5th Int Conf Remote Sensing Conf
San Francisco, Sept 17-20
2001

TARGET CUEING IN MULTI-SOURCE IMAGERY*

D M Booth, A L Reno, K Benson, P Harvey, P G Ducksbury
Defence Evaluation and Research Agency
Malvern, Worcestershire, England.

ABSTRACT

The enormous volume of imagery that will be generated by future reconnaissance systems suggests that analysts will require computer assistance in the form of cueing aids if effective exploitation is to be achieved. The necessity to adopt a multi-sensor, multi-algorithmic approach is apparent given the potential confusion between target and background i.e. a cluttered environment within which targets may be partially occluded or camouflaged, and where target and background are subject to numerous sources of environmental variation and viewing inconsistencies. This paper outlines some of the algorithms being developed as part of an all-source cueing system, particularly the fusion and architectural elements that are the basis for robust performance.

1.0 INTRODUCTION

The volume of imagery generated by modern reconnaissance sensor systems can be enormous, and, within operational time constraints, manual exploitation is often not feasible. The Joint Services Interoperable Exploitation System programme at DERA Malvern is attempting to address this problem by developing tools for analysts, most significantly for cueing potential targets (Booth et al, 1999).

In a complex, cluttered environment, where targets may be partially occluded or camouflaged, and where target and background are subject to numerous sources of environmental and viewing inconsistencies, the benefits of adopting a multi-sensor, multi-algorithmic approach are widely accepted. This paper focuses on several fusion and architectural aspects of the aided target detection work that reflect this.

The first is the application of evolutionary programming to generate high performance, compact discriminating functions, but more specifically, to generate vision architectures in which the flow of control is determined dynamically based on the decisions reached by particular algorithms and the uncertainties of those decisions. This is achieved by the co-evolution of Finite State Machines with embedded Genetic Programs (GP).

Many low level feature maps, which individually provide very little useful discriminatory information, can, when taken collectively, provide valuable cues. This is demonstrated by two detection / recognition algorithms. In the first approach, the low-level feature maps provide the inputs to an evolutionary architecture in which a hierarchy of fully recurrent neural networks are evolved, with the more architecturally complex and functionally sophisticated networks lying towards the top. The second adopts a Bayesian (flexible model) framework which, in addition to various feature maps, draws on models of both global and local shape characteristics to recognise (possibly incomplete) objects with anticipated modes of shape variation.

Finally, we examine the application of possibility and probability theory for fusing evidence from individual vehicle detections with contextual evidence provided by prescribed vehicle formations.

2.0 EVOLUTIONARY COMPUTING

Biological evolution can be modelled as a repeating two-stage process: selection followed by random variation. Selection is based on the behavioural response of an individual to its environment. While some individuals survive and flourish, others who exhibit behaviours that are poorly suited to the environment may perish i.e. the fittest survive. Random variation occurs during the reproduction cycle of the survivors. They reproduce, either sexually or asexually, and pass on the genetic material responsible for their behavioural traits. No individual is a perfect copy of its parent(s),

---

* Presented at the Fifth International Airborne Remote Sensing Conference, San Francisco, California, 17-20 September 2001

since individual genotypes are subject to mutation. This random variation leads to new behaviours that may, or may not be better suited to the environment. The evolutionary cycle is then repeated.

Evolutionary Computing deals with optimisation and search strategies that mimic biological evolution. These are collectively known as Evolutionary Algorithms (EAs). In the aided target detection (ATD) domain, an EA searches the space of ATD algorithms (or filters) for a high performance algorithm, where performance is measured against a training set (the environment) using a predefined metric based on detection and false alarm rates. The search begins with the creation of a population of 'random' candidate algorithms. Given their origins, the performances of these initial algorithms will be poor, however, some will be better than others. x% of the best performing algorithms are selected to be the parents of the next generation. These are then randomly mutated to produce children that replace the poorest (100-x)% of the population. This process is repeated over many generations resulting in a population of high performance algorithms.

Evolutionary Computing is being examined in two areas of the project: the construction of ATD filters / algorithms and the phyletic evolution of neural networks.

2.1 AUTOMATIC GENERATION OF ATD ALGORITHMS

Evolutionary computing offers the potential to generate high performance, compact functions, the operation of which is evident from their structure. Depending on the level of abstraction of its inputs, the operators used, and a suitable cost function, evolutionary programming could fulfil a number of roles. It could, for instance, construct functions (or filters) for enhancing or detecting objects in raw imagery, or it could be used to build control systems for complex image interpretation architectures. In both cases, the technique must be used selectively and, if possible, in a constrained manner. Essentially, the technique should not be used to model variability that could be removed or be modelled using principled techniques (such as by using physical models). This functionality can be constructed using genetic programming techniques, a field of evolutionary computing. Genetic Programs are encoded as tree structures in prefix notation. The internal nodes of a GP, known as functions, might be the set of mathematical operators {+, -, *, /}, and the leaf nodes, known as terminals, might take the form of a set of input variables.

The algorithms undergoing evolution consist of a main program and associated functions. To facilitate mutation, the main program is represented as a Finite State Automaton (FSA), with each program function represented by a state. Each state has an imbedded GP which effectively represents a discriminant function. The FSA architecture includes an inter-state logical function that specifies the way in which discriminant functions are combined. A new candidate algorithm is generated by applying one or more of the following mutations: add a state; delete a state; change the start state; change a transition(s); change a state logical function(s); change a GP function(s); change a GP terminal(s); extend a GP sub-tree; and shrink a GP sub-tree. These mutations allow the evolutionary process to design the architecture of the algorithm, a task normally performed by a human programmer.

The terminal set could take many forms, for example, it could be composed of relative pixel locations, functions, or function parameters etc. And the operators that are applied to the terminal set to produce discriminant functions might include mathematical operators +, -, *, and /, functions such as min and max, and shift operators for facilitating the capture of spatial relationships in the image. Each program function may draw upon different input features, thus allowing them to perform different but complementary tasks. The decisions made by these discriminant functions can then be combined logically before arriving at an overall classification. As a result, it is possible to build a complex network of co-operating algorithms, with the scheduling of individual algorithms determined dynamically based on previous results.

The architecture described has been applied to the detection of ships in SAR imagery collected by the European Remote Sensing Satellite (Benson, 2000). At 100m resolution, the problem was essentially blob detection, the ships being bright compact regions on a background exhibiting some structure from sea currents, ship wakes etc, and with the whole image subject to SAR speckle noise. In comparison with other techniques (multi-layer perceptron, Kohonen network and another Genetic Programming approach) this method performed best (Table 1). All of the techniques used similar feature sets (statistics computed over concentric annular filters), although there were differences in other aspects of the techniques, such as their training.

Algorithm	True +ve (targets 129)	False –ve
Kohonen network	101 78%	17
Two stage GP	89 69%	2
FSA/GP	102 79%	2

Table1. Ship detection performance on ERS SAR. Coverage is 100km square with around 2/3rds of the image content being sea.

2.2 PHYLETIC EVOLUTION OF NEURAL NETWORKS

Despite considerable study, the structural complexity and performance of evolutionary algorithms and neural networks has fallen far short of the achievements of the natural systems upon which they are based. This piece of work adopts the premise that a non-trivial system can only evolve in response to a complex incremental 'phylogeny' involving multiple related fitness requirements. Figure 1 shows a 'phyletic' architecture applied to the creation of high level spatial feature detectors. Sub-populations of genetically simple detectors are evolved to perform tractable approximations of part, or all, of a more complex and specific detection task. Restricted migration occurs between each 'micro' population and to a population of genetically more complex detectors that is also being evolved toward the more specific objective function. Genomes representing the neural detectors are allowed to expand and contract incrementally, both in size and complexity. In this way, evolution may progress in stages and, at each, functionality from the previous stage is exploited, compounded and refined. Genes are encoded so as to maximise their relevance between populations enabling crossover to exchange meaningful functionality between genomes. Two further developments included in the work are: a) a flexible architecture for simultaneous input of multi-band, multi-source and pre-processed image layers; and b) the creation of a useful target-specific feature (perpendicularity). In the former, input neurons are given genetically variable x and y offset from a central locating position to allow them to evolve individual input mappings for each image data set. In addition, a z offset was included to allow each neuron to take its input from a range of possible sources, allowing each network to concentrate its efforts on analysing regions of the input feature domain that contain useful discriminant information.

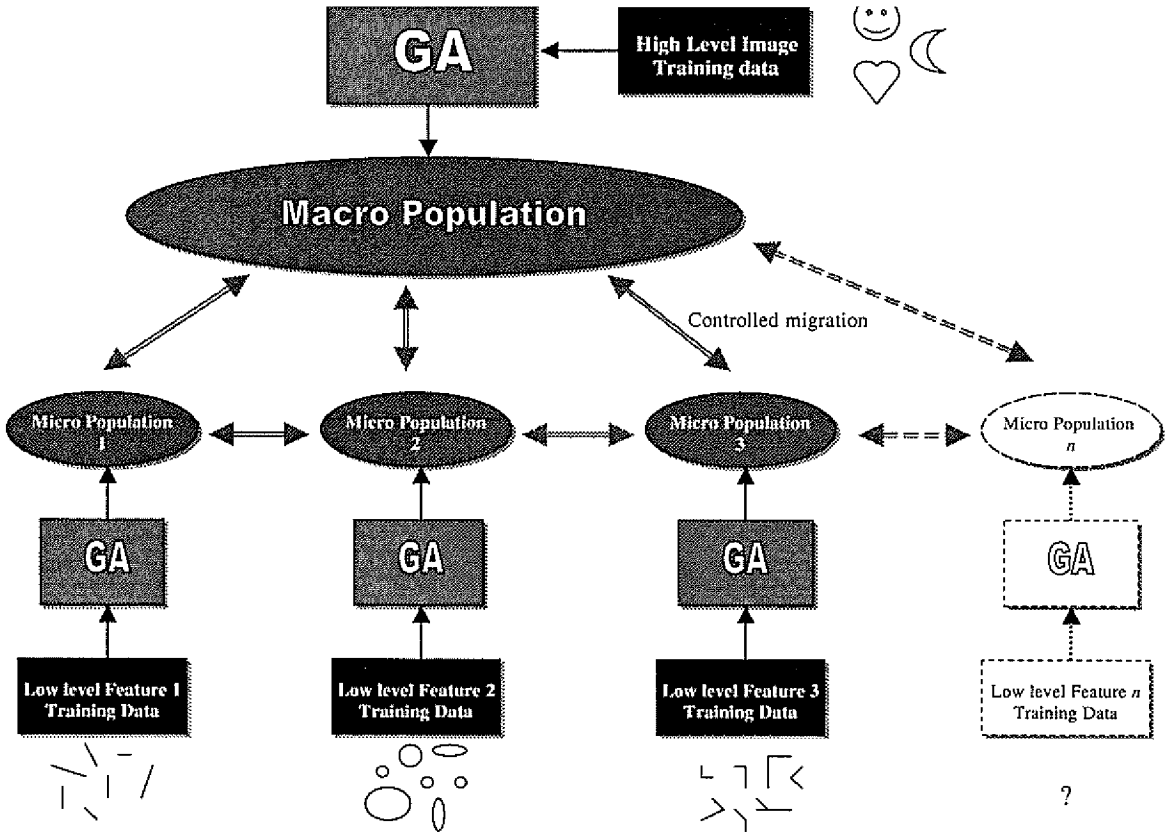


Figure 1. 'Phyletic' Architecture

Input sources might include greyscale images, hyperspectral bands, low-level features (resulting from an image processing or feature extraction algorithm, such as an edge detection operator, applied to the original image) *or even high-level tactical information* regarding the likely presence, and hence identity, of object classes. To aid in the detection of man-made fixed ground structures such as buildings and bridges (as might be required for geo-registration and feature mapping applications) a simple and robust feature was designed that exploits the uniquely "perpendicular" quality of such structures. That is, one of the distinguishing characteristics of a localised man-made ground structure that distinguishes it from nature, is that it usually contains edges that are perpendicular, or nearly perpendicular, to one another. This includes, but is not restricted to *corners* which, on their own, are more likely to be obscured. Our operator takes the output from a Spacek edge angle map and, using the rolling autocorrelation function $\int_0^{\pi} P(\theta)P(\theta+\frac{\pi}{2})d\theta$ computes the maximum local proportion of orthogonal edges (subject to a pre-defined tolerance of angle, edge magnitude and sample size) for every point in the image. This feature works best, for obvious reasons, when the size of the local window is comparable with the required structure. Because it relies on pixel-value statistics, the operator does not localise perfectly but is very powerful for eliminating false alarms caused by extended man-made structures, such as roads, that contain hard but non-orthogonal edges. Figure 2b shows the application of this feature operator on the image shown in Figure 2a. The architecture described previously was used to evolve a 25 node recurrent neural net capable of detecting building structures from a suite of aerial photographs. Each colour image was split into its 3 HSV components and in addition, the V-image plane was processed with a Spacek edge filter and the perpendicularity operator defined above. All 5 input planes were made available to the evolving detectors. Figure 2c shows the resulting best net operating on the image in Figure 2a, blue indicating a true positive.

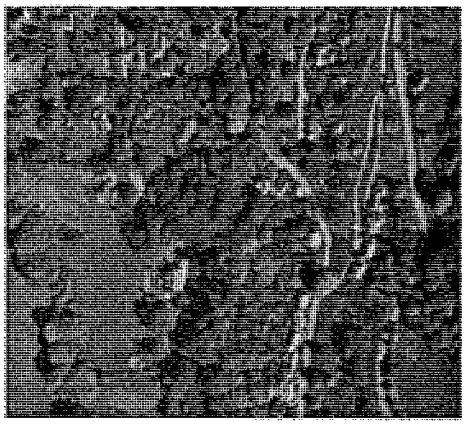


Figure 2a. Aerial image of West Malvern

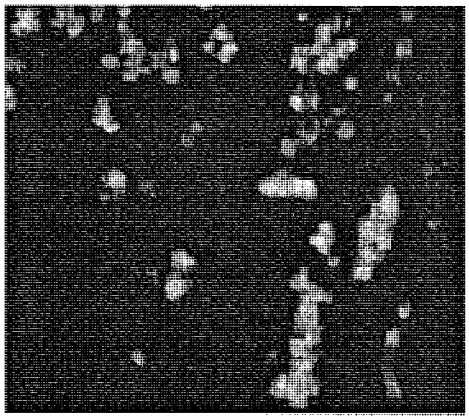


Figure 2b. New 'Perpendicularity' feature To locate building like structures

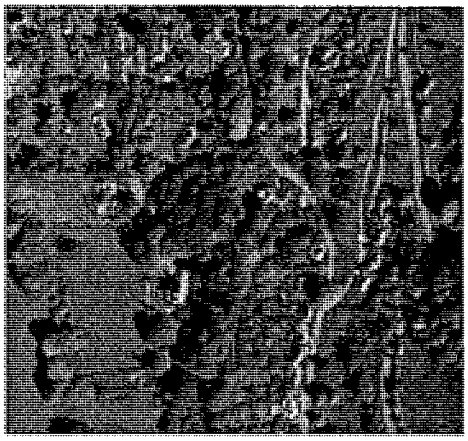


Figure 2c. An evolved building detector applied to Figure 2a

3.0 RECOGNITION

Often, object recognition can be achieved using some prior knowledge of the object's shape together with extracted imagery cues such as edges and regions. However, these cues alone may be insufficient in cluttered environments, particularly when a scene is poorly illuminated, and the object then suffers from poor contrast and definition. Here, object recognition is made more robust by exploiting a wider range of image cues, such as intensity, depth maps, texture, and multispectral components. These are fused with high-level structural knowledge of the object's expected shape. A 2*D* diffused field represents uncertainty and variability in the appearance of an object. This approach makes an important step towards unifying prior knowledge concerning object shape, with attributes derived from the image. The similarities in the representation of these components allow a more principled approach to be taken than existing methods. By applying a transformation to the input data, and to the model, and removing constant components, a closed-form probability density together with the associated drift terms are derived. The transformation effectively reduces the dimension of the problem from 2*D*, to 1*D*, and the resulting density can therefore be optimized very efficiently. The optimum configuration, which gives the most likely object shape, type and pose, is determined with a stochastic Markov Chain Monte Carlo sampler.

3.1 CUES

Cues derived from imagery include depth maps, grey level segmentations and local texture measures. Depth information (in the form of a disparity map) is provided by a region-based correlation algorithm which is applied to two globally registered image frames. This output, which can be thought of as providing an indication of relative height above the ground plane, is quantised using a thresholding with hysteresis procedure. Candidate grey level regions are generated by a simple region-growing segmentation algorithm which assumes pixel grey levels in homogeneous regions are planar. These regions are filtered for size, and basic shape compatibility and assigned a probability according to their crude resemblance of the type of object of interest. Finally, texture is characterised over small $n \times n$ windows using the grey level co-occurrence statistics: entropy; energy; and homogeneity.

3.2 FUSING CUES WITH SHAPE

The above cues are fused with shape information using Equation 1,

$$\log \pi(\mathbf{x}) = \oint_{\Gamma} \left(\int_0^{y(u)} \log \frac{\mathcal{G}_{\text{ref}}(\mathbf{v})}{\overline{\mathcal{G}}_{\text{ref}}(\mathbf{v})} dy \right) dx + \oint_{\Gamma'} \left(\int_0^{y(u)} \log \frac{p(\mathbf{I} \mid \alpha)}{p(\mathbf{I} \mid \overline{\alpha})} dy \right) dx. \tag{1}$$

where Γ is a deformable template, Γ' is its projection into the image under an affine transform, and $\mathbf{v}=(x,y)$ is a point. The first term represents the shape constraints, which are generated from prior knowledge of the expected shape of the object. Here $G_{\text{ref}}(\mathbf{v})$ is a Gaussian smoothed image of a binary reference shape, and $\overline{G}_{\text{ref}}(\mathbf{v})$ is a Gaussian smoothed image of the binary complement of this same shape. The second part, the data term, fits the model to the data. The vector $\mathbf{I}$ represents the imagery, and $p(\mathbf{I}|\mathbf{x})$ is given by

$$p(\mathbf{I}|\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|} \exp -\frac{1}{2}(\mathbf{I} - \overline{\mathbf{I}})^T \Sigma^{-1} (\mathbf{I} - \overline{\mathbf{I}}). \tag{2}$$

The covariance matrix, Σ, and mean vector, $\overline{\mathbf{I}}$, are computed over for the interior region when $\mathbf{x} = \alpha$, and over the background region when $\mathbf{x} = \overline{\alpha}$.

Training images are segmented manually into foreground and background, and the cues described earlier are computed for each image. Hence, the mean vector and covariance matrix of the foreground, $\alpha = (\overline{\mathbf{I}}_f, \Sigma_f)$, and background $\overline{\alpha} = (\overline{\mathbf{I}}_b, \Sigma_b)$ can be computed. Given these statistics, objects are detected in a previously unseen test image by computing the cues and applying Equation 3 to give the fusion image.

$$\log\frac{p(\mathbf{I}\mid\alpha)}{p(\mathbf{I}\mid\bar{\alpha})} = \log\sqrt{\frac{|\Sigma_b|}{|\Sigma_f|}} + \frac{1}{2}(\mathbf{I}-\bar{\mathbf{I}}_b)^T\Sigma_b^{-1}(\mathbf{I}-\bar{\mathbf{I}}_b) - \frac{1}{2}(\mathbf{I}-\bar{\mathbf{I}}_f)^T\Sigma_f^{-1}(\mathbf{I}-\bar{\mathbf{I}}_f), \quad (3)$$

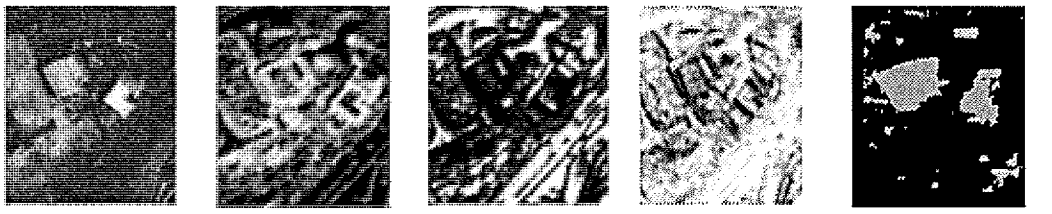
The fusion image is then integrated. This need only be done once, according to (1), and the problem becomes one dimensional. An example fusion image is shown in Figure 3g. Positive parts of this image indicates where the foreground hypothesis is stronger than the background hypothesis.

With this model, the shape component is maximized when the template is aligned with the modal shape defined by the reference model. The data component is maximized when the projection of the template in the image is aligned so that the interior region simultaneously minimizes the Kullback-Leibler divergence between α and the interior region, and maximizes the divergence between $\bar{\alpha}$ (Reno, 1999).

3.3 OPTIMISATION

Using the fusion image, and the shape model, we can estimate the most plausible configuration of the template. We adopt the Metropolis-Hastings (1953) sampling method, using Green's (1995) extension, which allows us to vary the resolution or detail in the template, and we simulate a diffusion component (Grenander and Miller, 1994), by applying jumps at two different scales. Starting from an initial configuration, we apply three different types of movement. Type 1 deforms the template by moving a single vertex. Type 2 modifies the transformation, and type 3 changes the resolution by adding or removing a vertex. Each change is either accepted or rejected, with a given probability.

3.4 RESULTS



(a) Image (b) Texture Entropy (c) Texture Energy (d)Texture Homogeneity (e)Disparity

(f) Intensity Segmentation (g) Fusion Image (h) +ve Likelihood (I) Fusion Image used to Extract Regions (j) Result

Figure 3: Different cues are generated for the image. After training, these can be combined into a fusion image. Positive likelihood regions where the foreground hypothesis is more likely than background.

To demonstrate this method, we took a fairly typical grey-level aerial image and used this to train the system. We computed the five cues, namely texture entropy, energy, homogeneity, disparity and the intensity segmentation. With a mask defining the objects in the scene, we computed the statistics of $\mathbf{I}$, in the foreground and in the background. With these cues, each $\mathbf{I}$ is a vector of five elements. So we compute the two mean vectors $\mathbf{I}_f$, and $\mathbf{I}_b$ each of five

elements, and two covariance matrices, Σ_f and Σ_b each of 5×5 elements. These give our statistics, α and $\bar{\alpha}$ for the foreground and background respectively. Then, with a separate test image (Figure 3a), we compute these five cues, and combine them into a fusion image using Equation 3. When convolved with a Gaussian, this gives the result in Figure 3g. Since the log ratio (Equation 3) is negative when I is more likely to be background, and positive otherwise, we show the positive parts of the fusion image (Figure 3h), which show those places where the foreground hypothesis is stronger than the background. The fusion image and the reference shape model are input to the optimisation process, in which we attempt to find the best candidate solution matching Equation 1. The template outlines the most plausible region, according to its shape, and its placement with respect to the fusion image. The models used to extract the two buildings in this example were hand-drawn. In both cases, the optimisation executed twenty thousand iterations. The result shows the most likely configuration. This result is overlaid on the original image Figure 3j, to show the accuracy.

4.0 MULTILAYOUT ANALYSIS

This section describes work initiated by Nifle (1998) who describes a hybrid fusion approach for the fusion of location information and object identification. Vehicle formations can be identified by combining evidence of individual vehicle locations and classifications with prior knowledge of their expected spatial context (eg formations such as convoys etc). Vehicle positions and classifications are provided by detection and 3D model matching algorithms described by Booth et al (1999). Their output includes: vehicle class together with an associated confusion matrix; target location with positional error estimates represented by a 2D circular Gaussian probability distribution; target orientation estimate with an angle estimation error represented by a Gaussian probability distribution orientated on the previously estimated direction (with a probabilistic error). This information is then passed to the multilayout analysis package that employs higher level reasoning techniques to identify individual layouts/formations. Possibility theory is used when information is less certain and more subjective, e.g. when dealing with prior knowledge of spatial relationships[†], whilst probability theory is more suitable for manipulating classification evidence.

Fuzzy logic is of particular interest here because it allows reasoning with uncertain information. A fuzzy set is represented mathematically by a membership function, μ, that is defined upon a frame of discernment denoted E. This concept enables the management of classes for which the limits are inaccurate. For example, "*about 1.80m tall*" can be represented by the following function (Figure 4).

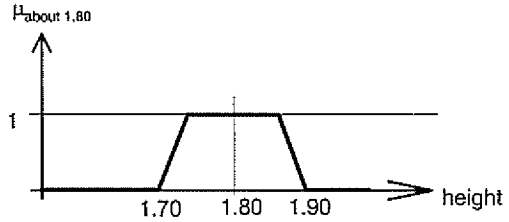


Figure 4 - Membership function

A man who is 1.73m tall belongs to this fuzzy set with a degree of $\mu_{about\ 1.8}(1.73)$. Here, the issue is how much 1.73m is "*about 1.80m*".

Figure 5 shows some typical layouts: tracked artillery; non-tracked artillery; and an ambulance exchange point. For each layout there is a set of constraints governing the positioning of the vehicles with respect to one another, this being illustrated for the second example[‡] (Figure 6).

The system can base its decisions purely on the geometrical context of the candidate targets, however, a number of the layouts do have similar geometrical structures. Confusion may arise, for example, between ambulance exchange points and ammunition dumps. However, to illustrate the modeling of context only, Figure 7 shows the detection rates obtained for a simple layout containing two multi-launch rocket systems (MLRS). Starting off with two MLRS vehicles a set distance apart (*left and right side of the graph*) and moving them closer together, the vertical lines simply delimit the transition points on the graph. The constraints for this layout are that the two vehicles have a set

---

† It may be that a certain event is possible but we are unable to assign a probability to it.

‡ This introduces concepts such as 'Near', 'Behind', 'To the left of' etc

distance of between 500 and 1000 metres apart[§], outside of this the layout starts to become infeasible. The line of the graph illustrates the characteristic shape of the possibility distribution Π as described by Nifle (1998). The vehicles move from being in an 'impossible' state into one that is 'possible' and then back into an 'impossible' one as they become to close together (*i.e. closer than 500 metres*).

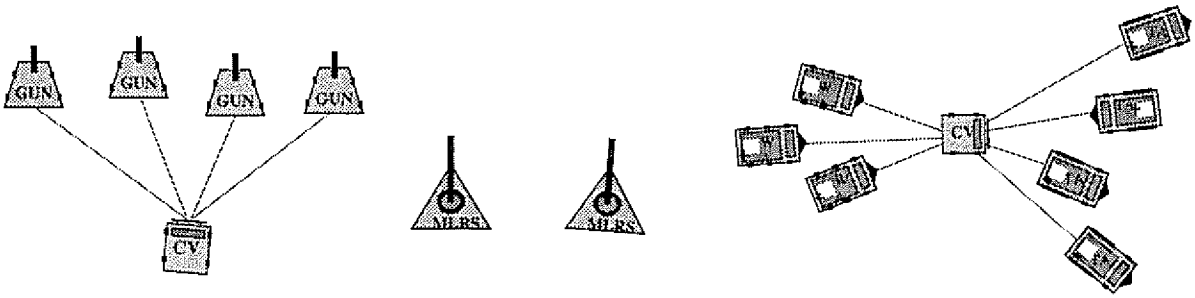


Figure 5. Tracked artillery, non-tracked artillery and ambulance exchange point

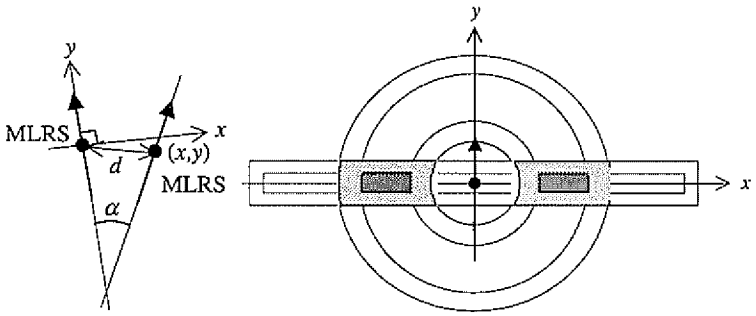


Figure 6. Configuration constraints for tracked artillery

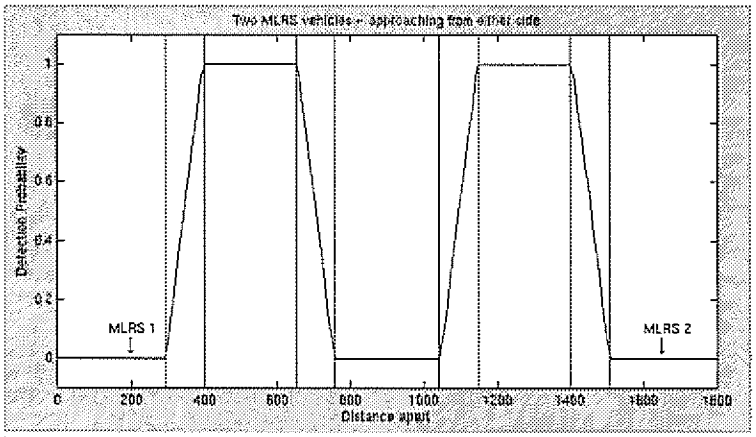


Figure 7, Layout 1 detection, default values (using IR probability matrix)

5.0 CONCLUSIONS

This paper has summarised some of the fusion and architectural aspects of a project dealing with the detection of features of interest in all-source imagery. The underlying theme is to exploit domain knowledge and complementary evidence from multiple sources (imagery and algorithms) to achieve robust performance. Domain knowledge, in the form of vehicle formations, provides some weak constraints that ultimately assist in the recognition of individual

§ In fact the maximum detection probability of 1.0 is achieve precisely between these distances.

vehicles, and hence into the compilation of an overall picture. The other detection and recognition techniques that have been described aim to exploit any useful evidence that is available, usually in the form of feature maps. This work has demonstrated that, in some cases, relatively crude algorithms can provide valuable evidence, and it is often more beneficial to draw on a wide ranging but crude inputs and their fusion than it is to devote attention to perfecting single algorithms whose utility is limited by the operating conditions etc. We have found this to be particularly true of depth maps.

6.0 REFERENCES

Benson, K. (2000) "Evolving automatic target detection algorithms that logically combine decision spaces," *Proc. British Machine Vision Conference*, p385-694, Bristol, U.K.

Booth, D.M., et al (1999) "Computer aided vehicle detection and recognition in multi-source recce imagery," *Proc. 4th Int. Recce Conf. / Canadian Remote Sensing Conf.*, Ottawa, Canada.

Green, P.J. (1995) "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *BIOMETRIKA*, 82(4):711-732, 1995.

Grenander, U. and M. Miller. Representations of knowledge in complex systems (with discussion). *Journal of the Royal Statistical Society B*, 56(4):459-603, 1994.

Harvey, P. and Boyce, J. (2000) 'Phyletic evolution of neural feature detectors,' *Proc. Of the IEEE Congress on Evolutionary Computation 2000*, July, San Diego, U.S.

Metropolis, N. A. Rosenbluth, W. Rosenbluth, M. Teller and E. Teller. (1953) "Equation of state calculations by fast computing machine," *Journal of Chemical Physics*, 21:1087-1091,1953.

Nifle A. et al, (1998) "Multi-layout Identification", *Proc. Eurofusion 98*, Malvern, Oct 1998.

Reno, A.L. (1999) "Statistical region measures for the separation of figure from ground," *Proc. IEEE Conf. on Electronics, Circuits and Systems*, Cyprus, 1999.