DATA FUSION FOR LONG RANGE TARGET ACQUISITION

Patrick Verlinde, Dirk Borghys, Christiaan Perneel, Marc Acheroy
Signal and Image Centre
Royal Military Academy
Renaissancelaan 30,.
B1000 Brussels
Belgium

SUMMARY

An approach to the long range automatic detection of vehicles, using multi-sensor image sequences is presented. The algorithm we use was tested on a database of six sequences, acquired under diverse operational conditions. The vehicles in the sequences can be either moving or stationary. The sensors are stationary, but can perform a pan/tilt operation. The presented paradigm uses data fusion methods at four different levels (feature level, sensor level, temporal level and decision level) and consists of two parts.
The first part detects targets in individual images using a semi-supervised approach. For each type of sensor a training image is chosen. On this training image the target position is indicated. Textural features are calculated at each pixel of this image. Feature level fusion is used to combine the different features in order to find an optimal discrimination between target and non-target pixels for this training image. Because the features are closely linked to the physical properties of the sensors, the same combination of features also gives good results on the test images, which are formed of the remainder of the database sequences. By applying feature level fusion, a new image is created in which the local maxima correspond to probable target positions. These images coming from the different sensors are then combined in a multi-sensor image using sensor fusion. The local maxima in this multi-sensor image are detected using morphological operators. Any available prior knowledge about possible target size and aspect ratio is incorporated using a region growing procedure around the local maxima. A variation to this approach, that will also be developed in this paper, combines the previous feature and sensor level fusion, by extracting the features in each sensor as before but using the feature level fusion directly on the combination of all features from all sensors in what is sometimes called a « super feature vector ». Tracking is used in both cases to reduce the false alarm rate.
The second part of the algorithm detects moving targets. First any motion of the sensor itself needs to be detected. This detection is based on a comparison between the spatial co-occurrence matrix within one single image and the temporal co-occurrence matrix between successive images in a sequence. If sensor motion is detected it is estimated using a correlation-based technique. This motion estimate is used to warp past images onto the current one. Temporal fusion is used to detect moving targets in the new sub-sequence of warped images. Temporal and spatial consistency are used to reduce the false alarm rate.
For each sensor, the two parts of the algorithm each behave as an expert, indicating the possible presence of a target. The final result is obtained by using decision fusion methods in order to combine the decisions of the different experts. Several « k out of n » decision fusion methods are compared and the results evaluated on the basis of the 6 multi-sensor sequences.

1 INTRODUCTION

Long range automatic detection of vehicles is of great military importance to modern armed forces. The most critical factor of any system for automatic detection is its ability to find an acceptable compromise between the probability of detection (= 1 - probability of a miss) and the number of false alarms. This is the classical trade-off one finds in binary hypothesis testing between the two types of error one can make : the false rejection (FR : which corresponds here to a miss : there is a target, but it has not been found) and the false acceptance (FA : which is in this case the same as a false alarm : there is no target, but the system thinks there is one). In a single sensor detection system it is well known that if one reduces one type of error, the other type of error automatically increases. A possible way-out of this deadlock is to use more than one sensor and to combine the information coming from these different « experts ». This combination or (data) fusion can be done on different levels. In this paper, only the (common) case of a centralised fusion processor with all its sensors connected in parallel will be considered.
In the specific data fusion literature [1-5] one often distinguishes between the following (or equivalent) fusion levels : low level fusion (also called score or measurement level fusion), medium level fusion (which includes feature and sensor level fusion), high level fusion (also called decision level fusion) and temporal level fusion. As can be expected, in real (-time) applications, there is a trade-off to be made between the amount of information that can be combined and the bandwidth necessary to communicate all this information to the centralised fusion processor. The lower the level of fusion, the more information is available to be combined, but the larger becomes the bandwidth necessary to communicate with the centralised fusion processor (or for a fixed bandwidth, the slower becomes the fusion process). Vice versa one sees that when the level of fusion gets higher, the available information diminishes, but so does the necessary bandwidth. Furthermore not all data fusion levels are always applicable. For instance, if low level fusion is going to be used, care must be taken to combine only similar entities (scores, measurement results,...). It is therefore impossible to use low level fusion to combine the raw results coming from two (or more) totally different sensors (e.g. an imaging sensor and a range finder). But this constraint doesn't exist any longer on the decision level, where each sensor is considered as a separate « expert », who decides on his own. In the special case of target detection where the « hard » binary decision rule is

used (the « hard » decision is indeed bi-valued : target present (1) or not (0)), the central fusion processor contents itself to combine only the decisions (the 1's and the 0's) coming from different sensors, without considering the type of sensor. As a general conclusion concerning the different data fusion levels, one can state that all different fusion levels have their importance and their specific applicability domain.

Based on these considerations, we have tried to use data fusion on several levels to try to optimise the use of the available data. That is basically why this paper describes an approach to tackle the previously exposed problem using four different data fusion techniques related to several levels : feature level fusion, sensor level fusion, temporal level fusion and decision level fusion. The only fusion level that is not used in this paper is the low level fusion. This technique (in the form of pixel level fusion) is mainly used in remote sensing applications [6, 7]. In the main approach, we do however use two different medium level fusion techniques. In the following sections the use of these different data fusion techniques will be explained in more detail.

2. IMAGE DATABASE

For the development and testing of the algorithm, a database of 6 multi-spectral image sequences, numbered MS01 to MS06 [1] was compiled. The sequences correspond to two scenarios. The first scenario is a typical surveillance scenario in which the sensor watches a scene and tries to detect targets entering its field of view(FOV). In this scenario, the targets are moving. The second scenario is a reconnaissance scenario in which the sensor is mounted in a new terrain and it tries to detect the presence of vehicles which can now be stationary or moving. During image acquisition the sensor is stationary in both scenarios; it can only perform a pan and tilt operation. The following table presents some properties of the sequences.

Seq nr	Targets	Target Motion	Sc	Type of Sensors
MS01	Helicopter Truck	Across FOV	1	LW,TV
MS02	Truck	Towards Sensor	1	LW,TV
MS03	Helicopter	Across FOV	1	LW,TV
MS04	2 Tanks	None	2	LW,SW,TV
MS05	2 Tanks +Camoufl.	None	2	LW,SW,TV
MS06	Helicopter	Across FOV	1	LW,R,G,B

Table 1: Properties of sequences.

In the table the Sc column presents the scenario to which the sequence corresponds. In the sensor column the following abbreviations are used: LW and SW denote long-wave and short-wave infrared respectively. TV is B/W visual images. R,G,B are the components of a colour visual image. Each set of three subsequent sequences were acquired by the same sensor set. Sequence MS05 is the same as sequence MS04 except for the fact that in MS05 the targets are camouflaged.

[1] MS01-MS03: Courtesy of Defense Research Establishment Valcartier, Quebec, Canada ; MS04-MS05: Courtesy of Naval Air Warfare Center, China Lake, US; MS06: Courtesy of ASIAT-DTT, Peutie, Belgium

3. OVERVIEW OF THE APPROACH

The proposed algorithm consists of two independent parts. The first part searches for targets in single images while the second part uses multiple subsequent images in order to specifically find moving targets.

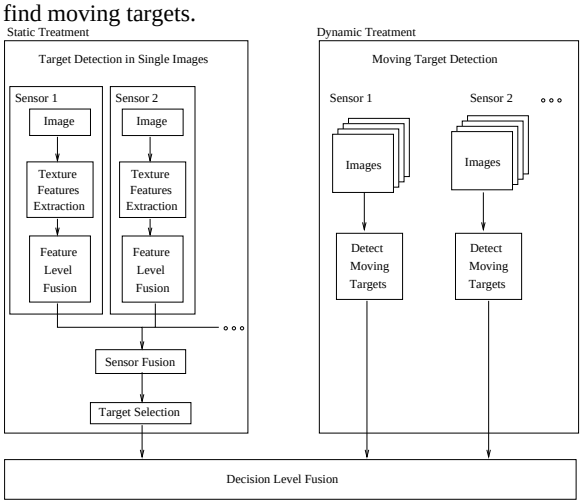


Figure 1: Global Overview of the method

For the first part of the algorithm we have chosen a semi-supervised approach based on texture feature extraction. Allthough we are not interested in explicitly modelling or measuring texture, these texture features are interesting because they are independent measurements of the local spatial distribution of grey values within an image and it is likely that some of these parameters will highlight the difference between targets and background. The texture parameters are even more appealing because it can easily be seen that features that are classically used for target detection such as intensity and gradient are just special cases of these texture parameters.
Feature level fusion is used to combine the texture features from each image into a new image in which the grey value at each pixel is proportional to the probability that the pixel belongs to the target. These images from the different sensors are fused in a sensor fusion step.
The second part of the algorithm detects moving targets in sub-sequences from each sensor separately.
Each part of the algorithm behaves as an expert indicating the possible presence of vehicles in the scene. Decision fusion is used to combine the outcomes from all experts.

4. TARGET DETECTION IN SINGLE IMAGES (TDSI Module)

Introduction

For the detection of targets in single images, a semi-supervised approach based on texture features was chosen. For each sensor type, one image was selected to constitute the learning database. On these images the true targets were delimited. Then several texture parameters were calculated at each pixel of these learning images and logistic regression [8] was used to find a combination of the texture parameters that is proportional to the probability of finding a target at the corresponding image location.

The actual detection algorithm then applies the same combination to the texture features calculated on the remainder of the image database (test images). When this function is applied to the texture features calculated at each pixel of a test image, a new image, called feature-level-fused image, can be formed in which the maxima correspond to likely target positions. These feature-level-fused images, obtained from all the different sensors, are then fused again in a subsequent sensor fusion step.

To find the possible target positions, first the local maxima are determined in this sensor-fused image and then available prior knowledge about possible target size and aspect ratio is used to reject false targets.

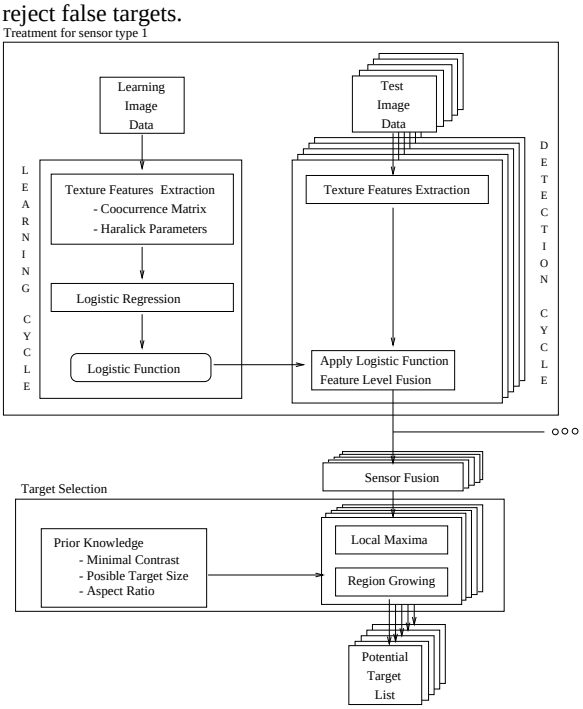


Figure 2: TDSI Module

Texture Parameters

The calculation of the texture features is based on the co-occurrence matrix. The co-occurrence matrix is defined as a function of a given direction and distance, or alternatively, as a function of a displacement (dx,dy) along the x and y direction in the image. For a given displacement (dx,dy), the (i,j) element of the co-occurrence matrix is the number of times the grey value G at the current position (x,y) is i when the value at the distant position (x+dx,y+dy) is j.

$$C^{dx,dy}(i,j) = P\left(G(x,y) = i \;\middle|\; G(x+dx, y+dy) = j\right)$$

The co-occurrence matrix can be calculated on the whole image. However, by calculating it in a small window scanning the image, a co-occurrence matrix can be associated with each image position. The centre of the window is denoted (x_c, y_c) and the corresponding co-occurrence matrix is $C^{dx,dy}_{x_c,y_c}(i,j)$

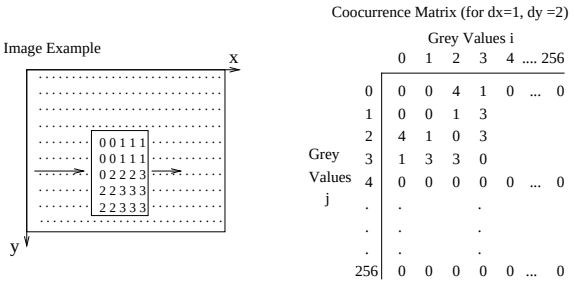


Figure 3: Co-occurrence matrix

In Figure 3 an example of a co-occurrence matrix is shown. The matrix corresponds to the small window of the image on the left and was calculated for a displacement of dx = 1, dy = 2. The textural features that were used, were introduced by Haralick [9-11] and are widely used in texture analysis. Based on the local co-occurrence matrix, the used parameters are defined as follows:

$$F_1(x_c,y_c) = \text{Energy} = \sum_i \sum_j C^{dx,dy}_{x_c,y_c}(i,j)^2$$

$$F_2(x_c,y_c) = \text{Contrast} = \sum_i \sum_j \left[(i-j) C^{dx,dy}_{x_c,y_c}(i,j)\right]$$

$$F_3(x_c,y_c) = \text{Max.Prob.} = \max\left[C^{dx,dy}_{x_c,y_c}(i,j)\right]$$

$$F_4(x_c,y_c) = \text{Entropy} = \sum_i \sum_j C^{dx,dy}_{x_c,y_c}(i,j) \log\left[C^{dx,dy}_{x_c,y_c}(i,j)\right]$$

$$F_5(x_c,y_c) = \text{Homogeneity} = \sum_i \sum_j \frac{\max\left[C^{dx,dy}_{x_c,y_c}(i,j)\right]}{\left[1+(i-j)^2\right]}$$

$$F_6(x_c,y_c) = \text{Variance} = \left[\sum_i (i-E_i)^2 \sum_j C^{dx,dy}_{x_c,y_c}(i,j)\right] \left[\sum_j (j-E_j)^2 \sum_i C^{dx,dy}_{x_c,y_c}(i,j)\right]$$

with $E_i = \sum_i i \sum_j C^{dx,dy}_{x_c,y_c}(i,j)$ and $E_j = \sum_j j \sum_i C^{dx,dy}_{x_c,y_c}(i,j)$

We are not interested in modelling or measuring texture but only in detecting a difference between target and background pixels. The "texture parameters" are only used as features of which we hope that some will highlight the difference between target and background. Because we do not intend to measure the texture within the target, the parameters are useful even for small targets and we can chose an arbitrary displacement (dx = 1, dy =1) for all calculations of the co-occurrence matrix.

The results for each texture feature can be converted into an image. Figure 4 shows the texture images corresponding to the first image set (IR and VIS) of sequence MS01.

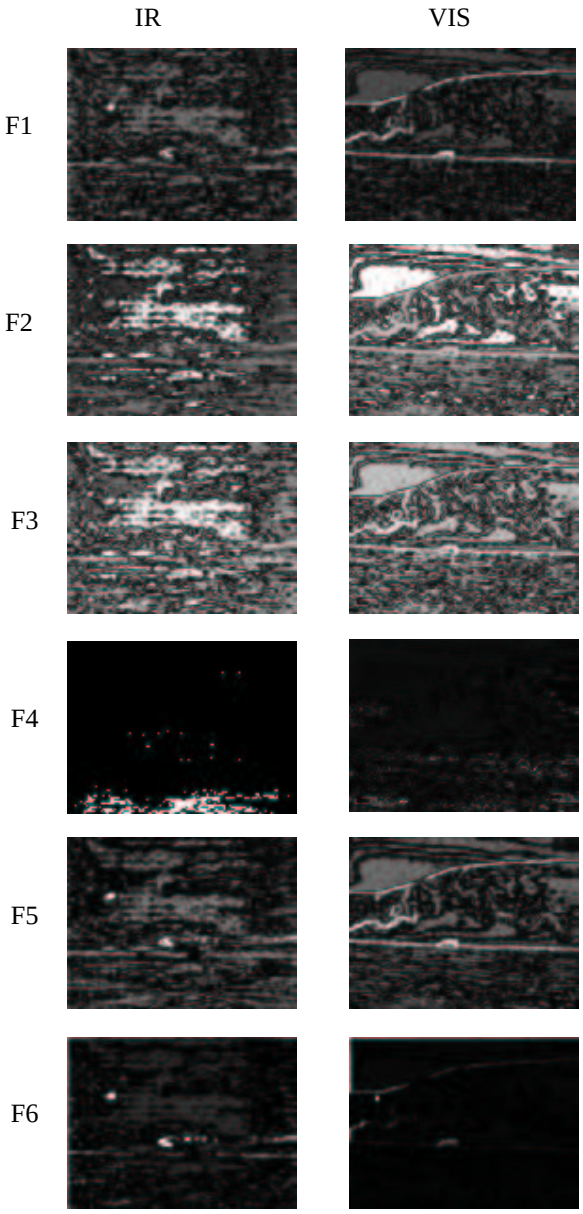


Figure 4: texture images

As can be noticed in Figure 4, the vehicles are clearly visible in some of the texture images. Hence the idea to combine the texture features to get an optimal discrimination between background and targets. When only two classes are involved, as is the case here (targets/ background), logistic regression offers an appropriate approach [8].

In the learning phase, at each pixel of the learning image(s), the texture features are calculated and stored in a table. Then the human operator interactively indicates the bounding rectangles surrounding the targets in the learning image(s) and a column is automatically added to the table assigning each measurement in the table to either class 0 (background), when it corresponds to an image pixel that falls outside the bounding rectangles, or class 1 (targets) when it is inside one of the rectangles. Logistic regression is then used to find a combination of the form :

$$p_{x,y}\left(\text{target}\,\middle|\,\vec{F}\right)=\frac{e^{b_0+\sum_i b_i F_i(x,y)}}{1+e^{b_0+\sum_i b_i F_i(x,y)}}$$

eq 1: Logistic Function

in which $p_{x,y}(\text{target}|\vec{F})$ is the conditional probability that a pixel (x,y) belongs to the class 1 (target class) given the vector of texture parameters $\vec{F}$ at the given pixel. The logistic regression was carried out using Wald's forward method. In this method, at each step, the most discriminant feature is added and the significance of adding it to the model is verified. This means that not all features will necessarily be included into the model.

Feature level fusion

If the learning images are representative for the images of a given sensor type, the most discriminating features for each sensor will have the highest weights b_i. Therefore, when using the same weights to combine the feature images of the remainder of the database into new images using equation eq 1, targets will appear as local maxima. This is the feature level fusion.

Sensor Fusion

The sensor fusion step combines the images obtained by the feature level fusion step. In the feature-level-fused images, for each sensor, targets appear as local maxima. Therefore it is possible to fuse these images by a simple multiplication. In the new images the targets will still appear as local maxima.

Region Growing around Local Maxima

In the sensor-fused image the local maxima will correspond to likely target positions. To detect the targets it is thus necessary to find these local maxima. A region growing procedure around the maxima is then used to incorporate available prior knowledge about possible target size and aspect ratio.

Local Maxima

The detection of local maxima is based on a succession of morphological operations [12, 13]. The basic operator is a dilation with a 2 by 2 structuring element.

Region Growing

To incorporate any available prior knowledge about the possible range of target size or aspect ratio, a region growing procedure is used. The initial regions for the region growing are the local maxima in the image. Surrounding pixels are added to these regions as long as their grey level differs less than a given threshold from the value at the local maxima. If the region becomes too large it is discarded. If the region growing of a given region stops before it reaches the upper size-limit, the other constraints are checked. If a constraint is not satisfied, the region is discarded.

Clutter rejection

To reduce the number of false targets, a simple clutter rejection scheme was implemented. A target is only declared if it was present in at least 7 out of 10 preceding images. More clever tracking methods [14, 15] could be used, but because our main interest is the exploration of data fusion methods, we did not implement this yet.

Modes of operation

The presented approach for the detection of targets in single images allows us to experiment with different levels of fusion. The target detection can be performed on the feature-level-fused images of each separate sensor (mode M1).
The second mode (M2) combines the feature-level-fused images from all sensors using the sensor fusion step described above.
In the third mode (M3), the logistic regression is applied to a superset of texture features, i.e. the feature-level-fused image is obtained by applying the logistic function to the set of features obtained from all sensors. This is only possible if the images from all sensors are registered.
A subdivision of modes 1 and 2 can be made according to whether the learning images used were acquired from the same sensor (i.e. the LW, SW and TV sensors for sequences MS04-05) or from the same generic class of sensors (i.e. using images from the LW and TV sensor of sequence MS01 to yield the weights for respectively all infrared-like sensors and all visual sensors).

5. MOVING TARGET DETECTION (MTD Module)

The second part of the algorithm focusses on the detection of moving targets. In order to detect moving targets, any sensor motion needs to be detected and its effects compensated first. Then, in a temporal fusion step, preceding images can be warped onto the current one. Moving objects will appear as a difference between the original image and the warped ones.

Detection of sensor motion

The detection of sensor motion is again based on co-occurrence matrices. This time the co-occurrence matrix is calculated between an image and the preceding one (temporal co-occurrence matrix).
$C_{x,y}^{dx,dy,dt}(i,j) = P\left(G(x,y,t) = i \mid G(x+dx, y+dy, t+dt) = j\right)$ If no sensor motion occurred between the two images, ideally, for dx=dy=0 (i.e. no spatial displacement), all non-zero elements of the temporal co-occurrence matrix should lie on the diagonal. However, due to noise, there will be a small spread along the diagonal. If one calculates the spatial co-occurrence matrix for a small displacement, the spread along the diagonal is due to noise and to the fact that the image is not homogeneous.Therefore, when comparing this spatial co-occurrence matrix with the temporal co-occurrence matrix, the spread along the diagonal is expected to be the largest in the former one if no motion occurred between the two images that were used to calculate the temporal co-occurrence matrix. When motion is present, the spread along the diagonal quickly becomes larger. The measurement we used to detect sensor motion is based on the percentage of off-diagonal points in both co-occurrence matrices:

$$MC = \frac{\sum_j \sum_{i \neq j} C(i,j)}{\sum_j \sum_i C(i,j)}$$

This is calculated for both the temporal MC_{temp} and for the spatial co-occurrence matrix MC_{spat}. Sensor motion is said to be present if

$$\frac{MC_{temp} - MC_{spat}}{MC_{spat}} \geq 0.005$$

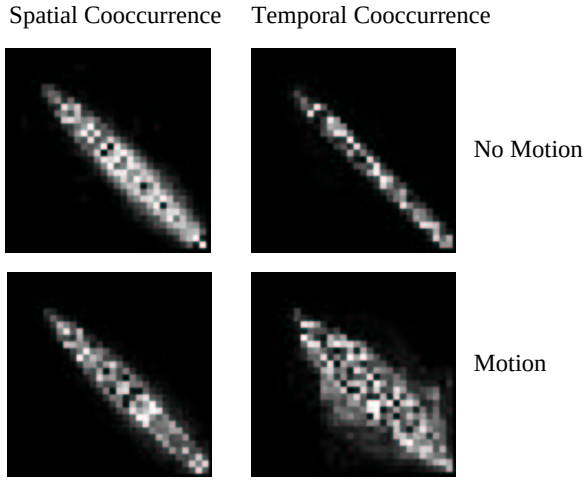


Figure 5: Detection of sensor motion

In Figure 5 the spatial and temporal co-occurrence matrix are shown. The upper images show the matrices for a part of a sequence where no sensor motion was present. The lower images show an example of both matrices calculated in a part of the same sequence where the sensor was moving.

Motion Estimation

If sensor motion is detected, we need to estimate it and compensate its effects on the images. Because it is known that the sensor is stationary and can only do a pan or tilt, the corresponding motion in the image will consist of a uniform translation. The motion is estimated by searching for the translation that optimises the correlation for a few horizontal and vertical lines. If the sensor is mounted in a moving vehicle or no apriori knowledge about the type of sensor motion is known, methods based on the model of a moving rigid planar patch [16] or optical flow techniques can be used [17].

Detection of moving targets

Once the sensor motion is estimated, preceding images are warped onto the current one. Then the original image is subtracted from the warped ones. If a moving object is present in the scene, we should find a large value at its position. The resulting images after subtraction are therefore thresholded and objects with acceptable size and aspect ratio are selected using a region growing procedure. Tracking is used to get the target list. Figure 6 shows the result of subtracting the original image from the warped ones.

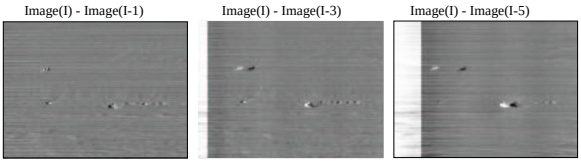


Figure 6: Detection of Moving Targets

6. DECISION FUSION

The two parts of the algorithm each behave as experts indicating the possible position of targets in the scene. The final decision is reached by fusing the results of these experts. Because each expert only provides a binary decision - i.e. either a target is present or it is not - the decision fusion is implemented as a weighted ''k out of N" voting-rule [1, 5]. The weights attributed to the decisions of each expert can depend upon several considerations.
For the detection of moving targets, each single sensor acts as an expert. For the detection of targets in single images, the decision was made after fusing the feature images from all sensors in Modes M2 and M3. In Mode M1, results from each separate sensor are passed to the decison level fusion step. The weights to be attributed to the decision of each expert will need to be adapted accordingly.
The weights may also depend on the type of scenario. In the surveillance scenario, the primary expert is the motion detector, whereas in the second scenario (reconnaissance), both types of expert are equivalent.

7. RESULTS AND DISCUSSION

In this section the results for the two parts of the algorithm are presented first. Then some results of decision fusion are presented.

Results of TDSI Module

Results of the feature-level fusion

For the feature level fusion a learning image set needs to be identified. For the so-called "generic sensor case"(GSC), the learning image set consists of the first Infrared (LW) and Visual image of MS01 on which the targets appear.
For the "sensor kind case"(SKC) three sets of learning images were identified. For sequences MS01-MS03 they are the same as the ones used in the "generic case". For sequences MS04-MS05 the first multi-sensor image set of MS04 is used, yielding weights for LW-,SW- and TV-type sensors. For sequences MS06 the first image set of MS06 is used. For the fusion of the superset of features (SSF) the learning images were the same as for the "sensor kind case". The following table presents the weights obtained for MS04-MS05.

Case	Sens	b0	b1	b2	b3	b4	b5	b6
GSC	LW SW	-15	2.4	0	0	4.7	0	0.01
	TV	-18	0	0	4.7	5.6	2.6	0
SKC	LW	-18	0	0.4	0	6.3	4.8	-0.008
	SW	-18	0	0	0	7.0	10	-0.06
	TV	-19	0	-0.06	4.6	7.6	3.5	-0.004
SSF	LW	-31	4.9	0	0	8.9	0	0
	SW	0	0	0	0	0	0	-0.05
	TV	0	0	0	0	6.8	8.8	-0.06

Table 2: Weights obtained by logistic regression for sequences MS04 and MS05.

Results for the single sensors (M1)

For each sequence, the probability of detection (Pd) and the average number of false targets per image (Nft) is given for both the "Generic Sensor Case" and the "Sensor Kind Case". Please note that for the first three sequences, the two cases are identical because the "generic sensor set" is the set of sensors that were used to acquire these sequences. For sequences MS04 - MS06 the results obtained for the SKC case are slightly better than those obtained for the GSC in most cases. However, the inverse is true for MS06LW. This is due to the choice of the learning image in the SKC case. The learning image is the first image on which the target appears. In the infrared image, the target happens to be white on a light grey background (clouds) and is very difficult to see. Because, in a part of the sequence, its background becomes a clear sky (dark), the weights are no longer appropriate and performance drops.

Sequence	Sensor	GSC		SKC	
		Pd	Nft	Pd	Nft
MS01	LW	85	10	85	10
	TV	32	4	32	4
MS02	LW	0	2	0	2
	TV	0.5	1.4	0.5	1.4
MS03	LW	84	11	84	11
	TV	21	16	21	16
MS04	LW	97	16	***98***	***13***
	SW	43	2	63	6
	TV	98	14	***93***	***13***
MS05	LW	81	12	***99***	***14***
	SW	0	1.16	9	4
	TV	51	14	***98***	***11***
MS06	LW	50	27	31	7
	RD	89	6	94	2.5
	GR	***96***	***0.4***	73	2.4
	BL	90	0.25	82	0.57

Table 3: Results for single sensors

In MS02 targets disappear in the large number of false targets caused by noise. They are only sporadically detected by the first part of the algorithm and rejected by the clutter rejection stage.

Results after sensor fusion (M2)

The following table presents the results after the sensor fusion step.

Sequence	GSC		SKC	
Sequence	Pd	Nft	Pd	Nft
MS01	83	3	83	3
MS02	20	5	20	5
MS03	16	16	16	16
MS04	95	2	***98***	***0.07***
MS05	94	0.2	***96***	***0.15***
MS06	23	7	47	0.8

Table 4: Results after sensor fusion

Results using superset of features (M3)

Sequence	Pd	Nft
MS01	**88**	10
MS02	0.5	1.7
MS03	**93**	31
MS04	0.4	12
MS05	0	20
MS06	40	6

Table 5: Results of superset of features

Results of MTD Module

For the moving target detection a threshold is defined as the number of subsequent images in which the target is detected. The maximal number of subsequent images is 9.
The following tables present the results as a function of the threshold T. Moving targets were only found in sequences MS01-MS03 and in MS06. For MS06 results for the three visual components were very similar, therefore only the red component is shown.

T	MS01				MS02			
	LW		TV		LW		TV	
	Pd	Nft	Pd	Nft	Pd	Nft	Pd	Nft
1	***88***	***0.06***	48	0.87	30	1.6	39	0.48
2	82	0.03	47	0.7	25	0.37	36	0.29
3	63	0.01	45	0.5	19	0.16	32	0.19
4	53	0	44	0.15	12	0.06	23	0.14
5	52	0	44	0.02	9	0.03	17	0.10
6	52	0	42	0.02	4	0.02	9	0.06
7	51	0	41	0	1	0.01	4	0.03
8	0	0	0	0	0	0	0	0

Table 6: Results of MTD for sequences MS01 and MS02

T	MS03				MS06			
	LW		TV		LW		RD	
	Pd	Nft	Pd	Nft	Pd	Nft	Pd	Nft
1	77	0.02	55	0.03	7	1.1	81	0.19
2	74	0.02	44	0.02	4	0.62	81	0.15
3	***73***	***0.01***	38	0.02	2	0.26	***80***	***0.15***
4	73	0.01	24	0.02	0	0.12	75	0.11
5	71	0.01	6	0.01	0	0.04	65	0.09
6	66	0.01	0	0.01	0	0	51	0.06
7	58	0.01	0	0.01	0	0	32	0.04
8	9	0	0	0	0	0	4	0

Table 7: Results of MTD for sequences MS03 and MS06

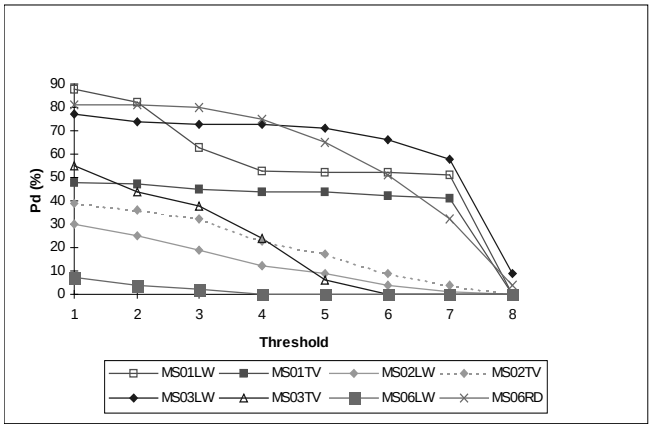


Figure 7: Probability of detection for MTD vs. threshold

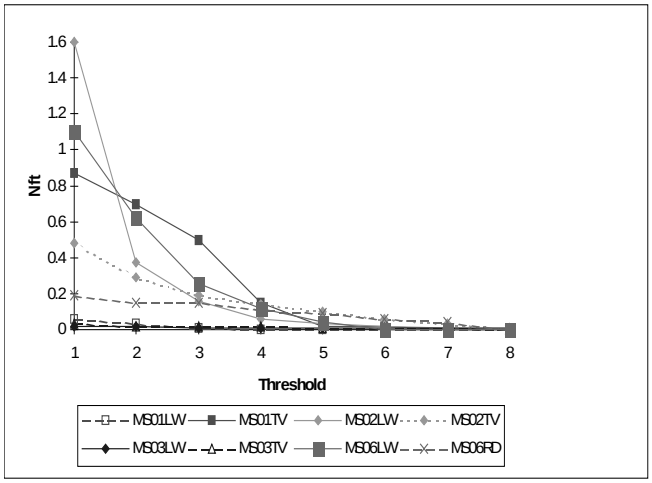


Figure 8: Average number of False Targets per image for MTD vs. threshold

In Figure 7 and Figure 8 it can be noted that by setting the threshold to 3 the number of false targets is greatly reduced while the probability of detection is hardly affected. Therefore we will use the threshold 3 in the decision fusion.
Another remark is that in sequences where moving targets are detected, the MTD module outperforms the TDSI module (cf. Table 3, Table 6 and Table 7)

Results after decision fusion

In sequences corresponding to scenario 1 (surveillance) the motion detector is the primary expert. In the decision fusion we should therefore attribute the highest weight to it.

In the reconnaissance scenario, the motion detector is equivalent to the target detector in single images and both experts should have the same weight.
The "K out of N" decision fusion only accepts a target if the number of experts $N_{experts}$ that have detected it is above a certain threshold *T*. Special cases for this threshold are:

- AND: $T = N_{experts}$
- OR: $T = 1$
- Majority voting: $T = \frac{N_{experts}}{2} + 1$

The following tables present the results for both scenarios at different thresholds. In Table 8 the results for sequences MS01-MS03 are shown after fusion of the outcomes for each individual sensor(mode M1). In Table 9 the results of fusing the sensor-fusion results with the motion data are shown for the same three sequences (M2). Please note that sequences MS01-MS3 were acquired using two sensors, the number of experts is 2 for the TDSI module (in mode M1) and 2 for the MTD module. In the decision fusion both parts of the algorithm have the same weight. In mode M2 only one expert is available for the TDSI module. However, it is counted twice to ensure that both parts of the algorithm have the same weight in the final decision.
Table 10 shows the results of the decision fusion for MS01-MS03 using the super-set of features (mode M3).
Table 11 shows the results of the decision fusion for sequence MS06 using the outcomes of the TDSI module applied to individual sensors (mode M1) and the results using the outcomes of the sensor fusion step. For both modes, the results are given for the "generic sensor case" and the "sensor kind case". Please note that, as in sequence MS06, we have 4 sensors (LW,RD,GR and BL), there are 8 independent experts.
In sequences MS04 and MS05 all targets are stationary and the MTD module does not report any targets. Therefore the decision fusion step is not necessary.

T	MS01		MS02		MS03	
	Pd	Nft	Pd	Nft	Pd	Nft
1	96	20	36	2	***85***	***11***
2	66	3	16	0.7	73	1
3	54	0.86	3	0.24	39	0.5
4	30	0.27	3	0.18	0	0.01

Table 8: MS01-MS03: Results of decision fusion using outcomes from individual sensors (mode M1)

T	MS01		MS02		MS03	
	Pd	Nft	Pd	Nft	Pd	Nft
1	65	3	***54***	***5***	79	20
2	49	0.9	17	0.4	63	0.8
3	3	0.11	3	0.1	19	0.18
4	0	0.01	0.5	0.01	6	0.01

Table 9: MS01-MS03: Results of decision fusion using outcomes from sensor fusion step (mode M2)

T	MS01		MS02		MS03	
	Pd	Nft	Pd	Nft	Pd	Nft
1	***95***	***11***	36	2	90	28
2	58	0.9	16	0.46	67	1.98
3	36	0.31	3	0.09	43	0.53
4	3	0.05	0.5	0.01	7	0.05

Table 10: MS01-MS03: Results of decision fusion using outcomes of the super set of features (mode M3)

T	MS06 (GSC)		MS06 (SKC)		MS06SF (GSC)		MS06SF (SKC)		MS06 (SSF)	
	Pd	Nft	Pd	Nft	Pd	Nft	Pd	Nft	Pd	Nft
1	98	34	98	13	98	0.79	***98***	***0.7***	84	6.5
2	98	3	94	2.5	69	0.59	64	0.3	55	1
3	86	1.4	73	1.1	29	0.41	19	0.3	6	0.2
4	58	0.7	32	0.7	0.9	0.02	0.9	0.1	4	5.9

Table 11: MS06: Results of decision fusion using the GSC and SKC for individual sensors and after sensor fusion and using the super set of features (SSF).

Discussion

For the discussion, the fusion method that gives the best results will be identified for each sequence separately. Then these "best results" will be analysed as a function of some of the properties of the sequences. The notion of best results depends on the application. For some applications the probability of detection is the critical factor while for others the ratio between probability of detection and number of false targets has to be maximised. To identify the "best fusion method" for each sequence, we have chosen the latter criterion (Pd/Nft). In the previous tables, these best result are highlighted with italic letters on a grey background. Please note that, for sequences where moving targets are present, the MTD algorithm used on a single sensor sometimes gives a higher ratio Pd/Nft than any of the fusion results. In all cases fusion will however increase the probability of detection.

Best results per sequence

MS01:
For MS01 the best results are obtained after decision fusion of the results obtained with the super-set of features (TDSI-Mode M3) and the results from the moving target detection.

MS02:
For MS02, the results of the TDSI module for single sensors (mode M1) and for the super-features (mode M3) are very poor. The targets are completely lost in the noise and rejected by the clutter rejection algorithm. For mode M2, results are slightly better. This is because the targets are enhanced by multiplying the feature-level-fused images while noise is tampered. The moving target detection gives better results. In the decision fusion, the best results are obtained by fusing TDSI-mode M2 with the MTD results.

MS03:
For this sequence the best results are obtained by the decision fusion of the results obtained by the TDSI module for single sensors and the MTD outcomes.

MS04 and MS05:
The MTD module does not report any targets. The best results are obtained by mode M2 of the TDSI module.

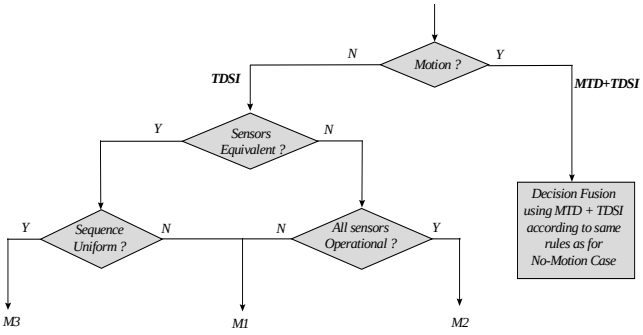
MS06:
The LW and TV sensors compete for the detection of the targets. The SW gives less good results. The best results are obtained after the decision fusion of the sensor fused data from the SKC case. It is interesting to note that the difference between the results of the SKC and the GSC case is very small.

Analysis of "best results":

We will now try to explain why a given fusion method gives the best results for each sequence. We will try to correlate subjective notions of image quality and sequence uniformity to the results.
For MS04 and MS05 the MTD doesn't find any targets, therefore it makes no sense to perform the decision fusion of TDSI results with MTD results. For these two sequences the LW and TV sensors give images that have a similar quality while the SW sensor gives a much lower contrast.
For MS01 both sensors provide images that have a similar quality and the sequence is uniform (targets have the same contrast with their surroundings throughout the sequence). For MS02 the images obtained from the LW sensor are less good than those obtained from the TV sensor.
For MS06 this is also the case. For MS03 the target is hardly visible in the beginning of the sequence while its contrast gradually increases.
We can make the following statements:

- If the sequence is not uniform, using M3 gives bad results because the test image is not characteristic for the rest of the database and using the super-set features causes features that allow detection for some sensors to be discarded.
- If the sensors are equivalent and the sequence is uniform, M3 gives good results.
- If one of the sensors provides images of lesser quality, both using the super-set of features (M3) and the single sensors (M1) give rise to too many false targets (due to noise). The sensor fusion step (M2) however reduces the potential false targets before the actual target detection.
- Because a simple multiplication is chosen for the sensor fusion step, the method based on sensor fusion (M2) is not applicable if one of the sensors is not operational.

The following figure presents these findings graphically.



Although these results seem logical, due to the fact that we only have 6 sequences to test this and because we did not use any objective measurements of image quality or sequence uniformity, this can not be generalised.

8. CONCLUSIONS

In this paper the use of data fusion at several levels is explored. The method was tested on a database of 6 multi-spectral image sequences. The approach consists of two main parts. The first part detects targets in single images (TDSI module) while the second part tries to detect moving targets (MTD module). The motion detection is performed for each sensor separately.
The TDSI module is based on texture features. Several texture features are calculated in each point of the images. These features are combined into a new image using feature fusion.
For the TDSI module three modes of operation are identified. In Mode M1 the detection is performed for each sensor separately. Mode M2 performs sensor fusion by combining the images obtained by feature-level-fusion from each sensor. Mode M3 determines the feature-level-fused images, using features from all sensors (super-features).
Decision level fusion is used to combine the results of the two parts of the algorithm.
Results show that the MTD module is very efficient when moving targets are present. In the TDSI module, the different sensors appear to be quite complementary: in some sequences the infrared sensors give the best results while in others, the visual sensor outperforms the infrared sensor.
The results show that the type of fusion that gives the best performance varies greatly from sequence to sequence. The performance is influenced by the presence of noise (in MS02), the presence of a less performant sensor (in MS02, MS04 and MS05), the type of scenario and the uniformity of the background. Although we do not pretend that there is a single optimal fusion paradigm that can solve all possible problems in all possible cases, for our test sequences, we did find a way to chose the optimal fusion method among the methods we tested, based on the criteria that are given above.

This paper has presented some of the advantages and/or disadvantages of using data fusion on different levels.
As a final conclusion one could state that different kinds of data fusion have different advantages and disadvantages and are therefore suited for solving different kinds of problems.

REFERENCES

[1] R. T. Antony, *Principles of Data Fusion Automation*: Artech House, 1995.
[2] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*: Artech House, 1992.
[3] B. V. Dasarathy, *Decision Fusion*: IEEE Computer Society Press, 1994.
[4] L. A. Klein, "Sensor and Data Fusion Concepts and Applications," *SPIE Optical Engineering*, 1993.
[5] E. Waltz and K. Llinas, *Multisensor Data Fusion*: Artech House, 1990.
[6] K. Schutte, "Fusion of IR and Visual Images," FEL-TNO, The Hague, Research Report FEL-97-B046, 4 Feb 1997 1997.

[7] L. Kuntz-Sliwa, “Optimisation d'une Configuration Multicapteurs donnée - Fusion Pixel,” in *Signal and Image dept.* Toulouse: Institut National Polytechnique de Toulouse, 1996, pp. 105.

[8] D. Hosmer and S. Lemeshow, *Applied Logistic Regression*: John Wiley & Sons, 1989.

[9] P. Verlinde, “Numerical Evaluation of the efficiency of a camouflage system in the thermal infrared,” in *ESAT*. Leuven: K.U.L., 1989.

[10] P. Verlinde and M. Proesmans, “Global approach towards the evaluation of thermal infrared countermeasures,” presented at Characterization, Propagation and Simulation of Source and Backgrounds, Orlando, USA, 1991.

[11] R. Haralick, “Statistical and structural approaches to texture,” *IEEE Proc.*, vol. 67, pp. 786-804, 1979.

[12] C. Perneel, M. d. Mathelin, and M. Acheroy, “Detection of important directions on thermal infrared images with applications to target recognition.,” presented at Forward Looking Infrared Image Processing, Orlando, USA, 1993.

[13] B. Borghys, P. Verlinde, C. Perneel, and M. Acheroy, “Long range target detection in a cluttered environment using multi-sensor image sequences.,” presented at Signal Processing, Sensor Fusion and Target Recognition, Orlando, USA, 1997.

[14] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications*. Norwood MA: Artech House, 1990.

[15] F. G. J. Absil, “Implementation of a set of Tracking Algorithms,” Koninklijke Militaire Academie, Amsterdam, Research report 94-21, 94.

[16] R. Y. Tsai and T. S. Huang, “Estimating three-dimensional motion parameters of a rigid planar patch,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 29, pp. 1147-1152, 1981.

[17] R. Morris, “Image Sequence Restoration using Gibbs Distributions,” in *Dept. Of Engineering*. Cambridge: Trinity College, 1995.