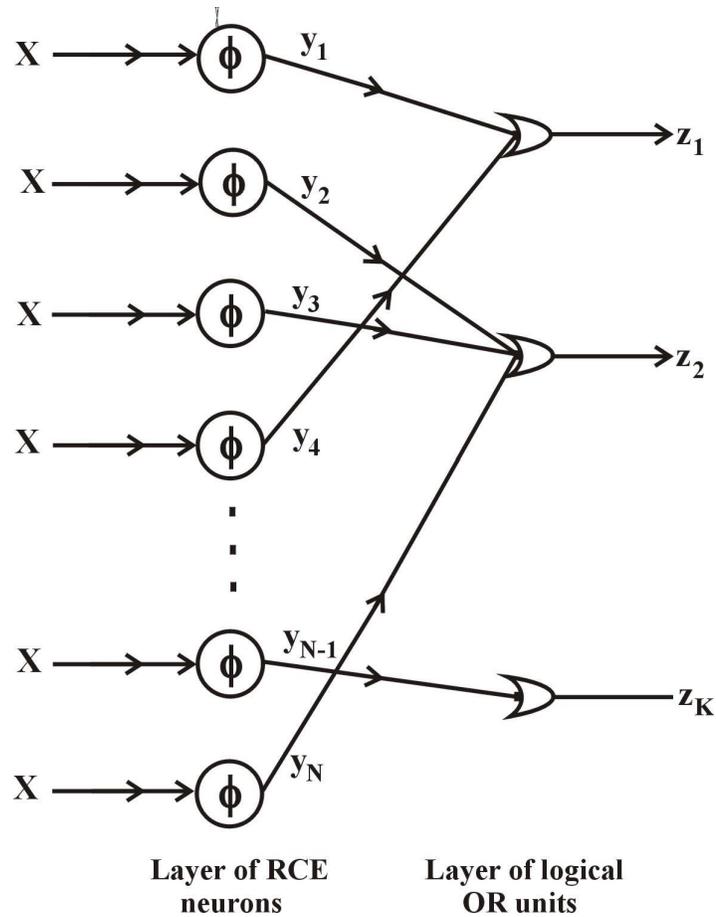


Figure 7: Structure of the RCE neural network.

4.2 The RCE neurons

Each of the RCE neurons is characterized by two parameters: a reference vector, which corresponds to the position in space of the associated "electric charge", and a non-negative scalar, which corresponds to the radius of the sphere in which its influence field is non-zero. These parameters are changed during the learning process of the neural network, so that each neuron ends up having its own reference vector and radius of influence. The information contained in the network is stored in these parameters, just as it is stored in the synapses of biological neurons.

Suppose that the patterns to be classified are represented by n -component vectors that belong to a vector space \mathbf{P}^n . Then the reference vectors for the RCE neurons will also be vectors in \mathbf{P}^n . Here is how these neurons function. When the i -th neuron receives an input vector \mathbf{X} , it computes the distance of this vector to its reference vector \mathbf{C}_i . If this distance is smaller than the radius R_i of its influence field, then it outputs $y_i = 1$, if it is not then it outputs $y_i = 0$. Mathematically, this can be expressed as follows:

$$y_i = \phi(\mathbf{X}, \mathbf{C}_i, R_i) = H(R_i - d_i) = \begin{cases} 1 & \text{if } d_i < R_i \\ 0 & \text{if } d_i \geq R_i \end{cases}$$

where H is the Heaviside function and d_i is the distance between \mathbf{X} and \mathbf{C}_i . It is said that a neuron is "excited" or "has fired" when its output is 1. Figure 8 illustrates this situation for 2-dimensional pattern vectors. Because the pattern vector \mathbf{X} lies in the sphere of influence of the first neuron, it makes it fire; its output will be $y_1 = 1$, while the other nearby neuron remains inactive; its output will be $y_2 = 0$.

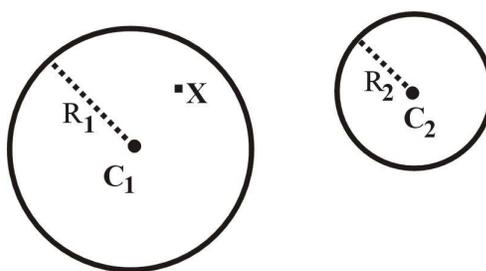


Figure 8: Illustration of the spherical regions of influence of two RCE neurons in 2-dimensional space. These regions are respectively centered on \mathbf{C}_1 and \mathbf{C}_2 , and have their respective radius equal to R_1 and R_2 . A pattern vector \mathbf{X} that lies in the region of influence of the first neuron will make it fire, while the second neuron will remain inactive.

In the ZISC implementation of the RCE network, there are two different distance functions that can be selected by the user: the L_1 or Manhattan distance, and the L_{Sup} distance. We recall that if \mathbf{A} and \mathbf{B} are two vectors, with respective components A_k and B_k , with $k = 1$ to n ,

$$\text{their } L_1 \text{ distance is } D_1(\mathbf{A}, \mathbf{B}) = \sum_{k=1}^n |A_k - B_k|$$

$$\text{and their } L_{\text{Sup}} \text{ distance is } D_{\text{Sup}}(\mathbf{A}, \mathbf{B}) = \max_{k=1, \dots, n} |A_k - B_k|.$$

We note that when these distances are used instead of the Euclidean distance, the "spheres" of radius R , centered at \mathbf{C} , defined as the set of points \mathbf{x} such that $D(\mathbf{x}, \mathbf{C}) \leq R$ are not geometrical spheres. Indeed, if D is the L_1 distance, this region is actually diamond shaped, with diagonal equal to $2R$, while if D is the L_{Sup} distance, it is a square, centered on \mathbf{C} , with side equal to $2R$. Nevertheless, in illustrating the regions of influence of the ZISC neurons, we shall continue to draw circles, as in Figure 8, these being meant as representations of spheres defined according to the adopted definition of distance.

4.3 How the RCE Identifies Patterns

Suppose that there are K types of patterns that the RCE neural network has to classify. Then there will be K logical OR units put in its output layer, each one of them corresponding to one of the K classes of objects. When a feature vector is presented to this neural network, some of the RCE neurons of the first layer will fire while some others remain inactive. The pattern input will be correctly identified if all the first layer neurons that it excites are connected to the OR unit associated with the proper type of this pattern. This OR unit will then output a

one while all other OR units output a zero. We shall explain below how the RCE neural network can be trained with examples of patterns so that it ends up behaving this way.

Mathematically speaking, the RCE neural network works as a function F_{RCE} from the pattern space \mathbf{P}^n to the binary space $\{0, 1\}^K$, that is the space of K-bit binary numbers. Let us denote by D_m the domain of the pattern space that is the union of the spheres of influence of all the RCE neurons of type "m". Figure 9 illustrates these domains for the case in which there are 3 types of objects ($K = 3$) and the pattern space is 2-dimensional ($n = 2$). The function F_{RCE} implemented by a well trained RCE network is such that the binary number $\mathbf{Z} = F_{RCE}(\mathbf{X})$ will contain only zeros except for a one at its m-th position, when the vector \mathbf{X} lies in the domain D_m . If \mathbf{X} lies outside of any of the domains D_m , $m = 1..K$, then all the bits of \mathbf{Z} will be zero, which is interpreted as meaning that the neural network does not know the identity of that particular pattern. If it happens that the data with which the network is trained is ambiguous, then some of the domains D_m , $m = 1..K$, will overlap. Then, an input vector \mathbf{X} that lies in the overlap of two or more domains, will result in the output binary number \mathbf{Z} having more than one "one", which will be interpreted as meaning that the neural network is undecided.

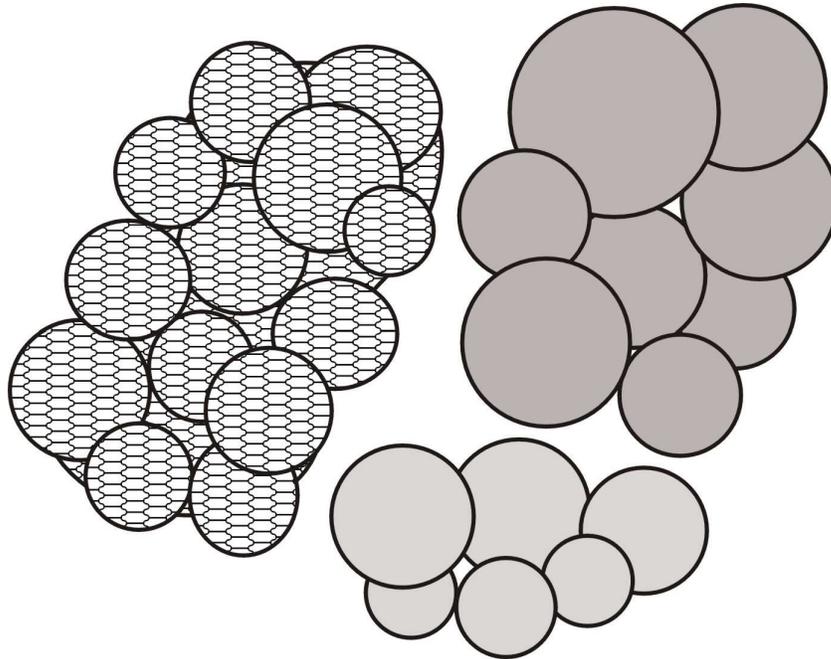


Figure 9: Three domains of the pattern space P^2 that are the union of the spherical influence fields of RCE neurons of three different types.

4.3 Training the RCE neural network

The RCE neural network is trained "by example", which means that its training process consists in showing it samples of feature vectors for each category of objects. The neural network has a mechanism to modify its internal parameters in order to realize the relation that exists between the feature vectors and their category. Here is how its training is done.

When the ZISC RCE neural network is initialized, all of the neurons in its first layer are "unallocated" which means that they are considered as not being part of the neural network as such. Their reference vector and the radius of their influence field are indeterminate. Some of these neurons will be "allocated", that is, added to the neural network, during the training process, as the need arises. They will then be assigned a reference vector and a range, and will be connected

to one of the output OR units. Note that when the training starts, it is not known how many neurons will end up being in the final RCE network. The only two parameters that have to be set by the user at the start are: the maximum and the minimum radius, R_{\max} and R_{\min} , which will be allowed for the spheres of influence of the RCE neurons, the role of which will be made clear hereafter.

Let the training data set be $\{(\mathbf{X}_i, K_i), i = 1..N\}$, where the \mathbf{X}_i 's are feature vectors and the K_i 's are the types of objects they correspond to. These data will be presented to the network as input, one after another, in a random order. When \mathbf{X}_j , the first of feature vector, is submitted to the network, a first RCE neuron is allocated: its reference vector \mathbf{C}_1 is set equal to \mathbf{X}_j , and its range R_1 is set equal to R_{\max} . This neuron is then connected to the first OR unit that will, from now on, correspond to category K_j of objects, which we call Cat-1. The domain D_1 of the feature space, described above, now consists in only the spherical region of influence of this first allocated neuron. During the learning process, more RCE neurons and OR units will be allocated, and there will be more than one domain defined in the pattern space \mathbf{P}^n by the spherical influence fields of allocated RCE neurons. At this point, when another vector \mathbf{X}_k of the training set that belongs to the category K_k is presented to the network, one of the following four cases can occur.

Case 1: The point \mathbf{X}_k in the vector space \mathbf{P}^n lies inside one of the influence field of one or more RCE neurons that belongs to the same category as K_k , and none that belongs to another category. Figure 10a illustrates this situation with a pattern vector \mathbf{X} and three neighboring neurons in a 2-dimensional feature space. This pattern vector is shown to lie inside the influence field of a single neuron that

belongs to category Cat-i, which is the same as the category of \mathbf{X} . The OR unit of the network for category Cat-i will then output a one, and all other OR units a zero. This is the correct output the network should produce so that its parameters are left unchanged.

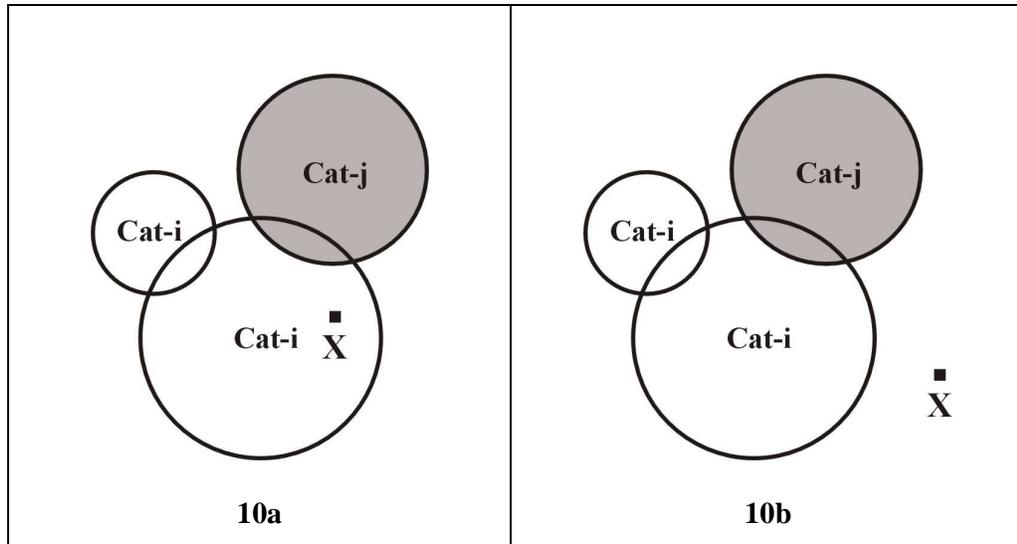


Figure 10a: Geometrical situation in the feature space \mathbf{P}^2 that corresponds to Case 1 in training the RCE network. **Figure 10b** corresponds to Case 2.

Case 2: The point \mathbf{X} , which belongs to category Cat-i, lies outside the influence field of all the RCE neurons presently in the network, as illustrated in Figure 10b, for a 2-dimensional feature space. Then, a new neuron will be allocated, with its reference vector \mathbf{C} set equal to \mathbf{X} , and its range set equal to the minimum of R_{\max} and the distances between \mathbf{X} and the center of RCE neurons of other categories than Cat-i. This new neuron is then connected to the OR output unit for category Cat-i. Figures 10a and 10b illustrate the two geometrical situations in the space \mathbf{P}^2 that are covered by this case.

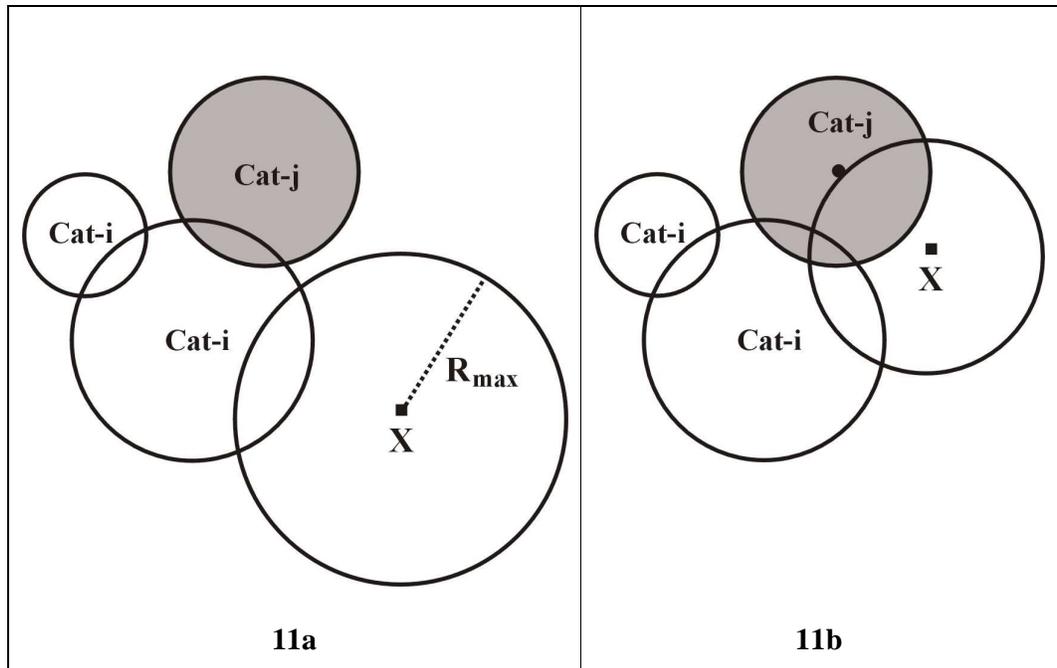


Figure 11a: Geometrical situation in the vector space \mathbf{P}^2 that corresponds to Case 2, when the radius of the sphere of influence of the new neuron centered on \mathbf{X} has its radius set to R_{\max} .

Figure 11b: Also in Case 2, but when that radius will be set to the distance to the center of the closest neuron of a category other than that of \mathbf{X} .

Case 3: The point \mathbf{X} , which belongs to category Cat-i, lies in a region where there is an overlap of the influence fields of RCE neurons that belong to many categories, one of which is the correct one. This situation is illustrated in Figure 12a for a 2-dimensional feature space. Then the learning algorithm reduces the radius of the sphere of influence of all the RCE neurons that are associated with the wrong category, so that the \mathbf{X} lies on the limit of that sphere. The resulting geometrical situation in the vector space \mathbf{P}^2 is illustrated in Figure 12b.

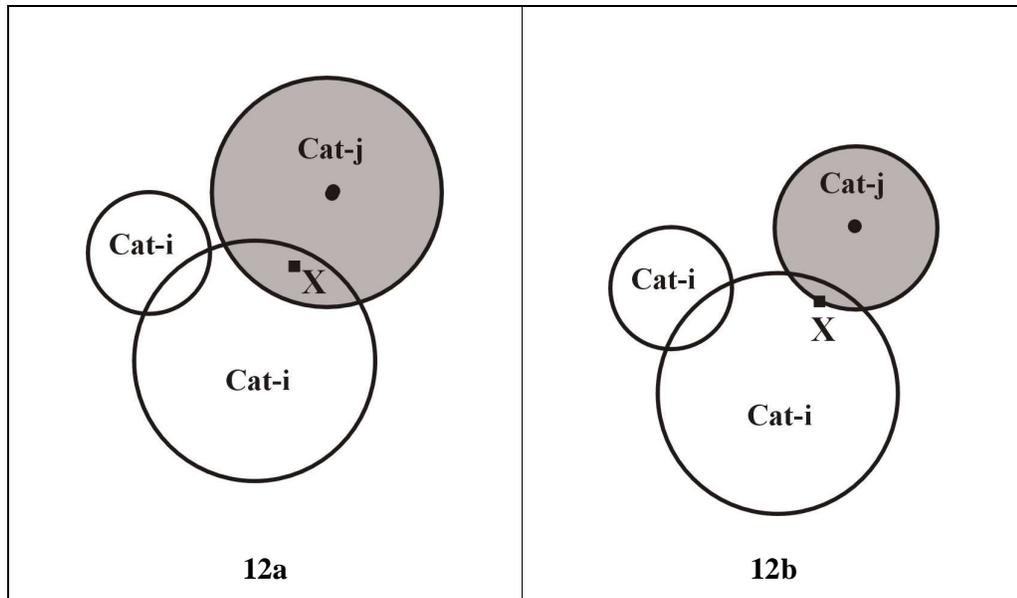


Figure 12a: The geometrical situation where the training vector \mathbf{X} lies in a region in which the influence fields of RCE neurons with different categories overlap. **Figure 12b:** The final situation produced by the training algorithm.

Case 4: The point \mathbf{X} , which belongs to category Cat-i, lies only in the influence fields of RCE neurons of the wrong categories. Figure 13a illustrates this situation in a 2-dimensional pattern space. Then, a new neuron will be allocated, with its reference vector \mathbf{C} set equal to \mathbf{X} , and its range set equal to the minimum of R_{\max} and the distances between \mathbf{X} and the center of RCE neurons of other categories than Cat-i. This new neuron is then connected to the OR output unit for category Cat-i. Furthermore, the radius of all the RCE neurons that contain \mathbf{X} in their sphere of influence is reduced until \mathbf{X} and lies on the limit of these spheres. Figure 13b illustrates the resulting geometrical situation in a 2-dimensional feature space.

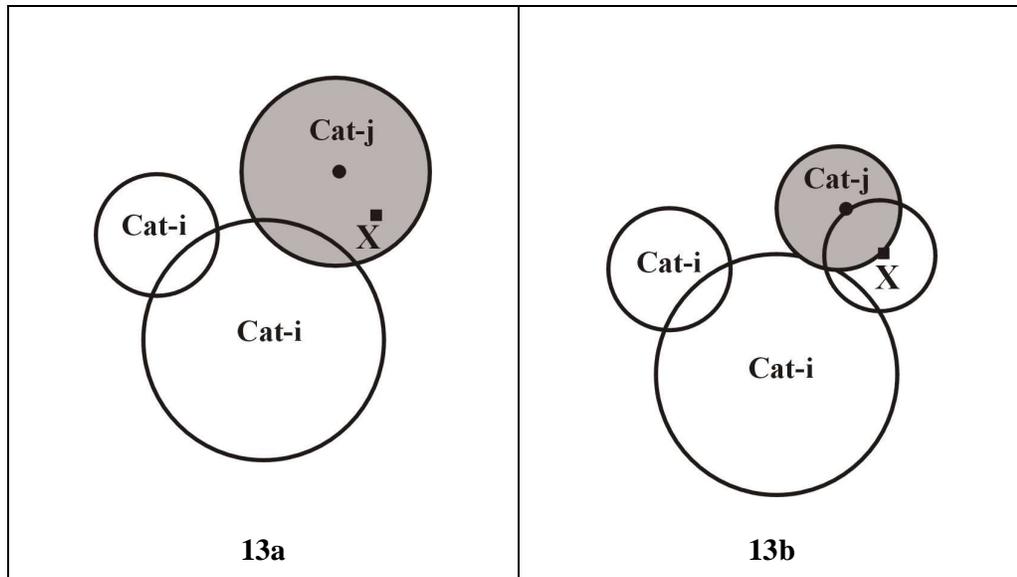


Figure 13a: The geometrical situation in which the feature vector \mathbf{X} lies only in the influence fields of RCE neurons with the wrong categories. **Figure 13b:** The final configuration produced by the training algorithm.

During the training process, some RCE neurons can be unallocated, that is removed from the neural network. This will happen when the radius of their sphere of influence would have to be reduced below R_{\min} .

In general, the network will learn properly only when the training set of sample data is presented many times with the training algorithm kept activated. How does one know then that enough training has been done? This is actually a question that is also pertinent to all other types of neural networks. One answer consists in using another set of pattern vectors that it has never seen as "validation set". The training mechanism for the neural network is then turned off, and the vectors of this set are presented to it. Statistics are gathered on its performance on this set, and if this is judged to be satisfactory, then the training is considered complete and the neural network ready to be used for pattern identification.

Another way of determining that learning is complete is when the parameters of

the neural network do not change anymore when the training set of pattern vectors is presented to it. This is the approach we have taken here.

4.4 The ZISC set-up

As mentioned above, our study was done with the ZISC installed on the EZB 624 PCI board. The implementation of the RCE neural network implemented on that board can have up to 624 neurons, and requires input vectors to have at most 64 components, each of which has to be an 8-bit integer. The components of our feature vectors had therefore to be normalized so that they are integers in the range 0 to 255. In this ZISC, the radius of the influence field of RCE neurons is stored in a 14-bit register, so that its values are in the 0 to 16383 interval. Because the ZISC has been expressly designed for image analysis and pattern recognition, it comes with a library of functions that cover all we need for our application. The ZISC on the PCI board is accessed through commands in Visual Basic or Visual C and can have up to 624 neurons.

The ZISC requires the pattern vectors to be presented sequentially, one component at the time. The chip then distributes each component of its input vector \mathbf{X} to all neurons in parallel. Each time a component is sent to the ZISC, all neurons simultaneously calculate the difference between it and the corresponding component of its reference vector \mathbf{C}_i , $i = 1$ to 624. All neurons then also simultaneously update the value of the distance $D(\mathbf{X}, \mathbf{C}_i)$, whether this is calculated with the L_1 or the L_{Sup} norm. Once the last component of \mathbf{X} is received and processed, each neuron determines whether or not "fire", i.e. to output a 1 or a 0, by comparing the distance it calculated with its radius R_i . When used in

pattern recognition mode, the ZISC then writes the distances and the category of each neuron that fired in some output registers. The PC software that communicates with the ZISC board is then used to retrieve this information and announce the category of the pattern. When used in learning mode, the user only has to present to the ZISC a sequence of patterns with their associated category; the whole RCE learning mechanism is implemented directly on the hardware chip and requires no further user intervention.

Details of the operation of the ZISC and of the data manipulations carried out inside the chip, with their timing, can be found in W.C. Deck's MSc Thesis [13]. The ZISC we used operates at 20 MHz. The documentation that comes with [61] claims that it takes $3.2 \mu\text{s}$ to receive a 64 component input vector and calculate its distance to all RCE neuron reference vectors, an additional $0.5 \mu\text{s}$ to place the distances and categories in the output list, and $2 \mu\text{s}$ to read a distance-category pair from the ZISC registers. This gives a processing time of the order of $5.7 \mu\text{s}$ per pattern vector. On the other hand, the same documentation claims that the ZISC can evaluate more than 250,000 pattern vectors per second. This would correspond to a time of $4 \mu\text{s}$ required to evaluate a pattern vector. This would be the order of magnitude of the processing time to be expected when the ZISC chip is mounted in a dedicated pattern identifier, as would be the case in our final aircraft-flare discrimination system. We note that when the ZISC is accessed on a PCI card in a PC, as is the case in our experiment, there is an additional communication overhead. We evaluated this overhead in our experiments.

5. RESULTS

5.1 Identification Efficiency Tests

Our ZISC neural network was trained and tested with the 1480 feature vectors of an equal number of single aircraft and single flare blobs that we had at our disposal. Its ability to discriminate between aircrafts and flares was tested with the method of M-fold cross-validation. This is a standard method for such a task that is discussed in most textbooks on pattern classification, as for example in Section 9.6.2 of Duda, Hart and Stork [19], in Section 9.8.1 of Bishop [4] and in Salzberg [60] and Ripley [57]. According to this method, the whole data set S of size N is randomly divided into M disjoint sets S_i , with $i = 1, \dots, M$, of equal size equal to N/M . The set S is initially randomly pruned so that it becomes an integer multiple of M . Then M similar experiments are carried out as follows. In the first one, the ZISC neural network, which starts with all its neurons inactive, is trained with the large set $(S - S_1)$. When the training is complete, the neural network is tested on the smaller set S_1 , which it has never seen in its training phase, and its errors are noted down. The second experiment follows exactly the same protocol; the ZISC is re-initialized so that its neural network has no active neurons, and this time the set S_2 is used in the role played by S_1 in the first experiment. This same experiment is thereafter repeated $(M-2)$ times, with each of the other S_i in the role of S_1 . For fairly large data sets, as the one we have here, it is generally considered that M should be taken between 5 and 10 (See for example Bishop [4], Duda, Hart and Stork [19], Salzberg [60] and Ripley [57]). We shall use 10 hereafter so that each subset S_i has 148 feature vectors.

An advantage of the M-fold cross-validation method is that it allows one to compute a confidence interval for the error rate. Indeed, if the probability that the neural network makes an error when asked to identify an object is represented by "p", the probability that it makes k errors when asked to identify N objects follows the binomial distribution (see Duda, Hart and Stork [19]):

$$P(k) = \binom{N}{k} p^k (1 - p)^{N-k}$$

and the maximum-likelihood estimates for p is $\hat{p} = \frac{K}{N}$, where K is the number of errors observed in our tests. According to the central limit theorem, for N as large as we have here, confidence intervals for the actual error rate can be computed as if the binomial distribution were a normal distribution. (See Section 9.1 of Walpole, Myers and Myers [71]). Thus, (1- α)% of the time, the error rate will be found in the confidence interval

$$\left[\hat{p} - Z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}}, \hat{p} + Z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \right] \text{ where } Z_{\alpha/2} \text{ is the positive number such}$$

that the area under the standard normal distribution in the interval $[-Z_{\alpha/2}, Z_{\alpha/2}]$ is (1- α). As can be seen in this formula when the number of data N is decreased, the confidence interval widens, that is the value computed for the error rate is less precise. The reciprocal behavior is obviously true when N is increased.

In our tests of the ZISC neural network, we used the L_1 norm to calculate the distances (the Manhattan distance then) between patterns, with the parameters: $R_{\max} = 5000$ and $R_{\min} = 10$, which were set after running some preliminary tests with the network. On the average, our ZISC neural network stabilized after only 3 training epochs, that is after all the training vectors were presented to it three

times. When the trained neural network was tested on the data used to train it, it never made any mistake (which is not necessarily always the case). Table 5 shows the results we have obtained in the 10 different experiments described above, when the trained neural network was tested on the validation data sets.

Exp. Number	Number of errors						Total
	Aircraft Blobs			Flare Blobs			
	F	?	AC&F	AC	?	AC&F	
1	1	3	3	3	2	3	15
2	1	0	3	4	2	4	14
3	1	1	7	4	4	6	23
4	1	3	5	0	1	4	14
5	2	0	8	0	2	3	15
6	0	5	5	2	1	8	21
7	4	0	4	1	3	4	16
8	3	0	2	4	3	3	15
9	0	1	4	1	4	2	12
10	1	1	2	1	1	1	7

Table 5: Identification errors made by the ZISC neural network on the validation data set in the 10 experiments required for its 10-fold cross-validation. The left-most column shows the number of the experiment. The following three columns show the errors made by the neural network in identifying aircraft blobs: an error is of type F if a blob was falsely identified as a flare, of type "?" if the network did not recognize the blob as belonging to any one of the two categories of objects it knows, and it is of type AC&F if both an aircraft and a flare neuron were activated in the network. The following three columns represent the corresponding results for the flare blobs. The last column to the right shows the total number of errors made in the corresponding experiment.

The total number of errors made by the network in the identification of $N = 1480$ blobs is $k = 152$. The maximum likelihood estimate for the probability "p" that

the neural network makes a classification error is therefore $\hat{p} = \frac{152}{1480} = 0.103$, that

is essentially 10.3% so that its expected success rate is 89.7%. As mentioned

above, confidence intervals for the actual error rate can be computed as if the binomial distribution were a normal distribution. Thus, we can say, with 95% confidence, that the neural network will make between $K - 2\sigma$ and $K + 2\sigma$ errors, where $\sigma = \sqrt{N\hat{p}(1-\hat{p})}$, when tested on 1480 sample blobs. This means that it is expected to make between 128 and 175 errors on 1480 identifications; in other words, its success rate will be between 88.2% and 91.3%. A similar calculation shows that 99.7% of the time, its success rate will be between 87.4% and 92.1%. Finally, we present the results of our tests in the form of a confusion matrix (see Kohavi and Provost [41]).

		Confusion Matrix				
		Actual identity				
		Counts		Percentages		
		Aircrafts	Flares		Aircrafts	Flares
ZISC identity	Aircrafts	669	20		90.4	2.7
	Flares	14	659		1.9	89.1
	?	14	23		1.9	3.1
	AC&F	43	38		5.8	5.1

Table 6: Two versions of the confusion matrix for our tests are shown in the white areas of this table. The one on the left-hand side shows the number of patterns that the ZISC classified as aircraft, flare, unknown, and ambiguous. The one on the right-hand side shows the corresponding percentages of patterns put in each class. The first column in each confusion matrix corresponds to feature vectors for aircrafts and the second column to feature vectors for flares.

In order to appreciate the excellent quality of these results, it is worthwhile looking at some of the images on which errors are made by the neural network. Figure 14 shows some representatives of these images; all of its other mistakes being made on very similar images as these.



(a) Aircraft mistaken for a flare.



(b) Aircraft mistaken for a flare.



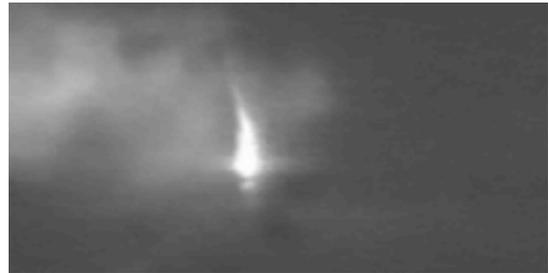
(c) Aircraft mistaken for a flare.



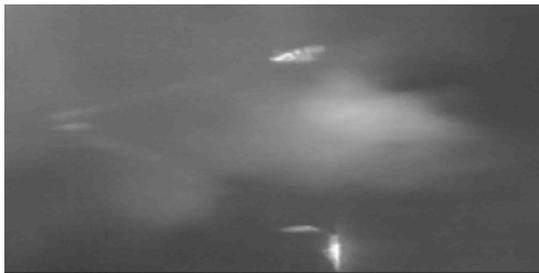
(d) Flare to the right mistaken for an aircraft. Aircraft at bottom mistaken for a flare.



(e) Flare under aircraft mistaken for aircraft exhaust.
Top aircraft recognized.



(f) Flare mistaken for aircraft exhaust.



(g) Flare at bottom of picture mistaken for an aircraft.
Top aircraft recognized.



(h) Flare mistaken for an aircraft.

Figure 14: Images of aircrafts and flares on which the RCE neural network made mistakes.

As these pictures show, the discrimination task that the neural network was asked to perform was far from trivial . Indeed, even human experts would be uncertain

and make mistakes with many of these images. In the first picture, the aircraft is seen from behind and does look very much like certain flares. The second picture is taken from a short video sequence in which the aircraft's appearance is very different from that in the majority of the other pictures, so that its characteristics had not really been learned by the neural network. This situation could be easily remedied by exposing the neural network to more video sequences of this type of aircraft image. In the 3rd picture, the aircraft is far away and its shape is very imprecise. Some of the flares for which the neural network makes mistakes have shapes that are close to those of aircrafts seen from behind, as shown in pictures (d) and (h). Some others, such as pictures (e) and (f), are very similar to those of aircraft exhausts, which actually happens to be all that is seen in some of our training infrared pictures of aircrafts. We note that some flares happen to deploy with characteristics that are so close to those of aircrafts that they are bound to be misidentified. Similarly, an aircraft that is far away and seen from behind will almost certainly be mistaken for a flare. As mentioned in our introduction, considering time dependent characteristics could then help improve the accuracy of discrimination in such cases.

5.2 Is it Fast Enough ?

As mentioned at the end of Section 4, the ZISC documentation claims that it can classify a pattern vector in a little less than 4 μ s. This time does not include the time required for communications with the ZISC chip itself. With our ZISC on the EZB 624 PCI board, this is the time required by the PC to communicate with the PCI board and the time required by that board to access the ZISC chip. In order to obtain an estimate of this communication time, we measured the total

time it takes to do a simple read operation from the PC to the ZISC, which executes in one clock cycle. We then subtracted the time of one clock cycle from the measured time to obtain the time $T_0 = 1.5 \mu\text{s}$ required by the communication process. We used the Intel machine language command "rdtsc" (read time stamp counter) to determine elapsed CPU clock cycles on the 2GHz PC that we were using, giving our time measurements a resolution of 0.5 ns. For each vector that is processed, the ZISC is accessed at least two and most of the time 3 times. The first time is when it is sent the feature vector as input. The second time is when its status register is read to see if some neurons have fired or if all firing neurons had the same category. If they did, then that category has to be read in the ZISC. Thus, the communication time overhead is at least $2T_0 = 3 \mu\text{s}$ and most of the time it will be $3T_0 = 4.5 \mu\text{s}$. In the experiment we described above, we measured the time required to identify each pattern vector of the validation set, once the neural network had been trained. We found an average time of $8.6 \mu\text{s}$. Since the communication time overhead is at least $3 \mu\text{s}$, the time required by the ZISC to identify a pattern is at most $5.6 \mu\text{s}$. The documentation for the Cognimem chip [11], which has replaced the ZISC, indicates that it takes 10 clock cycles to broadcast the 9 components of our input vector to all the neurons of the network, and 36 other clock cycles for the chip to return the category of the best match. The single Cognimem chip, with 1024 neurons, has its clock at a frequency of 27 MHz. Thus, the 46 clock cycles that are required for identifying a feature vector with the Cognimem chip correspond to $1.7 \mu\text{s}$. The Cognimem chip is therefore faster than the ZISC board that we used.

The maximum speed of fighter aircrafts such as the F-15E, F-16 and F-22 is about Mach 2 to 2.5, that is 2,450 km/h to 3063km/h, according to Air Force Link [1].

According to the Russian Aircraft Corporation [59] that of the MIG-31E is Mach 2.83 that is 3467 km/h. Upon taking Mach 3 as a maximum speed for modern aircraft fighters, we can calculate that such an aircraft will have traveled at most 5.7 mm in the time the ZISC takes to identify its target. With the processing time reported by Cognimem, the distance it traveled would only be 3.2 mm. Amongst the fastest IR guided missiles are the VT-1 version of the Crotale, developed by Thomson CSF Matra (now Thales) [70], and the Raytheon [64] Standard Missile that are reported to fly at Mach 3.5+. Supposing that such a missile did fly at Mach 4, it would then only cover 7.6 mm during the time required to identify the target. There is therefore no doubt that the processing speed of the ZISC is amply sufficient for its incorporation in a real time target seeker and tracker.

6. CONCLUSION

We showed how to construct translation and rotation invariant characteristics from the features of infrared images of aircraft and flares produced by a digital signal processing board. We then determined their discriminating power by constructing their histograms and comparing those obtained for the aircrafts and those for the flares. Those features for which these histograms showed considerable overlap were then dismissed and nine characteristics were left to use as input for our neural network.

In Section 5, we reported and discussed the results we obtained with real infrared video sequences of aircrafts and flares. These showed that the ZISC is very much able to discriminate between these two types of objects, with about 90% correct identification rate. In Figure 14, we presented representative images of those on

which the RCE neural network made its mistakes. One can see that even a human expert would have difficulties recognizing the objects in most of them. It would be worthwhile training the ZISC neural network on more data that also corresponded to a wider variety of situations. This would help determine the optimal performances this device can achieve. Moreover, the efficiency of the aircraft- flare discrimination process could be improved by also taking into account the dynamic features of the objects detected. Such features, computed from a few consecutive video frames, would be added as components to the vector of static characteristics used in the present study.

In our tests, we also measured the time required by the ZISC to identify a feature vector and found it to be at most $5.6 \mu\text{s}$. This time is short enough that the fastest aircrafts and missiles will have traveled only a few millimeters during that process. It is therefore clear that, from the point of view of processing speed, there is no problem in incorporating the ZISC in a real time missile target seeker and tracker.

That fact and the ZISC efficiency in aircraft-flare discrimination allow us to conclude that the ZISC hardware neural network is indeed a candidate of choice for the image identification sub-system of an infrared seeker and tracker.

Acknowledgments

The authors thank Dr A. Morin from DRDC-Valcartier for providing the infrared video sequences used in this study and some essential information about the infrared seeker and tracker at this institution.

REFERENCES

- [1] Air Force Link is a web site of the Office of the US Secretary of Air Force (Public Affairs). Document at: <http://www.af.mil/factsheets/factsheet.asp?id=103>
- [2] Air Force Link is a web site of the Office of the US Secretary of Air Force (Public Affairs). Document at: <http://www.af.mil/factsheets/factsheet.asp?id=78>
- [3] Bhuiyana S. M. A., Alam M. S., Alkanhal M.: Automatic target recognition and tracking in FLIR imagery using extended maximum average correlation height filter and polynomial distance classifier correlation filter. Proceeding of SPIE, Automatic Target Recognition XV, Edited by Sadjadi, Firooz A. Vol. 5807, 1-15 (2005).
- [4] Bishop C.M.: Neural Networks for Pattern Recognition. New York: Oxford University Press 1995.
- [5] Budnyk I., Chebira A., Madani K.: ZISC neural network base indicator for classification complexity estimation. Proceedings of the International Workshop on Artificial Neural Networks and Intelligent Information Processing ANNIIP 2007. Angers, France, 38-47, INSTICC PRESS Publisher, Portugal, 2007.
- [6] Cai Y., Hu Y., Siegel M., Gollapalli S. J., Venugopal A. R., Bardak U.: Onboard Feature Indexing from Satellite Lidar Images. Proceedings of IEEE International Workshop on ADC Modelling and Testing (IWADC), Perugia, Italy, IEEE Publisher, 2003.
- [7] Cayouette P.: Réseaux de neurones probabilistes pour la discrimination de cibles par imagerie infrarouge. MSc Thesis, Department of Mathematics and Computer Science, Royal Military College of Canada, May 2002. This thesis is available on demand from the author G. Labonté. It is also available through the normal inter-library loan system.
- [8] Cayouette P., Labonté G., Morin A.: Probabilistic neural networks for infrared imaging target discrimination. Proceedings of SPIE, Automatic Target Recognition XIII, Vol. 5094, 254-265, SPIE Press, 2003.
- [9] Chan L. A., Der S. Z., Nasrabadi N. M.: Dualband FLIR fusion for automatic target recognition Information Fusion Vol. 4, 35-45 (2003)
- [10] Chebira A., Madani K., Mercier G. : Neural Nets Simulation, Emulation and Implementation Multi-Neural Networks hardware and software architecture: Application of the divide to simplify paradigm DTS. Biological and Artificial Computation: From Neuroscience to Technology. Lecture Notes in Computer Science, Vol. 1240, 841-850 (1997)

- [11] Cognimem_1K Information document: Neural network chip for high performance pattern recognition, Version 1.2.1 at the Recognetics web site:
http://www.recognetics.com/download/CM1K_datasheet.pdf
- [12] David R., Williams E., de Tremiolles G., Tannhof P.: Noise reduction and image enhancement using a hardware implementation of artificial neural networks. Proceedings of the SPIE Ninth Workshop on Virtual Intelligence/Dynamic Neural Networks. Stockholm, Sweden, Vol.3728, 212-21 (1999).
- [13] Deck W.C.: Evaluation of the Computational Capacity of the Zero Instruction Set Computer in Target Tracking. MSc Thesis, Department of Mathematics and Computer Science, Royal Military College of Canada, 2007. This thesis is available on demand from anyone of the two authors of this article. It is also available through the normal inter-library loan system.
- [14] Deck W. and Labonté G.: A hardware neural network for target tracking. Proceedings of the SPIE Defense and Security Symposium, Signal and Image Processing for Tracking, Edited by Chodos, S. L., Thompson W. E., Orlando Florida, US, Vol. 6971, 697103-697103-8 (2008)
- [15] de Trémiolles G., Tannhof P., Plougouven B., Demarigny C., Madani K.: Visual probe mark inspection, using hardware implementation of artificial neural networks, in VLSI production. In Biological and Artificial Computation: From Neuroscience to Technology, Vol. 1240/1997 in the Series Lecture Notes in Computer Science, 1374-1383, Springer Berlin, 1997.
- [16] Dias F. M., Antunesa A., Motab A. M.: Artificial neural networks: a review of commercial hardware. Engineering Applications of Artificial Intelligence Vol. 17, 945–952 (2004).
- [17] Dinaki H. E., Shokouhi S. B., Soltanizadeh H.: Recognition of the Real Target in the Rosette Pattern Using Blind Source Separation and Hidden Markov Model, Proceedings of The International Association of Journals and Conferences (IAJC) and the International Journal of Modern Engineering (IJME) Conference (The 2008 IAJC-IJME International Conference), Paper#74, ENG 108, (2008).
- [18] Doo K.-S., Oh J.-S., Jahng S.-G., Hong H.-K., Choi J.-S., Seo D.-S.: Simulation of target detection in ultraviolet and infrared bands. Optical Engineering, Vol. 40, No 11, 2646 - 2654 (2001)
- [19] Duda R., Hart P., Stork D.: Pattern Classification, New York: John Wiley and Sons Inc. 2001.

- [20] Eide A., Lindblad T., Lindsey C.S., Minerskjold M., Sekhniaidze G., Szekely G.: An Implementation of the Zero Instruction Set Computer (ZISC036) on a PC/ISA-bus card. Proceedings of the SPIE Sixth, Seventh, and Eighth Workshops on Virtual Intelligence, Academic/Industrial/NASA/Defense Technical Interchange and Tutorials. International Conferences on Neural Networks, Fuzzy Systems, Evolutionary Computing, and Virtual Reality. Mary Lou Padgett and Thomas Lindblad, Eds., Vol. 2878, 40, SPIE Press 1996.
- [21] Feng Z., Shang-qian L., Da-bao W., Wei G.: Aircraft recognition in infrared image using wavelet moment invariants. *Image and Vision Computing*, Vol. 27, 313–318 (2009)
- [22] Gaughran P.: A Novel Approach to Binary Neighbourhood Operations Utilising a Hardware based RBF Network. *Applied Intelligence*, Vol. 18, No 2, 195-213, (2003)
- [23] Gliever C., and Slaughter D. C.: Neural Network Plant Recognition for Vision Based Robotic Weed control. Proceedings of the ASAE (American Society of Agricultural Engineers) International meeting, August 2001, Paper 01-3104 Session 95: Real-Time Image Applications, Published by the American Society of Agricultural and Biological Engineers, St. Joseph, Michigan, 2001.
- [24] Goddard P.: Advanced Infra Red Countermeasure Solutions. Presented at the 2008 Association of Old Crows (AOC) International Exhibition and Symposium at the Adelaide Convention Centre, May 2008. This paper is available at the web site: <http://www.oldcrows.org.au/symposium/Day2/folder1.htm>
- [25] Goldberg, S.: Infrared Countermeasures: The systems that cool the threat from heat-seeking missiles. In the *Smithsonian Air & Space Magazine*, July 01, 2003, available on line at <http://www.airspacemag.com/how-things-work/infrared.html?c=y&page=1>
- [26] Güven E. :Electrooptical and Infrared Applications. Proceedings of the 6th Int. Seminar Technology for Effective Defense, 7 – 8 May 2003, Defense, Armed Forces Communications and Electronics Association (AFCE), Turkey, 27 – 42, 2003.
- [27] Hammerstrom, D. (2006). A Survey of Bio-Inspired and Other Alternative Architectures. In Waser, Rainer (Ed.) *Nanotechnology*. Vol. 4: Information technology II. Weinheim: Wiley-VCH, 251-282 (2006)
- [28] Han S.-H., Hong H.-K., Jahng S.-G., Seo D.-S., Choi J.-S.: Target position extraction based on instantaneous frequency estimation in a fixed-reticle seeker. *Optical Engineering*, Vol. 39, No 9, 2568-2573 (2000).
- [29] Herskovitz D.: Is seeing believing? *Journal Electronic Defence*, Vol. 21, 41-46 (1998).

- [30] Holton, C.: Neural networks speed image classification. *Laser Focus World*, Vol. 43, Issue 6, 83-83 (2007).
- [31] Hong H.-K., Jahng S.-G., Doo K.-S., Choi J.-S., Han S.-H.: Adaptive Infrared Counter Countermeasure for Two Color Spinning Concentric-Annular-Ring Reticle Seeker, *Optical Engineering*, Vol. 40, No 6, 1093-1099 (2001).
- [32] Hung L. L., Webb D. L., Elliott D. F., Chandler V. T.: Image Processing Algorithms for KKVS with IR Imaging Sensors. Report for ROCKWELL INTERNATIONAL ANAHEIM CA, Sep. 1996. Available at the Internet address:
<http://oai.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA319958>
- [33] Jahng S.-G., Hong H.-K.: Simulation of Rosette Scanning Infrared Seeker and Counter-Countermeasure Using K-Means Algorithm. *IEIC Trans. Fundamentals*, Vol. E82-A, No 6, 987-993 (1999).
- [34] Jahng S.-G., Hong H.-K., Choi J.-S., Han S.-H.: Reticles –Nutating Systems, in *Encyclopedia of Optical Engineering*, R. G. Driggers (Ed.), 2417-2430, Marcel Dekker Inc. N.Y. 2003.
- [35] Jahng S.-G., Hong H.-K., Han S.-H., Choi J.-S.: Dynamic simulation of the rosette scanning infrared seeker and an infrared counter-countermeasure using the moment technique, *Optical Engineering*, Vol. 38, No 5, 921-928 (1999)
- [36] Jahng S.-G., Hong H.-K., Seo D.-S. and Choi J.-S.: New infrared counter-countermeasure technique using an iterative self-organizing data analysis algorithm for the rosette scanning infrared seeker. *Optical Engineering*, Vol. 39, No 9, 2397-2404 (2000)
- [37] Khan J. F. and Alam M. S.: Target Detection in Cluttered FLIR Imagery Using Probabilistic Neural Networks. *Proceedings of the SPIE, Automatic Target Recognition XV*. Edited by Sadjadi, Firooz A., Vol. 5807, 55-66 (2005).
- [38] Khan J. F. and Alam M. S.: Efficient target detection in cluttered FLIR imagery. *Proceedings of the SPIE, Optical Pattern Recognition XVI*, Casasent, David P., Chao, Tien-Hsin (Eds.), Vol. 5816, 39-53 (2005).
- [39] Kim, D. H. and C. Slaughter D. C. : Image-based real-time displacement measurement system. *Biosystems Engineering*, Vol. 101, Issue 4, 388-395 (2008)
- [40] Koch E.-C. : Pyrotechnic Countermeasures: II. Advanced Aerial Infrared Countermeasures, Propellants, Explosives, Pyrotechnics, Vol. 31, No.1, 3-19 (2006)

- [41] Kohavi R., Provost F.: Glossary of Terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. *Machine Learning*, Vol. 30, No 2/3, 271-274 (1998).
- [42] Labonté G., Morin A.: Probabilistic neural networks for target discrimination using their temporal behavior. *Proceedings of the SPIE, Defense and Security Conference 2004, Automatic Target Recognition XIV*, Vol. 5426, 46-57, SPIE Press 2004.
- [43] Lindblad T., Lindsey C. S., Minerskjöld M., Sekhniaidze G., Székely G., Eide A.: Implementing the new Zero Instruction Set Computer (*ZISC036*) from IBM for a Higgs search. *Nuclear Instruments & Methods in Physics Research, Section A (Accelerators, Spectrometers, Detectors and Associated Equipment)*, Vol.357, No.1, 192-194 (1995).
- [44] Lindsey C. S., Lindblad T., Eide A. J.: Method for star identification using neural networks. *Proceedings of the SPIE, Applications and Science of Artificial Neural Networks III*, Rogers S. K. (Ed.), Vol. 3077, 471-478 1997.
- [45] Lindsey C. S., Lindblad T., Sekhniaidze G., Székely G., Minerskjöld M.: Experience with the IBM ZISC036 Neural Network Chip. *International Journal of Modern Physics C*, Vol. 6, No. 4, 579-584 (1995)
- [46] Madani, K.: *From Integrated Circuits Technology to Silicon Grey Matter: Hardware Implementation of Artificial Neural Networks. Information Processing and Security Systems, Part III*, 327-351, Springer US, 2005.
- [47] Madani K., de Trémiolles G., Tannhof P.: Étude des Paramètres Influent Dans le Traitement d'Images par des Réseaux Neuronaux Évolutifs. *Proceedings of the 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications SETIT 005*, March 27-31, 2005 – TUNISIA, 2005
- [48] Madani K., de Trémiolles G., Williams E., Tannhof P.: Neural based production yield prediction : an RBF based approach. *Proceedings of the SPIE, Applications and science of computational intelligence II*, Orlando FL, Vol. 3722, 466-475 (1999).
- [49] Madani K., Mercier G., Chebira A., Duchesne S.: Real time intelligent adaptive control on a 16 chips parallel zero instruction set computer (ZISC) single board. *Proceedings of the SPIE*, Vol.3077, 136-47 (1997).
- [50] Meller R.: Neue Lenkflugkörper Luft-Luft Internationale Rüstungskooperation der Luftwaffe, *Soldat und Technik*, Vol. 5, 302-302 (1996).

- [51] MICA infrared seeker brochure at the web site of Sagem, Défense Sécurité, Safran Group at <http://www.sagem-ds.com/pdf/en/D084.pdf>
- [52] Morin A., Lessard P.: Development of an infrared imaging seeker emulator for countermeasure studies. Proceedings of the SPIE, Tracking and Pointing Acquisition XIII, Vol. 3692, 255-268 (1999).
- [53] Morin A.: Adaptive spatial filtering techniques for the detection of targets in infrared imaging seekers. Proceedings of the SPIE, Acquisition, Tracking and Pointing XIV, Vol. 4025, 182-193, (2000).
- [54] Nair D. and Aggarwal J. K. :Robust Automatic Target Recognition in Second Generation FLIR Images. Proceedings of the Third IEEE Workshop on Applications of Computer Vision 1998. WACV96, 262-263, Published by IEEE, 1996.
- [55] Nair D. and Aggarwal J.K.: Bayesian recognition of targets by parts in second generation forward looking infrared images. Image and Vision Computing, Vol. 18, 849–864 (2000)
- [56] Reilly D. L., Cooper L. N., Elbaum, C.: A neural model for category learning. Biological Cybernetics, Vol. 45, 35-41 (1982).
- [57] Ripley B. D.: Statistical Theories of Model Fitting. Proceedings of the NATO Advanced Science Institute on Generalization in Neural Networks and Machine Learning, C. M. Bishop Editor, Series F: Computer and Systems Sciences, Vol. 168, Springer, New York, 1997.
- [58] Roger S. K., Colombi J.M., Martin C.E., Gainey J.C., Fielding K.H., Burns T.J., Ruck D.W., Kabrisky M., Oxley M.: Neural Networks for Automatic Target Recognition. Neural Networks, Vol. 8(7/8), 1153-1184 (1995)
- [59] Russian Aircraft Corporation web site:
http://www.migavia.ru/eng/military_e/MiG_31_E_e.htm
- [60] Salzberg S. L.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data Mining and Knowledge Discovery, Vol,1, 317-327 (1997)
- [61] Silicon Recognition Inc.: ZISC Technology Reference Guide Version 2, 2003. NOTE that the Silicon Recognition web site has been deactivated, the ZISC having been replaced by the Cognimem. Their web site is at: <http://www.recognetics.com/contact.html>

[62] Singstock B. D.: Infrared Target Recognition. MSc Thesis in Electrical Engineering, Faculty of the School of Engineering of the Air Force Institute of Technology, Air University, Dec. 1991.

[63] Soltanizadeh H., Shokouhi S. B.: Increasing Accuracy of Tracking Loop for the Rosette Scanning Seeker Using Improved ISODATA and Intelligent Center of Gravity, Journal of Applied Sciences, Vol. 8, No. 7, 1159-1168 (2008)

[64] Standard Missile-2 (SM-2) brochure, at the Raytheon web site:
http://www.raytheon.com/capabilities/rtnwcm/groups/rms/documents/content/rtn_rms_ps_sm2_datasheet.pdf

[65] Smith L. S.: Implementing neural models in silicon. Handbook of Nature-Inspired and Innovative Computing Integrating Classical Models with Emerging Technologies. 433-475, Springer US, 2006.

[66] Specht D. F.: "Probabilistic neural networks", Neural Networks, Vol. 3, No. 1, 109-118, (1990).

[67] Singh A, Pettersson R., Karlholm J., Berndt A., Brunnström K.: A two-stage approach for target identification and pose estimation in infrared images. Proceedings of the SPIE, Military Remote Sensing. Edited by Kamerman, Willetts, David., Vol. 5613, 177-188 (2004).

[68] Stinger missile brochure, at the web site of Raytheon Cie. at:
http://www.raytheon.com/capabilities/products/stellent/groups/public/documents/content/cms01_057590.pdf

[69] Titterton D. H.: Development of Infrared Countermeasure Technology and Systems. Springer Series in Optical Sciences, Vol. 118, 635-671, Springer Berlin, 2006.

[70] VT-1 brochure at the Thales web site:
<http://www.thalesgroup.com/markets/Activities/Product-page.html?url=/Activities/Air-Defence/Weapon-and-Missile-Systems.html&link=3757664C-2D57-644E-391C-6A2B201A7D23:central&locale=EN-gb&Title=VT1&dis=1&marketId=0A50731B-644D-1E7E-3C5B-150646382719&type=Market>

[71] Walpole R. E., Myers R. H., Myers S. L., Ye K.: Probability & Statistics for Engineers & Scientists, Seventh Edition, Prentice Hall, Upper Saddle River, NJ, 2002.

[72] Yang F., Paindavoine M.: Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification. IEEE Transactions on Neural Networks, Vol. 14 Issue 5, 1162-1175 (2003).

[73] Yu S.; Azimi-Sadjadi, M.R.: Neural network directed Bayes decision rule for moving target classification . IEEE Transactions on Aerospace and Electronic Systems, Vol. 36, Issue 1, 176 - 188 (2000)

[74] Zhang D., Ghobakhlou A. and Kasabov N.: An Adaptive Model of Person Identification Combining Speech and Image Information, Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision, Kunming, China, Vol. 1, 413-18, IEEE Publisher USA, 2004.

[75] Zhao H., Shah S., Choi J. H., Nair D., Aggarwal J. K.: Robust Automatic Target Detection/Recognition System For Second Generation FLIR Imagery. Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision 1998. WACV'98. 262-263, Published by IEEE, 1998.

FIGURE LEGENDS

Figure 1: On the left-hand-side: reticule mask and on the right-hand-side: rosette scan used initially in infrared seekers.

Figure 2: Basic configuration of the infrared seeker and tracker.

Figure 3: Pre-processing of the infrared image and its segmentation.

Sub-figures in Figure 3

- (a) Image produced by the IR camera
- (b) Image processed with a double-gated filter
- (c) Segmentation in three blobs

Figure 4a: EZB 624 PCI board. The eight ZISC78 chips that make the 624 neurons neural network are visible to the right of the board. **Figure 4b:** The Cognimem chip with 1024 neurons.

Figure 5: Radial distance, from the centroid to the perimeter of the blob.

Figure 6: Histograms of the invariant features of the blobs that correspond to aircrafts and flares. The aircraft and flare histograms are respectively represented by solid and dashed lines. In these histograms, the feature values are separated in 20 bins, and the ordinate is the number of blobs that have the value of this feature within the boundaries of the corresponding bin.

Figure 7: Structure of the RCE neural network.

Figure 8: Illustration of the spherical regions of influence of two RCE neurons in 2-dimensional space. These regions are respectively centered on C_1 and C_2 , and have their respective radius equal to R_1 and R_2 . A pattern vector \mathbf{X} that lies in the region of influence of the first neuron will make it fire, while the second neuron will remain inactive.

Figure 9: Three domains of the pattern space \mathbf{P}^2 that are the union of the spherical influence fields of RCE neurons of three different types.

Figure 10a: Geometrical situation in the feature space \mathbf{P}^2 that corresponds to Case 1 in training the RCE network. **Figure 10b** corresponds to Case 2.

Figure 11a: Geometrical situation in the vector space \mathbf{P}^2 that corresponds to Case 2, when the radius of the sphere of influence of the new neuron centered on \mathbf{X} has its radius set to R_{\max} . **Figure 11b:** Also in Case 2, but when that radius will be set to the distance to the center of the closest neuron of a category other than that of \mathbf{X} .

Figure 12a: The geometrical situation where the training vector \mathbf{X} lies in a region in which the influence fields of RCE neurons with different categories overlap.

Figure 12b: The final situation produced by the training algorithm.

Figure 13a: The geometrical situation in which the feature vector \mathbf{X} lies only in the influence fields of RCE neurons with the wrong categories. **Figure 13b:** The final configuration produced by the training algorithm.

Figure 14: Images of aircrafts and flares on which the RCE neural network made mistakes.

Sub-figures in Figure 14

- (a) Aircraft mistaken for a flare.
- (b) Aircraft mistaken for a flare.
- (c) Aircraft mistaken for a flare.
- (d) Flare to the right mistaken for an aircraft. Aircraft at bottom mistaken for a flare.
- (e) Flare under aircraft mistaken for aircraft exhaust. Top aircraft recognized.
- (f) Flare mistaken for aircraft exhaust.
- (g) Flare at bottom of picture mistaken for an aircraft. Top aircraft recognized.
- (h) Flare mistaken for an aircraft

TABLE LEGENDS

Table 1: The first column is the generation number, the second column is the method of detection of the signal, the third column is the type of IR detector used, and the fourth column gives examples of missiles using the corresponding technology. The last column gives a reference to that technology.

Table 4: The numbers in parenthesis is an identification number we use for that characteristic.

Table 5: Identification errors made by the ZISC neural network in the 10 experiments required for its 10-fold cross-validation. The left-most column shows the number of the experiment. The following three columns show the errors made by the neural network in identifying aircraft blobs: an error is of type F if a blob was falsely identified as a flare, of type "?" if the network did not recognize the blob as belonging to the two categories of objects it knows, and it is of type AC&F if both an aircraft and a flare neuron were activated in the network. The following three columns represent the corresponding results for the flare blobs. The last column to the right shows the total number of errors made in the corresponding experiment.

Table 6: Two versions of the confusion matrix for our tests are shown in the white areas of this table. The one on the left-hand side shows the number of patterns that the ZISC classified as aircraft, flare, unknown, and ambiguous. The one on the right-hand side shows the corresponding percentages of patterns put in

each class. The first column in each confusion matrix corresponds to feature vectors for aircrafts and the second column to feature vectors for flares.