An Improved Method for creating Shared Belief in Communication Constrained Sensor Networks

Eelke van Foeken [1], Peter Hiemstra [2], Leon Kester [3]

TNO Defense, Security and Safety, Integrated Systems,
Oude Waalsdorperweg 63, 2597 AK The Hague

[1] eelke.vanfoeken@tno.nl [2] peter.hiemstra@tno.nl [3] leon.kester@tno.nl

Abstract:
Networked systems that gather sensor data in order to react to phenomena in their surroundings are faced with a growing need for adaptive behavior to operate in dynamically changing environments. In designing a networked system the data processing chain can be decomposed into functional components. These functional components interact by requesting information they need and fulfilling requests received from other components. Local evaluation of the available data with respect to the different requests and available resources is a key process in each component. An improved evaluation method is presented which is capable of locally balancing the information value against the resource costs of data. The experiments show that the evaluation method with the same amount of communication costs, results in a higher common picture quality with multiple objects and overlapping detection ranges.
Keywords: Intelligent Sensor Network, Information Value, Communication Constraints

1 Introduction

Many application domains, such as public safety, traffic management, crisis management and defense are demanding increasingly intelligent ways to observe the environment.

For example, due to the movement of maritime operations from the high seas to coastal waters, there is an increase in the complexity of image formation and situation awareness. Nowadays the image formation is platform centric, i.e., active and passive sensors are used to build an operational picture separately on each platform and only when it is more or less complete it will be exchanged through data links such as Link 11 and 16 in the form of tracks.

Considerable improvement of the common picture can be achieved by a multi-platform network-centric approach. In such an approach data from the sensors on different platforms can be exchanged (e.g., like plots) and combined in an early stage. Advantages of this approach are, amongst others, an increased chance of detection, a higher accuracy and a better track continuity.

In general, a system interacts with its environment by observing phenomena in the environment and acting on these observations to reach its goals. When this environment is dynamically changing it becomes more difficult to design such a system, in particular a decentralized networked system that is appropriate for multi-platform situation awareness. We have chosen to design the sensor network following the Networked Adaptive Interactive Hybrid Systems (NAIHS) model [1]. Using this model the architecture is decomposed into several functional components. These functional components interact by posing requests for information they need and fulfilling requests they receive from other components. This chain of interacting components from data collectors to effectors acting on the environment fulfills the role of the classic Observe-Orient-Decide-Act (OODA) cycle [2] in the NAIHS model. Evaluation of the available data with respect to the different requests and available resources is a key process in each component. This allows for a robust and flexible system where the number of interfaces needed by each function to perform its task is limited. In [3] the functional network and the platform architecture of [1] are clearly summarized.

The main contribution of this paper is the continuation of the research started in [3] and [4]. In [3] the functional architecture NAIHS is used in a simulated sensor network and a simple evaluation method is described for evaluating data. In [4] a more elaborate and general evaluation method is introduced, which provides better results. These two methods are compared in this article. We use a more elaborate scenario with multiple objects in overlapping detection ranges. This is explained in detail in section 5. NAIHS is discussed in section 2. The evaluation method is discussed in section 3. In section 4 the reward function to evaluate the data is given. Experiments and comparing the evaluation methods in a communication constrained network is described in section 5. Finally, a discussion and conclusion will be given in section 6 and section 7, respectively.

2 Functional Architecture

A networked system can be decomposed in functional components that interact by requesting information and providing requested information amongst each other. The Observe-Orient-Decide-Act (OODA) cycle [2], the Active Sensor Network (ASN) [5] and the Joint Directors of Laboratories (JDL) model [6] are three models that make a decomposition of the system. Just like these models, the Networked Adaptive Interactive Hybrid Systems [1] (NAIHS) model decomposes the system into a chain of functional components ranging from collector to effector. [5] also uses a provider-consumer architecture. The NAIHS model has the strongest resemblance with the JDL model. A thorough explanation is given in [7]. The NAIHS model uses a decomposition into eight functional components:

F1 Signal processing - generate a feature space from raw sensor data (sense).

F2 Filtering in feature space - select phenomena from the feature space likely to originate from objects of interest (detect).

F3 Filtering in time - associate detected phenomena in time and estimate state (track).

F4 Recognition - Classify and identify phenomena and/or objects (recognize).

F5 Situation assessment - determine relationships between entities (ass.sit.).

F6 Relevance assessment - threat evaluation and/or risk assessment (ass.rel.).

F7 Action assessment - decide on the actions to take (ass.act.).

F8 Execution - execute action (act).

The decomposition into different functional components is only one dimension in which the NAIHS model makes a decomposition. According to the NAIHS model it is possible to decompose a networked system along three distinct "dimensions"([3]):

1. decomposition into different information processing functions,
2. decomposition over physically separated platforms,
3. decomposition into time scales on which the system has to interact with the environment.

Our sensor network is decomposed into the first two dimensions. The first four components are used in the experiments (see fig. 1). The functional components have to communicate data and/or information requests in order to cooperate. Higher level components *request* information from lower level components, and these *deliver* information back according to this request.

3 Data Evaluation

Crucial for the operation of a system architecture as described above, is the ability of components to locally evaluate the *reward* of data given following three constraints:

1. the actual requests,
2. the internal state of the component,
3. the resources available for processing, storage and communication.

Using the results of the evaluation, the component can locally decide how and when to deliver the data to consumers, request more data from its providers, and exchange data with its siblings on other platforms.

The general aim of the research is for object assessment agents (level F2 to F4) to build up a mutual belief of the environment in a decentralized fashion. This mutual belief is to fulfill a common information need from the situation assessment process in a communication constrained network.

To create mutual belief, the object assessment agents have common methods, procedures and algorithms to process incoming information equivalently. This high commonality between two separate components is seen as interoperability level 4, given in [1]. Another

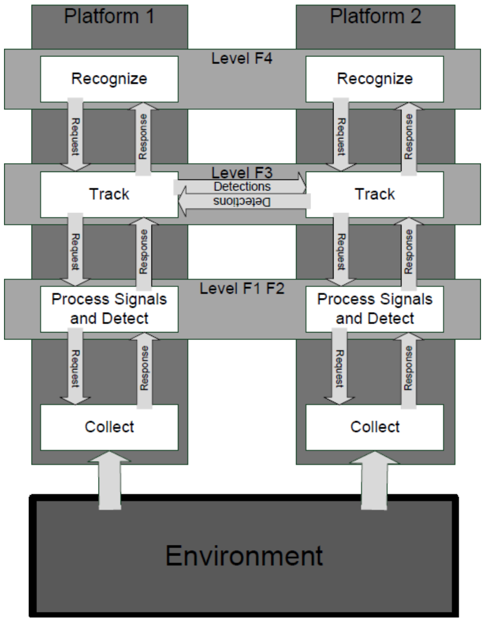

Figure 1: Schematic layout for a system of components cooperating to deliver a common state of the objects detected by the sensors on the platforms

assumption is that there is a global information request (goal), known to all object assessment agents, officially the *benevolence assumption* (see [8]).

In [5] local beliefs of distributed agents are shared by determining what they have in common. Although [5] does not assume interoperability level 4, this indicates that for fusing distributed local beliefs, common information is important.

In the next section a formulation of a general *reward function* is presented, capable of locally determining the reward of data given the three constraints. Afterwards a distributed tracking example illustrates the advantages of data evaluation.

4 The Reward Function

To create mutual belief, every object assessment agent should maintain a synchronized common state estimate $\hat{X}$ of the objects. Situation assessment agents give a global information request. The object assessment agents together build up the $\hat{X}$ based upon this request and the communication, processing and/or memory constraints. Every agent locally determines the reward R of incorporating incoming data Z_k into $\hat{X}$ by weighing the value of data $V(Z_k)$ against the costs of communicating, processing and/or storing the data $C(Z_k)$:

$$R(Z_k) = V(Z_k) - C(Z_k) \tag{1}$$

This reward function is a multi-dimensional function. By iterating over all possible subsets Z_k the maximum rewarding subset of data can be calculated. In the tracking example the reward of every single contact z_k is calculated. The reward should be higher than zero to be communicated.

Although the cost evaluation of data is a very important issue we ignore this problem in this paper and focus on finding means of valuing information. The cost $C(Z_k)$ is determined and assumed to be provided by the communication service.

In the next section the utility of information difference is used to calculate the value of local data. In [9] information-based utility is also discussed, but the utility measure used there is more based on entropy instead of situational dependent requests. They also assume unlimited communication resources as we do not.

4.1 The Value Function

Evaluating information in order to create mutual situation awareness in a resource constrained network is not new. In for example [10] policies are presented which are able to maintain a shared belief of the state of the environment in a communication constrained multi-agent team, by locally balancing information value against communication costs. The policies are based on the assumption that agents do not know anything about each other, hence not assuming functional equality. Information gain is locally determined and the relevant global information is assumed to emerge from the local interactions. In our research, commonality is assumed and has the advantage that individual agents can directly value the global information gain of local information. Moreover where the global information is not synchronized and mutual in [10], a synchronized mutual belief has the advantage of enabling well coordinated distributed actions. Most importantly, our method is able to cope with changing circumstances, like changing global requests, network structure or sensor availability.

The *value* function is a domain-specific function for locally determining the value of incorporating a data set to estimate the state of a component, given the information need or request from higher level components.

The global request or information need is a set of requests, each with a desired state and a *utility* function associated with it. The desired state is not the ground truth, because the components do not have it, but the best estimate of the state given all locally available information. To have an absolute measure of valuing information, the state estimates are compared with the desired state. Two information differences, where an information difference measure compares probabilistic states, are measured:

1. The information difference between the previous/current state estimate without the data set and the desired state

2. The information difference between the current state estimate updated with the data set and the desired state

The utility function is an abstract function for the determination of the importance of the request. This function can for example be a certain threshold, a sigmoid or a step function. By using the utility function the component calculates the utility of the first and the second information difference. Last, the value function calculates the difference between these utilities.

In formula it looks like this:

$$V_{k+t}(Z_k^i) = U(\Delta(X_{k+t}^i, \hat{X}_{k+t} \mid Z_k^i)) - U(\Delta(X_{k+t}^i, \hat{X}_{k+t})) \tag{2}$$

This function calculates the utilities U for both information differences Δ at a certain time $k + t$ separately, and subtracts them to result in the value V of data set Z at component i. Δ can be any information difference measure, like the KL-divergence, Alpha divergence, etc.. The choice is application dependent. When the information difference between the desired state X_{k+t}^i and the estimated state $\hat{X}_{k+t} \mid Z_k$ is very low compared to the other information difference, the value $V_{k+t}(Z_k^i)$ is very high, meaning this information is relevant. For a more detailed description consult [4].

5 Example: Tracking

To show the value of data evaluation in a communication constrained network of several cooperating components, consider two physically separated cooperating platforms where the local tracker agents provide an object tracking function to the recognition agents (see figure 1).

Both have a local source of detections coming from below from which object detections can be requested.

The tracking algorithm used in both tracker agents of the platforms is as follows: Object detections are delivered as contacts describing range and bearing relative to the sensor positions. The tracking algorithm is able to fuse and filter contacts into *tracks* (estimates of object position and velocity using nearest neighbor association). The tracking algorithm performs three basic steps: a *time update*, an *association* step and a *track update*. The time update can be seen as a prediction forward in time, and the track update as a correction, see [11].

The limited bandwidth and the latency in a network are considered as communication constraints. Latency is considered to be a value of time with an uncertainty. The latency in the network can be calculated with high confidence. The component sends the time of the detection along with the detection itself. This enables the trackers to do the measurement update at the appropriate time. After the measurement update it simply performs another time update.

The two platforms are assumed to be part of a larger network. We investigate a single scenario: Evaluation of the communication costs of data is assumed to be provided by the communication service on the platform. The tracker agents locally consider communication by calculating the reward of single contacts. In this scenario three possible situations can occur per time-step: no agent communicates, one agent communicates, both agents communicate. E.g. if both agents receive data-sets which bear high local value, they will send it both. Otherwise if the data-sets bear a low value, nothing will be communicated.

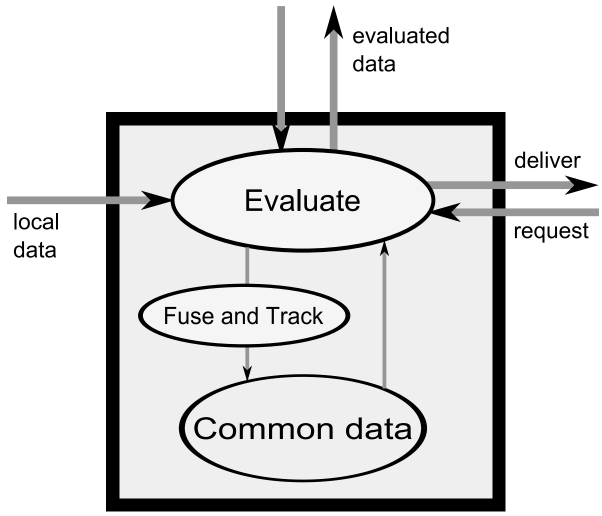

Figure 2: Schematic layout of the local process which evaluates the incoming local data and sends relevant data to the sibling components. The evaluated data is used for creating the mutual belief which is delivered to the higher recognition agents. Picture is copied from [3]

Both tracker agents receive the same global request: "Deliver a situational picture consisting of tracks for objects in the environment of the platforms. This picture should be equal and synchronized for all platforms involved". Possibly additional conditions could be imposed on the type and qualities of the data: region-of-interest, track accuracy/clarity/continuity. In all tests the individual tracker agents build up a local $\hat{Y}$ and a common $\hat{X}$ state. They process all incoming data to build up their tracks. Only contacts with high value are communicated to the other tracker agent. At a common update time both agents simultaneously update the common state to keep it synchronized. In short: both agents process the same data to build up the mutual belief.

The investigation consists of two experiments where there are multiple objects to track. The first experiment shows the difference in the resulting quality of the common picture between a scenario where the detection ranges overlap and a scenario where the objects are covered by a single sensor only.

The second experiment consists of a comparison between two methods of valuing data. The resulting track quality is measured. In the first test the local agents use the *association* *method* which selects only associated data to communicate. The second test adds the *value method*, discussed in the previous section. The requesting recognition agents request an increase in the accuracy of the tracks.

5.1 Evaluation methods

5.1.1 Association Method

This evaluation method checks whether a contact can be associated to an existing track in the common state, therefore neglecting those contacts which are either false alarms or initiations of new tracks. If the contact can be associated it is distributed to other object assessment agent and thereafter used to update the common state. Not every contact contributes to the common state $\hat{X}$. If the contact cannot be associated it is added to the local track database $\hat{Y}$. When locally a new confirmed track is found the individual contacts are also distributed to update the common state.

5.1.2 Value Method

Even though the association method limits the communication, the valuing of data can still be more sophisticated. The *value method* should be capable of more sensitive data evaluation.

The value function (equation 2) is implemented to apply to the tracking example. The data is evaluated on the effect it has on the accuracy of tracks, by means of the value function. The data consists of detections and only associated detections are evaluated. Every tracker agent uses its simple tracking process ϕ to perform the *prediction* and *association* step with a certain data set Z_k^i on the common state $\hat{X}$. The association method is used to find all the associated contacts. Now the *value method* will evaluate every contact $z \in Z_k^i$ individually. Evaluation starts by *temporarily* updating the common state with a contact and proceeds by using the value function to determine the difference in utilities between the *updated* state estimate and the *predicted* state estimate. If the reward R of the contact is higher than zero it will be communicated and used for the *permanent* measurement update of the common picture.

We use the Kullback-Leibler divergence [12] to measure the information difference Δ between the desired state X, which describes the desired accuracy, and the state estimate $\hat{X}$. The total value formula becomes:

$$\begin{aligned} V_{k+t}(z) =& U(D_{KL}(X_{k+t}^i \mid\mid \hat{X}_{k+t} \mid z)) \\ & - U(D_{KL}(X_{k+t}^i \mid\mid \hat{X}_{k+t})) \end{aligned} \tag{3}$$

$\hat{X}_{k+t}$ is the common state shared between the agents and consists of one or more state estimates of tracks $\hat{X}_{k+t} = [x^1, \ldots, x^l]$. A state estimate (consisting of a mean position μ and a covariance λ) of a track l at time $k+t$ is $x_{k+t}^l = [\mu_{k+t}^l, \lambda_{k+t}^l]$.

In the open network the reward function $R(z)$ is used. Every agent i measures the individual rewards R_{k+t} of all sensor readings $z \in Z_k^i$.

$$R_{k+t}(z) = V_{k+t}(z) - C_{k+t}(z) \tag{4}$$

If $R_{k+t}(z)$ is higher than zero the contact is communicated, and the synchronization step of the common state is done at time $k + t$. $\hat{X}_{k+t} \mid z = \phi(\hat{X}_k, z)$

5.1.3 The Utility Function (U)

U is the defined utility of the information difference d. In this experiment this is simply the following:

$$U(d) = -d^{1.1} \tag{5}$$

This means the utility of d decreases slightly exponentially with higher d.

5.2 Experiments

5.2.1 Scenario

The architecture of the demonstrator for simulating the scenario is the same as in [3]. For further information about the demonstrator please consult this article. Note, that the demonstrator displays a fair amount of reality.

The scenario for the route of the object consist of two ships observing an area on open sea by using radars. Two unidentified objects fly into the detection ranges of both ship 1 and ship 2 and both make a mirrored movement of two loops. The tracks of the objects consist of sharp curves and straight parts. In sharp curves of objects detections will generally add more accuracy information to the tracks than detections in straight parts. The value method is capable of identifying this, and this scenario is useful for showing this quality.

In comparison with the scenario in [4] there are two additions, which make this scenario more complex. First, there are the overlapping detection ranges and second, multiple objects that need to be observed. The overlapping detection ranges result in object detections of both ships, hence in near-simultaneous decentralized data evaluation of the common picture.

5.2.2 Result Experiment 1

Overlap vs No Overlap The first experiment is object to show that with equal communication rate two overlapping detection ranges make a higher quality common picture than when the objects are only detected by a single radar. The communication rate in both tests is equal and has a rate of 1.23 transmissions of detections per second. For both scenarios we did 10 runs.

In figure 3 the Mean Squared Error (MSE) of the runs are displayed of two cases: no overlapping and overlapping detection ranges. It shows that the overlapping case results in a significantly lower MSE.

5.2.3 Result Experiment 2

Value Method vs. Association Method The *association method* and the *value method* are compared by setting the number of communications against the quality of the tracks. The hypothesis is that the *value method* will cause a higher track quality with the same amount of communications, which means the communications have higher relevance.

The Association Method To simulate the communication constraints with the association method the virtual communication service comes up with an allowed percentage of commonly associated contacts. For example, if the percentage is 50% than the algorithm will incorporate every second associated contact with the same track. We have performed 10 runs of varying percentages between 10% and 100% and plot the quality of the tracks against the percentages.

The Value Method In this case the virtual communication service gives a certain fixed resource cost $C(z)$ for a contact. We perform another set of 10 runs, which cause the same spectrum of communication percentages.

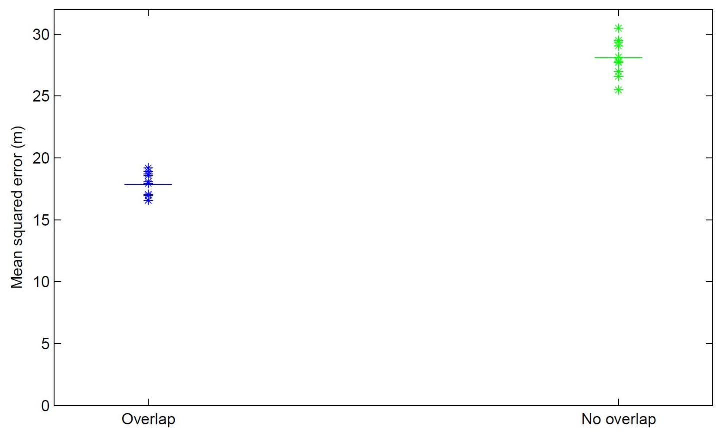

Figure 3: No overlap vs. overlap with equal communication rate. The horizontal line is the mean of the dataset. The quality of the common picture is higher in the case with visual overlap, which gives a lower MSE.

Results Figure 4 shows that for all percentages of communication the value method results in a better track quality compared with the association method. The graph shows that the lower the percentage of communication, the higher the track qualities differ.

In figure 5 parts of the tracks are displayed with the update moments in red. The difference in update rate in the curves and straight parts shows here.

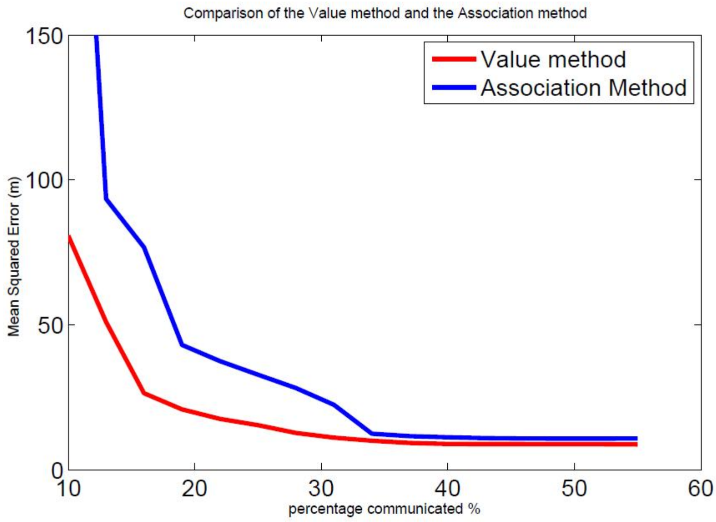

Figure 4: The value method gives a better track quality than the association method. The lower the communication rate, the higher the track qualities differ.

6 Discussion

We have done two experiments. The first experiment consisted of a comparison of the resulting quality of the common picture between non-overlapping and overlapping scenarios. The resulting figure showed an improved quality of the common picture with overlapping detection ranges. This result can be explained by the different angles from which the two ships observe the objects. The first sensor is more accurate in one direction where the other is more accurate in another direction. So, both sensors change the probability density functions of the tracks differently. The Kullback-Leibler divergence is sensitive for these different changes, and this results in the lower MSE of the tracks with an equal communication rate. In conclusion this means that the value method is capable of selecting the more informative information in case of an equal communication rate.

The second experiment consisted of a comparison of two evaluation methods: the *association method*, described in [3] and the *value method* described in [4]. First of all, the main difference between the two methods is that the value method is more abstract, and therefore applicable to a larger spectrum of functions, as long as the state information is probabilistically represented. The association method is solely aimed at tracking algorithms. Nevertheless, the comparison is of value for showing the improvement of the value method.

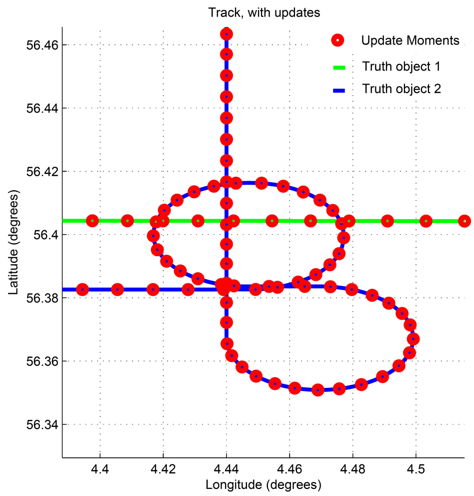

Figure 5: A part of the scenario, which shows a higher update rate in curves than in straight parts

In this more complex scenario than in [4], the value method also results in better track qualities. Thus, the value method is applicable in situations where there are multiple objects to be tracked and the detection ranges overlap. It shows that this method achieves more relevant communication than the association method. This is caused by the trackers communicating more detections and therefore performing more track updates when the object moves in a curve and less when the object moves in a straight line. The value method is more ”conscious” of a change in the direction of the object.

For different situations (different sensors, trackers, objects, etc.) the value method with fixed parameter acts differently. To make the value method work better in the future it should be more adaptive to changing circumstances. This can be done by for example changing the utility function or by learning the appropriate parameter settings.

7 Conclusions

We have chosen to design a sensor network using the NAIHS model. The *Networked Adaptive Interactive Hybrid Systems* model provides a good way to decompose a networked system into several functional components. Besides this decomposition into functional components the NAIHS model also provides a way to decompose a networked system over physically separated platforms and to decompose the system into different time scales.

We have used again the general method to create a mutual belief state in a decentralized fashion between object assessment agents, but now on a more elaborate scenario than in [4]. The *reward function* combines the three constraints mentioned in section 4 to determine the reward of updating the mutual belief state with new local incoming data.

This general method is applicable to all dynamically distributed sensor networks where the goal is to create mutual belief between object assessment agents, with the constraint that states are represented probabilistically. In this paper the focus has been on how to value information and not on how to determine the costs of information.

The results of the first experiments show that with the value method, a higher quality common picture results with overlapping versus non-overlapping detection ranges. This means the value method is well capable of dealing with near-simultaneous distributed detections from multiple platforms and can find the more relevant detections.

The results of the second experiment show that the general method has proven to work on a more elaborate scenario where there are multiple objects and overlapping detection ranges. The mutual belief state becomes more accurate than the resulting belief state of the benchmark. This means the value method is capable of a more sensitive evaluation, which results in more relevant communication in the communication constrained sensor network. In other words, the same quality of the common picture results with less communication. The value method has also proven to be adaptive to different communication constraints.

The method works on overlapping detection ranges with near-simultaneous distributed detections of the same object. Thus, decentralized data evaluation works.

We have shown that the method works with a static global information need, a single type of sensor input in different scenarios. In the future we wish to experiment with different information needs, heterogeneous sensors, more scenario's and other object assessment algorithms like recognition and classification. The greatest challenge is to make the method adaptive to different information needs and different scenario's.

References

[1] L.J.H.M. Kester, "Model for Networked Adaptive Interactive Hybrid Systems," in *Proceedings of COGIS 2006: COGnitive systems with Interactive Sensors*, Paris, France, March 15–17 2006.

[2] J. Boyd, "A discourse on Winning and Losing", Maxwell AFB Lecture, 1987.

[3] M. van Iersel, L. Kester, J. Bergmans, P. Hiemstra, K. Benoist, B. van der Broek, "Creating shared situation awareness in a multi-platform sensor network," to be published in *the 10th International Conference of Information Fusion 2008*, Cologne, Germany, June 30–July 3 2008

[4] E. van Foeken, L. Kester, "Shared Belief in Communication Constrained Networks", to be published in *The IASTED International Conference on Sensor Networks SN 2008* , Crete, Greece, September 29–August 1 2008

[5] Alexei Makarenko , Hugh Durrant-Whyte, "Decentralized Bayesian algorithms for active sensor networks", *Information Fusion*, Vol. 7, No, 7, 2006,pp. 418-433

[6] A. Steinberg, C.L. Bowman, "Rethinking the jdl data fusion levels," in *Proceedings of the Military Sensing Symposia (MSS) National Symposium on Sensor and Data Fusion (NSSDF)*, Colombia, 2004.

[7] L.J.H.M. Kester, “Designing Networked Adaptive Interactive Hybrid Systems”, *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Republic of Korea, 20 - 22 August, 2008.

[8] M. Wooldridge, *An Introduction to MultiAgent Systems*. Location: John. Wiley & Sons, Chichester, England, 2002, pp. 190

[9] B. Grocholsky, “Information-Theoretic Control of Multiple Sensor Platforms”, PhD Thesis, The University of Sydney, 2002

[10] P. Velagapudi, O. Propopyev, K. Sycara, P. Scerri, “Maintaining Shared Belief in a Large Multiagent Team,” in *10th International Conference of Information Fusion 2007*, Quebec City, Canada, 9–12 July 2007, pp. 1–8

[11] G. Welch, G. Bishop, “An Introduction to the Kalman Filter”, *ACM SIGGRAPH 2001 Course Notes*, University of North Carolina at Chapel Hill, Department of Computer Science, 2001, pp. 23

[12] S.H.K. Zhou, R. Chellappa, “From Sample Similarity to Ensemble Similarity: Probabilistic Distance Measures in Reproducing Kernel Hilbert Space”, *Transactions on Pattern Analysis and Machine Intelligence*, vol 28, No. 6, pp. 917-929., June 2006